

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PROGRAMAÇÃO EM LÓGICA, MONTAGUE E  
REGRAS LIVRES DE CONTEXTO:  
Teoria formalizada em lógica de um  
processador para a Língua Portuguesa

PAULO ROBERTO FERRARI MOSCA

Dissertação submetida como requisito parcial para a obtenção do  
grau de Mestre em Ciência da Computação

Prof. Paulo Alberto Azeredo  
Orientador  
Prof. Antonio Carlos da Rocha Costa  
Co-orientador

Porto Alegre, fevereiro de 1988.

Mosca, Paulo Roberto Ferrari

Programação em lógica, Montague e regras livres de contexto: teoria formalizada em lógica de um processador para a Língua Portuguesa. Porto Alegre, Pós-Graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul, 1988.

iv.

Diss. (mestr. ci. comp.) UFRGS-PGCC, Porto Alegre, BR-RS, 1988.

Dissertação: Processador de Linguagem natural: Inteligência Artificial: Prolog

428

681.3.11(043)  
MB94P

CPD  
1991/31674-0  
1991/12/17

UFRGS/CPD	
BIBLIOTECA	
N.º	0A28
DATA:	17.12.91

## SUMÁRIO

	(pag.)
RESUMO .....	07
ABSTRACT .....	08
PREFÁCIO .....	10
1 INTRODUÇÃO .....	19
1.1 O computador como uma metáfora para a Lingüística.....	19
1.2 Inteligência Artificial e Lingüística Teórica .....	22
1.3 História das "grandes idéias" no processamento artificial da linguagem natural .....	24
1.4 Prolog e processamento automático de linguagem natural .....	32
2 QUESTÕES FORMAIS .....	41
2.1 Gramáticas transformacionais, linguagens recursivamente enumeráveis, reconhecimento e aprendizagem .....	41
2.1.1 Gramáticas transformacionais e linguagens r. e. ....	41
2.1.2 Gramáticas transformacionais e reconhecimento de sentenças .....	44
2.1.3 Gramáticas transformacionais e capacidade de aprendizagem .....	47
2.2 Motivações teóricas para o formalismo proposto .....	49
2.3 Formalismos do modelo proposto .....	52
2.3.1 Símbolos complexos .....	52
2.3.2 Dependência não ligada e categorias derivadas .....	56
2.3.3 Definição e propriedades formais de meta-regras .....	59
2.3.4 Interpretação das regras de estrutura de frase .....	63
2.4 Anexo: Um comentário sobre as dendrolinguagens .....	65
3 UMA INTRODUÇÃO À GRAMÁTICA DE MONTAGUE .....	67
3.1 Características gerais da sintaxe e da semântica .....	67
3.2 Conceitos básicos da semântica de Montague .....	68
3.3 Um exemplo simples .....	70
3.4 Verdade relativa a um modelo .....	73
3.5 Expandindo Lo .....	75
3.6 Variáveis e quantificação .....	77
3.7 Tipos e operadores lambda .....	79
3.8 Denotação de locução nominal .....	83
3.9 Tratamento do tempo e da modalidade .....	85
3.10 A lógica intensional do PTQ .....	92
3.11 Alguns problemas na semântica das crenças ..	98

4	TRATAMENTO LÓGICO DA SENTENÇA DECLARATIVA, DA DA COORDENAÇÃO, DA ORAÇÃO RELATIVA E DAS PERGUNTAS .....	100
4.1	Tratamento da sentença declarativa .....	100
4.2	Tratamento da coordenação .....	102
4.3	Tratamento da oração relativa .....	107
4.4	Dependências nas árvores .....	113
4.5	Tratamento das perguntas .....	119
5	TRATAMENTO LÓGICO DA LOCUÇÃO NOMINAL, DA LOCUÇÃO ADJETIVA, DA LOCUÇÃO PREPOSICIONAL E DA LOCUÇÃO ADVERBIAL .....	129
5.1	Tratamento da locução nominal .....	129
5.2	Tratamento da locução adjetival .....	134
5.3	Tratamento da locução preposicional .....	135
5.4	Tratamento da locução adverbial .....	137
6	TRATAMENTO LÓGICO DA LOCUÇÃO VERBAL .....	139
6.1	Tratamento da locução verbal .....	139
6.2	Análise da passiva .....	141
6.2.1	Crítica da abordagem transformacionalista .....	141
6.2.2	Tratamento em regras livres do contexto ...	144
6.3	Tratamento das construções com auxiliares ...	151
6.3.1	Tratamento tradicional dos auxiliares .....	151
6.3.2	Tratamento lógico das construções com auxiliares .....	153
7	TRATAMENTO COM LÓGICA DOS VERBOS .....	160
7.1	Introdução .....	160
7.2	A noção de aspecto .....	161
7.3	Um cálculo lógico sobre os predicadores .....	165
7.3.1	A decomposição do significado de um verbo .....	165
7.3.2	Uma tipologia dos predicadores .....	167
7.3.3	A lógica dos predicadores .....	169
7.3.4	O papel do cálculo de predicadores na gramática .....	174
7.3.5	Uma extensão para locuções preposicionais .....	178
7.4	Tratamento com lógica do tempo verbal .....	181
7.4.1	A interação dos tempos verbais principais com os advérbios de tempo: o modelo F-P melhorado .....	181
7.4.2	Advérbios temporais aspectuais no modelo F-P melhorado .....	185
7.4.3	O significado de [ESTAR [presente simples] + GERÚNDIO] e o modelo I-D .....	189
7.4.4	Uma outra proposta: o modelo D-I .....	193
7.5	A noção de modalidade .....	201
7.5.1	O problema geral .....	201
7.5.2	Tratamento dos modos verbais .....	202
7.5.3	Futuro e modalidade .....	203

8	A GUIA DE CONCLUSÃO: O CAMINHO DA PESQUISA	205
8.1	O problema dos funtores naturais	206
8.2	O problema dos quantificadores naturais	210
8.3	O problema da pressuposição e do acarretamento	216
8.4	Uma avaliação	223
9	BIBLIOGRAFIA	225

## RESUMO

Esse trabalho assume que não existe uma diferença teórica importante entre as linguagens formais e as linguagens naturais. Uma teoria formalizada em lógica para um processador para a língua portuguesa é apresentada. A sintaxe usa apenas regras livres de contexto. A semântica é construída como está no PTQ de Montague. Existem ainda um cálculo lógico sobre o "aspecto verbal" e um tratamento lógico do tempo verbal.

## ABSTRACT

This work assume that an important theoretical difference not exists between formal and natural languages. A logic formalized theory for a Portuguese language processor is presented. The syntax use context free rules only. The semantics is Montague's PTQ like. Moreover, there are a logic calculation for "verbal aspect" and a logic treatment for verbal tense.

"I reject the contention that an important theoretical difference exists between formal and natural languages. On the other hand, I do not regard as successful the formal treatment of natural languages attempted by certain contemporary linguists. Like Donald Davidson, I regard the construction of a theory of truth - or rather, of the more general notion of truth under an arbitrary interpretation - as the basic goal of serious syntax and semantics; and the development emanating from the Massachusetts Institute of Technology offer little promise towards that end." /MON 70:189/.



## PREFÁCIO

Um observador da história pode dizer que não costuma existir nada misteriosamente objetivo sobre a seleção de um contexto histórico para um fenômeno; isto ocorre porque os contextos históricos estão embebidos num amplo contexto de coisas pessoais, tais como objetivos intelectuais, premissas filosóficas e escolhas metodológicas.

Com isto em mente e sem pretender justificar as idéias-chaves deste trabalho por razões de natureza supostamente objetiva, o autor justifica-as aqui por razões de natureza intelectual. Assim ele deve confessar que a primeira grande idéia que subjaz a este trabalho é a da possibilidade de formalização de uma língua natural. A idéia-chave aqui é a tese da inexistência de diferença entre as linguagens naturais e as linguagens formalizadas, tese esta claramente expressa por Montague alguns anos atrás. Isto contudo não é facilmente confessado por alguém que tem em sua história pessoal uma formação em Medicina, Pedagogia, Psicologia, Linguística e agora Computação, porque as três primeiras áreas não comportam facilmente coisas como formalizações; o único reduto defensivo existe nas últimas duas áreas, e nelas o autor vai se prender.

O autor reconhece entretanto que os trabalhos até agora desenvolvidos na Linguística e na Computação mostram que este projeto formalizador apresenta muitas dificuldades. Uma primeira e bastante séria dificuldade é que não está provado que a formalização total de uma língua natural (como o Português) seja possível: não está demonstrado que o Português seja um conjunto recursivamente enumerável. Em relação a isso, o autor não se propõe a fornecer aqui qualquer resposta. O que aqui é feito é a aceitação de algumas hipóteses bem definidas e o desenvolvimento delas; a partir daí outras hipóteses mais restritas são assumidas e postas a funcionar, e assim recursivamente, de modo a ir estreitando o campo dos instrumentos de trabalho (no caso, o campo formal).

Como instrumento de validação dos instrumentos, de estado em estado da construção feita, este método compara os resultados do formalismo contra a linguagem natural. Deve ser entendido que este método de comparação na verdade não prova a teoria (conjunto organizado das hipóteses) em uso, mas sim explora seus limites e na verdade oferece contribuições para a continuação do caminho. Ele não prova a validade da teoria porque, de um lado, ele não consegue explorar todas as sentenças do Português, embora se preocupe em encontrar sentenças que provem que a formalização esteja errada (ou seja insuficiente para cobrir uma sentença do Português, ou seja capaz de gerar uma cadeia não aceitável); de outro lado, a concordância da teoria com os fatos linguísticos não prova a validade da teoria, porque várias teorias alternativas podem mesmo ser possíveis: uma linguagem pode ser gerada por um número infinito de gramáticas, como mostra a Teoria das Linguagens Formais.

Em relação ao método usado de ir das hipóteses para os fatos linguísticos, deve ser expressa a crença do autor de que não é possível um caminho que brote naturalmente dos fatos. Isto é justificado pelo autor com a afirmação de que não existe uma observação pura dos fatos, tirando daí as suas essências, para um mortal como o autor. Crê contudo o autor que estas essências existem, mas ele não as conhece e com isso é obrigado a formalizar hipóteses, bater seus efeitos contra a realidade linguística, reformular a teoria se necessário e seguir de novo, construindo redes teóricas que procurem apanhar o real (no caso, a língua portuguesa).

O outro critério de validação é que a teoria não seja inconsistente. Quanto a isso, não é necessário aqui muitos comentários, visto que o formalismo utilizado pelo autor é a lógica de primeira ordem, em especial o universo das cláusulas definidas de Horn. Em suma então, sob estas duas meta-restrições, a maquinaria formal é construída logicamente e explorada empiricamente contra a linguagem natural.

Um problema importante com este método é o da especificação de qual a linguagem natural a ser utilizada. Aqui podem ser dadas duas respostas: a linguagem que as pessoas possuem nas suas cabeças (a competência linguística das pessoas) ou a linguagem que as pessoas falam (aquilo que aparece como desempenho linguístico). Como escolha metodológica aqui é assumida a segunda opção, sendo neste trabalho afastada qualquer preocupação de índole psicológica ou neurológica. Aqui surge um tema importante mesmo para a Inteligência Artificial: o autor aceita uma proposta de que para construir um programa inteligente (no caso, um programa que encontre o significado das sentenças tecladas para um computador), o melhor modelo é como as pessoas fazem isso. Contudo o autor é obrigado a reconhecer que o conhecimento disponível no campo da Neurofisiologia hoje é insuficiente para tal empreendimento; por outro lado, o conhecimento disponível no campo da Psicologia Cognitiva atém-se a construções de modelos abstratos, os quais ainda estão (de um modo bastante interessante) muito imbuídos de noções e formalismos produzidos no campo da Inteligência Artificial. Então este empreendimento (o de utilizar "modelos reais") ainda não parece ser totalmente viável, e o autor abandona-o aqui.

Dizendo as coisas de um modo mais claro, o autor não formula a hipótese de que a mente (ou a cabeça) das pessoas processa a língua portuguesa segundo os moldes da teoria aqui apresentada. Não pode o autor aqui oferecer nenhuma justificativa psicológica ou neurológica para a sua teoria, nem é este seu objetivo. O campo restrito (e daí criticável) de seu trabalho se restringe em estabelecer um formalismo que seja implementável (isto é, que exista na forma de um programa) e que processe uma linguagem natural, no caso o Português, no sentido de parecer ao usuário a) que o programa compreende o significado das sentenças portuguesas digitadas num teclado, b) que este programa armazene as informações recebidas nestas sentenças e c) que possa responder a perguntas feitas de um modo natural.

A pergunta subsequente, e presa visceralmente a anterior, provém dos estudos da Sociolinguística e leva em conta que as pessoas falam diferentemente, dependendo de sua classe social, localização geográfica, idade, etc. A restrição aqui imposta pelo autor é a de trabalhar com a linguagem natural que o autor considera como natural, isto é, para resolver problemas de gramaticalidade o autor conta exclusivamente com a sua intuição linguística do Português. Assim aquelas cadeias que são acopladas com um julgamento de aceitabilidade positivo são vistas pelo autor como sentenças do Português. Em comum com a maioria dos linguistas do campo da Linguística Formal, o autor deste trabalho assume que as cadeias inventadas e certos julgamentos intuitivos sobre elas constituem dados legítimos para o trabalho de formalização. Além disso, a posição deste autor é que, se certa cadeia é julgada como inaceitável ou desviante, a classificação do desvio em sintático ou semântico não é parte da intuição do falante, mas um produto da teoria construída.

Um segundo problema, também importante, diz respeito ao significado do formalismo, no sentido do estabelecimento de sua função. Isto está calcado na constatação de que uma linguagem natural existe fundamentalmente para a comunicação de significados e intenções entre as pessoas. A pressuposição, quanto a isso, implícita em todo este trabalho, é que estes significados e intenções sejam formalizáveis e, além disso, implementáveis numa linguagem de programação (no caso, Prolog).

Em relação a isso, deve ser lembrado que a codificação linguística é uma expressão binária que designa duas entidades: a entidade codificada (o significado, a mensagem, a função) e a entidade codificante (o signo, o código, a estrutura). Nos sistemas biológicos, a distinção entre estas duas entidades não é trivial; porém, no que diz respeito ao estudo da linguagem, ela é similar ao estudo de anatomia-cum-fisiologia. Isto é: afastando os importantes aspectos socio-culturais, psicoemocionais e estéticos do terreno das funções da linguagem natural, é possível reconhecer tres aspectos funcionais que podem receber codificação sistemática e distinta na linguagem humana: a semântica lexical, a semântica proposicional e a pragmática do discurso.

No que diz respeito à base funcional da semântica lexical, a idéia é que existe primariamente, no estoque do conhecimento geral, um léxico; este conhecimento pertence a fenômenos, conceitos ou pontos de referência relativamente estáveis que constituem uma rede intrincada que é parte do mapa cognitivo humano dos fenômenos do universo. Por outro lado, a base funcional da semântica proposicional pertence a informações específicas guardadas em proposições, as quais são codificadas sintaticamente em sentenças; isto envolve primariamente dois aspectos da proposição: sua caracterização como estado, evento ou ação e a caracterização dos participantes na proposição (isto é, a especificação de seus papéis semânticos em relação ao predicado). As questões que dizem respeito à base funcional pragmática do discurso envolvem o sequenciamento das proposições atômicas dentro de um amplo contexto comunicativo, isto é, no discurso, onde o núcleo deste contexto específico envolve os objetivos dos falantes, a interação social entre o falante e o ouvinte, e o contexto do discurso.

No que se refere a estes tres níveis, deve ser dito que este trabalho na verdade se restringe ao nível da semântica lexical e da semântica proposicional. Quanto a estes dois níveis, este trabalho assume que eles são codificados via meios estruturais, de modo que eles podem ser descobertos e descritos via meios estruturais (o que é imposto pela necessidade de construir um programa de computador). Nesta abordagem restrita, a noção de função de comunicação não chega a desempenhar um papel explícito na descrição da estrutura da linguagem natural; porém fica implícito uma assunção forte relativa à natureza não-arbitrária da correlação entre código e mensagem. O que é levado em conta para o estabelecimento desta correlação é a ordem das palavras, a morfologia gramatical e flexional e certas restrições sintáticas. Levando isso a um contexto de um programa de computador, isto envolve aceitar que a ordem das palavras e a forma das palavras sejam suficientes para a realização da extração do significado das proposições do falante que está a "conversar com o computador".

A segunda grande idéia deste trabalho é que o formalismo proposto por Montague é melhor do que o formalismo defendido na tradição gerativo-transformacional. Com esta proposta o autor pisa em valores bastantes sérios para os linguistas modernos, porque as gramáticas transformacionais, nas suas diversas versões, trouxeram e ainda trazem contribuições importantes para a hipótese da formalização das linguagens naturais.

O caminho escolhido contudo é o montaguiano, e isto por dois motivos principais: o primeiro é que as gramáticas transformacionais foram abandonadas como modelos para a construção de processadores de linguagem natural já na década dos anos 60, visto a dificuldades muito sérias de implementação (uma discussão mais aprofundada aparece numa secção deste trabalho); em segundo lugar, e isto de um modo positivo, o formalismo de Montague utiliza a lógica, a qual é o campo onde o autor se dispõe a trabalhar. Dentro deste campo montaguiano, o autor apresenta contudo algumas assunções restritivas: a primeira delas envolve trabalhar somente com regras livres de contexto (fora então das transformações chomskianas e de algumas regras estabelecidas mesmo por Montague); em segundo lugar, o autor assume a hipótese de que para cada regra sintática exista uma regra de interpretação semântica (Hipótese Regra-para-Regra), sendo que as duas são escritas juntas na mesma cláusula, como ocorre no modelo das Gramáticas das Cláusulas Definidas de Colmerauer. Deve ser referido aqui que este caminho enfrenta dificuldades, basicamente devido a que a tradição montaguiana ainda é muito pequena nos trabalhos de Linguística no Brasil e em Portugal.

A terceira grande idéia é de natureza mais restrita ainda: se a segunda grande idéia é uma especificação da primeira, a terceira é uma especificação da segunda. A proposta aqui é que a formalização no "espírito das idéias de Montague" seja feita na linguagem de programação Prolog. Em relação a isso, este trabalho não apresenta nenhuma apresentação didática da linguagem Prolog, assumindo que ela seja do conhecimento do leitor.

A segunda e a terceira grandes idéias não são neste prefácio fortemente desenvolvidas, visto constituírem o corpo dos capítulos que constituem este trabalho. A pretensão do autor, em relação a estas duas grandes idéias, é a de simplesmente mostrar aqui como elas foram se desenvolvendo, isto é: contar sua gestação, do ponto de vista da história intelectual do autor.

No que diz respeito à história intelectual do que aqui é apresentado, o seu nó inicial pode ser encontrado num trabalho (origem de dissertação de mestrado do autor em Linguística) apresentado nos Encontros Linguísticos da PUC/RS em 1982 e chamado de "How to do things (real conversations) with speech acts: Questions and answers around a Brazilian kitchen", o qual tratava da formalização dos atos de fala de pedido de informação e de resposta. O autor procurou aí discutir basicamente a proposta formal da Teoria dos Atos de Fala pela qual estes deveriam ser tratados como aplicação de uma função  $F$  (intenção ilocutória) ao conteúdo proposicional  $p$  do ato. Fazia parte desta proposta erotológica também uma classificação do par pergunta-resposta, ocorrendo sobre isso uma formalização do aspecto semântico através de uma lógica intensional que usava conectivos lógicos, quantificadores, operadores  $\lambda$ , operadores de intensão e de extensão. Este trabalho foi publicado em parte com o título de "Sobre perguntas e respostas", ficando contudo a parte formal excluída devido a problemas de impressão.

Um novo contexto se construiu a partir de 1983 a partir do trabalho, em conjunto com o professor Antônio Carlos da Rocha Costa, de construção na linguagem de programação Logo de um processador de linguagem natural, usando como quadro teórico a Gramática de Casos. Este programa gerava uma representação semântica (identificação do predicado e dos casos-argumentos) das sentenças declarativas escritas e também respondia a perguntas contra um banco de dados em Logo (armazenado na forma de listas de propriedades). O programa construído (sistema PAMONHA) foi logo abandonado, sendo visto pelos autores como um mero exercício intelectual.

Uma nova mudança importante, concomitante com o surgimento do Grupo de Inteligência Artificial na UFRGS, ocorre em 1984 na utilização de interpretadores da linguagem Prolog. Ao lado disso ocorria o trabalho sobre SIACIA de Rosa Viccari, o qual envolvia, entre outras coisas, uma interface de linguagem natural implementada em Prolog e baseada no formalismo conhecido como Gramática de Extraposição.

Decorrente desta guinada para Prolog, o autor passou a elaborar formalizações potencialmente implementáveis em Prolog, sendo que durante o ano de 1984, alguns textos sobre o componente sintático de um processador de língua portuguesa foram produzidos pelo autor. Estes textos não tiveram publicação formal, apenas circulando entre os componentes do Grupo de Inteligência Artificial da UFRGS. É importante observar que estes textos constituem o nó duro para a realização de uma crítica ao formalismo conhecido como Gramática Transformacional.

Encetou o autor, a partir daí, um projeto de ligar a vertente, perdida em 1983, de formalização da língua portuguesa usando lógica intensional com a vertente de utilização da linguagem de programação Prolog, tendo como objetivo a construção de um processador de língua natural para uma base de dados em Prolog. Assim em 1984 o autor apresentou no I Simpósio Brasileiro de Inteligência Artificial, organizado pelo Grupo de Inteligência Artificial e realizado na UFRGS, um trabalho sobre o formalismo conhecido como Gramáticas Transformacionais, procurando mostrar a inadequação (ao nível formal e ao nível de implementação) deste formalismo para a implementação de um processador de linguagem natural.

A esse trabalho seguiu-se a proposta, apresentada em 1985 no II Simpósio Brasileiro de Inteligência Artificial, de um programa que procurava estabelecer um tratamento em lógica da língua portuguesa, usando uma sintaxe composta exclusivamente de regras livres do contexto (acrescidas de símbolos complexos, isto é, que são como termos) e uma semântica formulada em lógica intensional.

O projeto tomou um corpo mais avançado no trabalho, apresentado em 1986 no III Simpósio Brasileiro de Inteligência Artificial, sobre o tratamento em lógica das construções com auxiliares da língua portuguesa, usando somente regras livres do contexto com símbolos complexos. O tratamento semântico do tempo verbal contudo foi apresentado aí com sérios problemas; uma proposta de solução, através do uso de operadores temporais fora da lógica de Prior, só foi construída em 1987.

A elaboração mais plena do trabalho aqui apresentado ocorreu no ano de 1987, onde os vários formalismos foram construídos, testados e depurados.

Analisando esta história, pode ser dito que o trabalho apresentado aqui envolveu o casamento da vertente linguística e da vertente computacional, havendo alguns bosquejos no campo da epistemologia e da ontologia, especialmente no que diz respeito ao uso da Teoria dos Modelos, a questões de natureza metodológica e a alguns problemas de formalização.

Em relação à vertente linguística, deve ficar claro que ocorreu a rejeição, por razões formais e práticas, de encetar um trabalho com qualquer uma das versões das gramáticas transformacionais e a aceitação dos seguintes pressupostos:

- sintaxe composta por regras livres de contexto, ampliadas com símbolos complexos e meta-regras;
- semântica do tipo teoria dos modelos;
- sintaxe e semântica compondo uma linguagem intensional, no "espírito" da Gramática de Montague.

A vertente computacional, por seu turno, fornece de um lado o suporte concreto para a realização da vertente linguística, e de outro oferece como ferramenta de trabalho para a formalização linguística a lógica oriunda da linguagem de programação Prolog. Assim sendo, os pressupostos aqui são tais que:

- a teoria definida na vertente linguística deve ser implementável facilmente na linguagem Prolog, e
- a ferramenta de formalização deve estar bem próxima a um conjunto de cláusulas Prolog.

A vertente propriamente filosófica diz respeito a:

- comparações epistemológicas entre o nível de conhecimento do programa e os conhecimentos das pessoas sobre a língua portuguesa, e
- representação em cláusulas Prolog de uma ontologia dos predicadores da língua portuguesa.

No que diz respeito à importância de um trabalho de formalização dentro do campo da Inteligência Artificial, deve ser dito que a Inteligência Artificial não possui hoje um conhecimento plenamente formalizado (exceto na área dos provadores de teorema) sobre os sistemas de inteligência artificial, e isto quer na parte mais propriamente teórica (o conceito de inteligência artificial, por exemplo), quer na parte mais da construção de sistemas com inteligência artificial. Isto está ligado ao reconhecimento, muito difundido na área, de que a maioria dos sistemas computacionais inteligentes possuem uma estrutura com muitos pontos ad hoc.

O campo em que esta deficiência é mais notória diz respeito ao que se convencionou chamar de "representação do conhecimento", embora até hoje não exista um acordo sobre o que este termo realmente significa /BRA 80/ /BOB 85/. Uma idéia importante hoje neste campo é que todo conhecimento consiste na formação e uso de uma representação apropriada: dada a natureza computacional, em Inteligência Artificial, esta representação tem que ser formal, porém o campo apresenta hoje uma pergunta sem resposta sobre qual a melhor forma de representação.

Pode ser colocado que a formalização de uma linguagem natural, na forma da construção de um processador, possui algumas contribuições ao processo histórico do desenvolvimento da Inteligência Artificial.

Em primeiro lugar, e de um modo fraco em relação à pergunta acima, a construção de um processador de linguagem natural permite tornar mais amigável a interação com os sistemas inteligentes artificialmente; esta tem sido a utilização mais tradicional na área, em especial a sua utilização em interfaces em língua natural para consulta a bases de dados (as vezes, bases de conhecimentos).

Em segundo lugar, e de um modo mais ligado à pergunta acima, a teoria e a construção dos processadores de linguagem natural pode oferecer uma resposta provisória à questão do problema da representação do conhecimento. A idéia chave aqui é que a maior parte do conhecimento humano hoje parece adquirido via linguagem natural (escrita ou falada), e neste sentido um sistema computacional que "souber falar uma linguagem natural" pode apreender vários conhecimentos de muitos campos científicos. Abolindo questões que dizem respeito à eficiência (tempo de resposta e tempo de aprendizagem), esta solução é interessante o suficiente para ser encetada: ela assume, por exemplo, que, entre os meios cognitivos onde ocorre a representação do conhecimento humano, existe um nível profundo único onde todas as formas de representação são mapeadas /MOS 83/ /MOS 87/ /MOS 87a/.

Uma hipótese forte hoje pode mesmo assumir, além disso, que a representação em lógica é este "último nível" e que ela é suficiente para suportar as outras formas de representação (pictóricas, procedurais, redes semânticas, frames, scripts), e isto tem implicações importantes para a construção de sistemas artificialmente inteligentes.

Um outro ponto que merece ser comentado ao fim deste prefácio diz respeito ao trabalho de formalização dentro do campo da Linguística propriamente dita. A intenção do autor aqui, a esse respeito, é de apenas defender o projeto de formalização da língua portuguesa. Quanto a isso, o autor deste trabalho acredita que a formalização é importante porque força o teórico a listar suas primitivas e a expor sua ontologia, o que neste trabalho é encetado. Por outro lado, a formalização obriga a estandarização do rigor e da exatidão na expressão da teoria: as propriedades lógicas de um sistema formalizado são, de uma certa forma, independentes de sua interpretação por um humano. Isto se opõe ao trabalho com sistemas informais, onde a apresentação de um contra-exemplo (o qual desempenha um papel importante na proposta do autor) assume as vezes um caráter perverso. Do ponto de vista do autor, a formalização possui um caráter heurístico, na medida em que permite restringir modelos a estruturas que podem jogar um papel importante (negativo e positivo) no processo de descoberta: colocando uma formalização precisa mas inadequada para uma conclusão inaceitável (no caso, uma cadeia que não corresponde a uma sentença do Português, ou uma resposta não aceita como tal), pode ser exposta a fonte desta inadequacidade e logo pode ser ganho uma compreensão profunda sobre os dados linguísticos. Por outro lado e mais positivamente, uma teoria formalizada pode prover soluções para problemas além daqueles para os quais ela foi explicitamente montada.

A questão da formalização é hoje extremamente importante, do ponto de vista do autor, para os que trabalham no campo da Linguística Portuguesa, devido à insuficiência de trabalhos na área. Poderia se objetado aqui que a formalização da sintaxe e da semântica do Português, nos moldes feito pelo autor, é ainda prematura, porque a linguística brasileira e a portuguesa estão ainda aí engatinhando. Esta objeção ao trabalho do autor é, de seu ponto de vista, enganadora por princípio, visto que a formalização de uma teoria não é nunca prematura: se a teoria é trivial e incoerente, então a formalização rapidamente irá tornar este fato evidente. Se a teoria é só fracamente preditiva, então isto ficará transparente; por outro lado, se a teoria é forte e altamente preditiva, então isto ficará óbvio. Na posição do autor, se for pensado que nenhuma teoria deve ser formalizada até que esteja correta, então toda formalização linguística seria prematura.



Este trabalho não conclui por uma resposta afirmativa às tres grandes hipóteses referidas antes (ou, se se quiser, à única hipótese referida, com as últimas duas sendo vistas como restrições). Este trabalho não conclui também pela falência desta proposta. O que este trabalho faz é mostrar uma maquinaria teórica formal, implementável em Prolog, que atinge setores importantes da língua portuguesa, tal que este programa Prolog possa conversar com um usuário falante do Português. Em conclusão: existe uma rede teórica que apanha pedaços importantes do Português e até aqui "a coisa está dando certo".

Por último, mas não por ser o menor, o autor expressa aqui seu agradecimento ao professor Antonio Carlos da Rocha Costa pelo acompanhamento feito a este projeto de trabalho.

## 1 INTRODUÇÃO

### 1.1 O computador como uma metáfora para a linguística

A primeira vista, "ciência" e "metáfora" parecem vir de mundos diferentes: a ciência está baseada na matemática, no raciocínio formal e na precisão, enquanto a metáfora enfatiza a poesia, a analogia imprecisa e a sugestão. Isto no entanto pode ser verdade para as teorias detalhadas da "ciência normal", mas não vale para o modo como os novos paradigmas são desenvolvidos e como as assunções de um campo são determinadas.

A chave é que um cientista, olhando para um novo paradigma, costuma ser fortemente influenciado por outras ciências as quais estão no momento tendo um sucesso no que diz respeito a seu desenvolvimento. E isto significa que este cientista trata essa ciência como um "modelo" e importa metaforicamente suas idéias, especialmente sobre quais perguntas devem ser feitas e que tipos de respostas devem ser aceitos.

A história da Linguística mostra que ela é uma ciência bastante aberta a este tipo de redefinição, em geral usando as "ciências duras" como bases para as analogias.

Uma metáfora que foi usada no estudo da linguagem humana foi a da "gramática prescritiva", a qual assumiu a Linguística como algo preso a leis de Direito. Deve ser dito aqui, a bem da verdade, que nenhuma teoria linguística séria hoje assume esta metáfora.

No século 19, um paradigma que emergiu para a Linguística foi o da filologia comparada: aqui os linguistas começaram a se preocupar com as similaridades entre as línguas, especialmente o vocabulário e os padrões de sons. Muito do sucesso deste paradigma é devido ao estilo da História Natural, especialmente o estudo da evolução biológica.

A revolução que se seguiu à filologia comparada supôs a troca do foco na família de linguagens humanas para o foco na estrutura de uma única linguagem. O problema básico aqui foi o de descrever as regularidades encontradas nas elocuições de uma linguagem natural. A linguística estrutural, nome deste movimento, foi aí fortemente influenciada pelo behaviorismo, paradigma que dominou a psicologia americana aproximadamente durante o mesmo período. Deve ser ressaltado que a análise dos dados linguísticos foi aí modelado por uma visão positivista das ciências empíricas, a qual enfatizava o uso de técnicas experimentais que rigorosamente deveriam controlar a estrutura. Esta metodologia foi usado no estudo da fonologia, mas encontrou sérios problemas quando aplicado ao estudo da sintaxe.

Nos últimos 25 anos um novo paradigma emergiu, em grande parte devido ao trabalho do grupo de Noam Chomsky. Este paradigma rejeita a metodologia positivista, colocando em oposição que a análise estrutural não capturava a criatividade linguística (elemento essencial à linguagem humana). O problema central aqui passa a ser o de caracterizar a gramática da linguagem, ou seja, o conhecimento tácito que um falante/ouvinte tem sobre a linguagem humana que ele usa. Esta abordagem estabelece o conceito de uma competência abstrata, o qual é fortemente relacionado à noção de prova matemática.

A corrente de trabalhos produzidos (dita Linguística Gerativa) vê a linguagem humana como um objeto matemático e procura construir teorias para a linguagem de um modo muito próximo a um conjunto de axiomas e regras de inferência. O maior sucesso da Linguística Gerativa foi na área da sintaxe, mas ocorreu um bom trabalho também na área da fonologia gerativa.

Nos últimos anos apareceu um novo paradigma para o estudo da linguagem natural, a partir da utilização da metáfora do programa estocado num computador digital. Este paradigma assume que existe um programa e um conhecimento estocado, ao lado de um processador e de estruturas temporárias (estruturas de trabalho e estruturas assinaladas). O processador usa o corpo dos conhecimentos estocados (a gramática, o dicionário, etc) para operar sobre as estruturas temporárias, de acordo com o programa estocado: o processador opera sobre as estruturas dadas (produzidas por outros processos) e gera estruturas que outros processos podem usar. Estas divisões são contudo não homogêneas dentro do campo de trabalho e também são problemáticas, em primeiro lugar porque a distinção entre programa e conhecimento estocado (dados) apenas toca na questão da distinção entre um programa e seus dados e, no que diz respeito, mais à Inteligência Artificial, toca na discussão entre a representação procedural e a representação declarativa. Por outro lado a distinção entre processador e programa é também problemática, porque toca na questão da existência de um processador especializado para certos processos versus a existência de um processador geral que usa o programa; em relação a isso, deve ser dito que na maior parte dos modelos descritos para processamento de linguagem natural a questão do tipo de hardware ainda aparece pouco. Por último, a distinção entre as estruturas permanentes (programa e conhecimento estocado) e estruturas temporárias (estruturas assinaladas e estruturas de trabalho), é uma simplificação, porque a maioria dos modelos computacionais da linguagem usa só o processamento ao nível da sentença, sem levar em conta, por exemplo, o processo de aquisição de um novo conhecimento.

Deve ser ressaltado que os dois últimos paradigmas (o gerativo e o computacional) apresentam algumas zonas de desacordo. A primeira zona é que o paradigma gerativo divide o conhecimento da linguagem no aspecto da competência e no aspecto do desempenho, tal que a estrutura da competência possa ser caracterizada independentemente de qualquer processo pelo qual ela se manifeste: o componente do desempenho é tido, então, como teoricamente secundário à especificação independente da competência. O paradigma computacional, em oposição, vê a estrutura da linguagem como derivada da estrutura dos processos: os modelos computacionais se preocupam com a organização do conhecimento linguístico de um modo que possa servir como um procedimento efetivo. A idéia geral aqui é então a oposição entre gramática e procedimento: esta oposição, contudo, pode ser abaixada dentro do campo das linguagens declarativas de natureza lógica /KOW 79/, nas quais a competência ficaria ligada ao conjunto das declarações (gramática) e o desempenho seria ligado ao processamento propriamente dito.

Uma segunda zona de desacordo diz respeito à relevância do conhecimento não linguístico: a maioria dos pesquisadores no campo do paradigma computacional preocupam-se com as interações entre o conhecimento linguístico e o não linguístico, e no desenvolvimento prático de sistemas computacionais eles enfatizam o que há de comum entre a linguagem e as outras faculdades humanas. No paradigma gerativo, em especial o chomskyano, é assumido que existe uma "faculdade específica da linguagem", possuída pelos homens e não por outros animais, tal que esta faculdade determina a estrutura (a competência) da linguagem.

Uma primeira grande limitação hoje do paradigma computacional diz respeito à pouca importância dada à dinâmica social da linguagem humana, especialmente no que diz respeito à dificuldade do tratamento da linguagem como um fenômeno social. Deve ser ressaltado, contudo, que o paradigma computacional não é contraditório com os aspectos sociais da linguagem humana, mas existem hoje muitas dificuldades no estabelecimento de um trabalho em conjunto das duas áreas.

Uma segunda limitação do paradigma computacional diz respeito às dificuldades no tratamento dos aspectos emocionais da linguagem. Algumas tentativas foram feitas no campo, por exemplo, da "poesia computacional", mas o trabalho aqui está no seu início.

Uma terceira limitação está ligado ao problema de que o paradigma computacional não atinge hoje o aspecto da mudança histórica que sofre uma linguagem humana. Isto é ainda muito pouco discutido dentro do paradigma computacional, porque toca num dos problemas bastantes sérios qual seja o dos "programas que aprendem e evoluem".

Em adição a estas limitações, impostas pelo foco hoje na estrutura do conhecimento de um falante/ouvinte, existem algumas outras limitações potenciais devidos à premissa de que este conhecimento pode ser modelizado por um conjunto de estruturas formais.

## 1.2 Inteligência artificial e linguística teórica

Este trabalho é o produto do relacionamento entre os campos da Inteligência Artificial e da Linguística Teórica. A área da Inteligência Artificial representada aqui é aquela cujo núcleo é a produção de programas em computador capazes de analisar uma linguagem natural, usualmente para algum objetivo prático como a construção de respostas a perguntas feitas contra uma base de dados (mais especificamente, uma base de conhecimentos). A linguagem natural referida no caso é a língua portuguesa falada no Brasil, e mais especificamente aquela falada pelo autor deste trabalho. Por outro lado, a Linguística Teórica, como representada aqui, toma o estudo da linguagem natural como um fim em si mesmo, e tem como campo de interesse fornecer um teoria coerente de como a linguagem natural funciona.

Claramente cada campo pode ser capaz de fazer uso do outro campo. O linguista teórico pode ver um programa de computador como um modo válido de demonstrar que uma teoria é de fato consistente. O trabalhador no campo da Inteligência Artificial pode fazer uso das propostas teóricas dos linguistas como alternativas para a própria construção de seus programas.

No campo prático do relacionamento da Inteligência Artificial e a Linguística Teórica, deve ser observado que é verdade que alguns trabalhos sobre processadores de linguagem natural foram teóricos desde o início, mas na grande maioria das vezes o que é publicado como "relações entre o processamento prático e os resultados teóricos" são construções meramente post hoc /WIL 83/, nestes casos, a idéia procedural motivadora do processador precedeu a busca da justificação teórica.

Desde o início dos projetos de tradução por máquina nos anos 50, esta "visão procedural" da linguagem natural prevaleceu, não só pelo fato de que um programa é escrito, mas também num sentido mais fundamental divisado por Wittgenstein e apresentado como "não pergunte pelo significado, pergunte pelo uso". Isto fez com que a ênfase distintiva no produto do processamento artificial se opusesse a um tratamento formal, e isto acarretou a visão de que as gramáticas dos programas e os conhecimentos linguísticos eram totalmente diferentes. Além disso, esta distintividade foi reforçada pela sensação de auto-confiança ganha pelos trabalhadores no campo computacional, devido à construção de sistemas computacionais que funcionam, em especial nos últimos dez anos.

Apesar disto, existem boas razões para olhar o campo do processamento artificial de linguagem natural como uma tradição intelectual. E isso é feito brevemente a seguir no que diz respeito às "grandes idéias" subjacentes na construção dos processadores de linguagem natural.

Clarificando em primeiro lugar a história das "grandes idéias" da Linguística Ocidental, podem ser detectadas no século 20 duas grandes teses /MOS 84j/:

- a formalização da sintaxe das linguas naturais é possível via regras de reescrita;

- a formalização da semântica das linguas naturais é possível via teoria dos modelos.

A primeira tese começou com o projeto das Gramáticas Transformacionais de Chomsky, no início dos anos 60. A segunda tese teve como pilar fundamental os trabalhos de Montague, no início dos anos 70.

A primeira tese tem aqui aceita sua forma, mas não seu conteúdo, pois este trabalho apresenta uma recusa ao projeto das gramáticas transformacionais. Aqui são mostrados alguns problemas críticos ao projeto transformacionista, e é apresentado justificadamente como substituto um formalismo em lógica de primeira ordem (restrito a cláusulas definidas ou de Horn) e que só utiliza regras livres de contexto, sem transformações. No que diz respeito à segunda tese, na metade dos anos 60 os linguistas ocidentais reconheceram que um projeto para a Linguística era a construção de uma teoria explícita e compreensiva do significado para as sentenças das linguagens naturais. Ao lado disso, dentro da lógica matemática, existiam já teorias do significado para as linguagens formais empregadas pelos lógicos: a semântica de valor verdade para o cálculo proposicional foi viabilizada por Boole, a semântica referencial para o cálculo de predicados de primeira ordem existe desde Frege, a semântica dos mundos possíveis para a lógica modal existe desde Kripke. O que todas estas teorias semânticas tinham em comum era seu apelo a condições de verdade: elas assinalavam a cada sentença das linguagens lógicas as condições nas quais estas sentenças podem ser verdadeiras (esta concepção semântica de verdade já está em /TAR 44/).

Em torno dos anos 70, a necessidade de uma teoria do significado entre os linguistas era imperativa. A única teoria que havia sido desenvolvida até então dentro do paradigma transformacionista (a teoria de Katz e Fodor) não era tomada muito a sério. No mesmo período, filósofos influentes da linguagem (Harman, Davidson) e lógicos importantes (Montague) começaram a dizer que não haviam diferenças teóricas importantes entre as linguagens naturais e as linguagens artificiais dos lógicos. Nos anos 70 ocorreu uma confluência importante entre estes dois grupos, sendo que as posições mantidas aí foram as seguintes:

- as linguagens naturais e as linguagens formais não são teoricamente distintas;

- as sentenças de uma linguagem natural mapeiam-se em sentenças de uma linguagem formal;

- os significados das sentenças de uma linguagem natural podem ser capturados por suas contrapartes na linguagem lógica;

- os significados dos funtores lógicos das linguagens naturais correspondem aos significados de suas contrapartes na linguagem lógica;

- os significados das sentenças de linguagem lógica são dados por uma teoria que estipula as condições sob as quais as

sentenças são verdadeiras.

Nos anos 80, foram detectados alguns problemas no que diz respeito ao significado dos quantificadores das línguas naturais, das conjunções, das expressões modais, da negação e das pressuposições. Uma vertente que procurou solucionar estes problemas assumiu a solução no campo da pragmática, campo este iniciado nas aulas sobre "lógica e conversação" dadas pelo filósofo Paul Grice em 1967, em Harvard; estas aulas forneceram a base para uma gama de trabalhos de linguistas durante o final dos anos 70 e nos anos 80. O tema principal destes trabalhos é que a lógica das condições de verdade para o significado das linguagens naturais poderia ser salva se ela fosse aumentada com aspectos do significado ligados ao uso da linguagem na comunicação. A proposta de /GRI 75/ é que a conversação segue certas máximas do tipo "Seja relevante!", "Seja organizado!", "Faça sua contribuição tão informativa quanto necessário".

Isto permitiu no início dos anos 80 pensar o significado de cada sentença como um composto híbrido de uma semântica restrita (baseada em condições de verdade) e de uma pragmática (de base ampla). Porém a sequência dos trabalhos mostrou que os fatores pragmáticos jogam um papel importante na sintaxe (e mesmo na fonologia) das linguagens naturais - tal que o componente pragmático funciona como um contexto para os outros componentes da linguagem natural.

O grande problema, do ponto de vista deste trabalho, é que o componente pragmático não tem ainda noções suficientemente claras, nem está devidamente formalizado, pois os trabalhos mais importantes na pragmática (Grice, Searle, Givon) são ainda auto-conscientemente informais.

### 1. 3. História das "grandes idéias" no processamento artificial de linguagem natural

A Idade da Pedra do processamento automático é representada pelos trabalhos iniciais sobre a tradução por máquina: de um modo geral, na falta de uma tradição autônoma própria, o trabalho de tradução de fato tendeu a ficar dentro do paradigma estruturalista dominante na linguística teórica. Este período foi dominado por preocupações predominantemente sintáticas e orientadas para a sentença em si. Os problemas de encontrar um significado para o texto de entrada levou a propostas de soluções ad hoc, por exemplo na assunção de um universo bem restrito para o discurso ou no provimento de rotinas interpretativas de objetivos especiais como parte das entradas lexicais. Deve ser observado, de um modo interessante, que as versões mais sofisticadas destas idéias existem ainda hoje na forma dos processadores que usam frames, dos processadores que usam a semântica procedural e nos processadores "especialistas em palavras".

O campo de trabalho da tradução automática, além de todos as limitações e dificuldades internas citadas antes, sofreu em 1964 um ataque público, através do Relatório Alpac, servindo isto para, de certa forma, desestabilizar seriamente os trabalhos de pesquisa e desenvolvimento na área.

A partir do reconhecimento das limitações e dificuldades internas, algumas das pessoas engajadas no trabalho de tradução automática passaram a manter a posição de que a semântica deveria ter um papel mais dominante no processamento, atuando efetivamente como um condutor e não apenas como polícia ou finalizador do processamento. Assim os primitivos semânticos e a noção de preferência semântica passaram a ser defendidos como instrumentos de processamento, embora elas tenham sido aplicadas neste momento de um modo bem simples. Ao mesmo tempo, a abordagem da estrutura de frase para a descrição sintática foi sujeita, no campo da Linguística Teórica, ao peso das gramáticas transformacionais.

A construção dos processadores artificiais passou a usar então a versão das gramáticas transformacionais conhecida como Teoria Standard (anos 65-70). Os problemas para o processamento das transformações foram de dois tipos: o problema da decidibilidade e o problema da processabilidade. O primeiro problema existiu porque a Teoria Standard não possui a propriedade da decidibilidade formalmente provada. O segundo problema pode ser encontrado nos relatórios dos processadores baseados nas gramáticas transformacionais que foram construídos na década dos 60 e 70 /ZWI 65/ /PET 65/ /PLA 73/ /KAY 73/. Os problemas de processabilidade mais sérios aí referidos foram: o mesmo programa não consegue gerar e analisar sentenças, devido à ordem das transformações; o ordenamento da aplicação das transformações é mal definido dentro do campo das gramáticas transformacionais; a utilização das transformações opcionais é difícil; o tratamento dos ciclos de aplicação das transformações é complicado. Pode ser dito, de um modo simples, que os grandes problemas foram encontrados quase que exclusivamente na implementação do componente transformacional.

Alguns anos atrás, ocorreu um debate vigoroso na revista Cognition sobre as relações entre o trabalho feito dentro do campo da Inteligência Artificial sobre o processamento artificial de linguagem natural e o trabalho feito na Linguística Teórica de inspiração chomskyana /DRE 76/ /WIN 77/ /DRE 77/ /SCH 77/. O debate em si gerou muito pouca luz sobre o tema, devendo ser destacado que, ao final do mesmo, cada grupo se retirou para o seu próprio campo de trabalho.

O reflexo mais importante das elaborações mais recentes das Gramáticas Transformacionais, especificadamente a versão Teoria Standard Extendida, foi a implementação do sistema PARSIFAL /MAR 80/. No entanto, algumas assunções feitas, de modo a tornar o processamento determinístico, de certa forma opacificam o modelo transformacional. Isto ocorre porque a hipótese de um processamento determinístico encontra ainda uma rejeição forte no campo da Linguística Teórica, além de não ser justificável em termos meramente computacionais: deve ser acrescentado aqui que a colocação de que o processamento psicológico de uma sentença é determinístico é também discutível.



Deve ser observado contudo que PARSIFAL falha no processamento de sentenças conhecidas como "garden paths" e ele não fornece as duas leituras de uma sentença ambígua. Além disso, /CHU 80/ mostra que o buffer de PARSIFAL que examina três palavras seguidas e/ou locuções nominais tem dificuldades para processar locuções preposicionais unidas entre si. Outro problema é que o pré-processamento de locuções nominais dá para PARSIFAL uma visão infinita para trás ao nível das palavras, porque as locuções nominais bem formadas podem consistir de muitas palavras. Na prática isto envolve um processamento lento e isto (para usar um argumento caro a Marcus) é psicologicamente implausível /BRI 83/.

O programa PARSIFAL possui um componente (PIDGIN) que implementa a gramática proposta em Lisp. Este processador usa dois estoques temporários para guardar palavras, locuções e constituintes parcialmente construídos. O buffer é uma lista de cinco itens, acessada na base FIFO, dos quais somente três itens podem ser acessados por vez. As palavras entram no buffer pela direita e progridem para a esquerda, e as locuções residem no buffer até que elas sejam ligadas à árvore sintática. Existe também uma pilha de sub-árvores sintáticas construídas parcialmente à qual os itens do buffer podem ser ligados.

Cada regra gramatical tem um conjunto de padrões e um conjunto de ações que são construtoras de estrutura. O conjunto de itens que as regras podem descrever com seus padrões envolve três itens na janela atual e os itens no topo da pilha (o "modo corrente"). As regras são agrupadas em "pacotes" e o processador mantém uma lista dos pacotes correntemente em uso no processamento da sentença.

Ao lado deste processamento sintático, PARSIFAL utiliza um interpretador baseado em casos (frames), tal que as regras semânticas podem realizar certas verificações relativas à colocação do sujeito e dos objetos em torno do verbo. Marcus ainda faz com que os nodos sintáticos possuam "registradores" ligados a eles, contendo itens de dados (por exemplo, ponteiros para outros nodos); ele diz que estes registradores são usados somente para informações semânticas (pg 144), mas na verdade existe alguma confusão entre as noções "registrador", "tipo" e "filho", porque o nome do "tipo" de um nodo é igual ao nome do registrador acessado, e existe um "registrador especial wh" (pg 286) para alguns nodos do tipo S.

Em PIDGIN, existem nodos especiais ditos "traces", tal que um nodo "trace" olha para um nodo locução nominal (sob alguns aspectos): o nodo "trace" possui ponteiros (registradores) para um nodo locução nominal, de modo a simular a idéia de ligação do traço, a qual foi obtida da Teoria Standard Extendida de Chomsky. Este formalismo é usado para fornecer a estrutura das orações relativas (que possuem um buraco de locução nominal). Deve ser observado que o nodo "trace" é marcado tanto como sendo um traço, quanto como sendo uma locução nominal, e as regras sintáticas tratam este nodo como sendo uma locução nominal.

Cada nodo na árvore sintática possui um "tipo", e existem comandos PIDGIN para manipular estas marcações. Não é clara todavia a diferença entre um "tipo" e um "traço", e qual a informação que um "tipo" possui. Marcus todavia menciona (pg 42) que o tipo "é também um de seus traços" (tradução do autor), mas na descrição de PIDGIN eles são comandos que se referem ao tipo de um nodo mais do que mencionam seus traços.

Além disso, um nodo pode ter mais do que um "filho" com o mesmo "tipo": o problema, aparentemente não resolvido por Marcus, é se o "tipo" de um nodo se refere ou não a seu conteúdo gramatical em relação a seu constituinte pai. Outro problema, também relacionado com este, é que as regras de Marcus aceitam uma mudança de "tipo: um item "pp" é colocado como um tipo "predp" (pg 293).

Outro problema no modelo proposto por Marcus diz respeito ao mecanismo referido como "attention-shift": uma regra "attention-shift", para Marcus, possui um padrão único que pode ser comparado a uma posição numa janela, e, no caso da comparação for correta, a janela move-se para a direita. No entanto, não é claro como isto foi implementado, porque Marcus declara que o programa em uso não faz exatamente isso, embora a implementação tenha um resultado semelhante (pg 180-4). Um outro problema é quando e como as regras "attention-shift" são comparadas contra o buffer de entrada: Marcus diz que estas regras são checadas em cada ciclo do comparador de padrões (pg 184), e então, uma vez que uma regra foi comparada e determinou um shift na janela, não existe nada para prevenir que a regra ative de novo o mesmo item (o que acarretaria uma série de "attention-shifts" vacuosos). Poderia ser pensado que as regras "attention-shifts" são avaliadas só quando um item é lido no buffer da lista de entrada, mas isto acarretaria que existe um relacionamento entre o conjunto das regras "attention-shifts" correntemente ativas e o momento no qual um item torna-se viável para induzir um shift; mas, como as regras "attention-shifts" estão em pacotes, elas não são globalmente viáveis todo o tempo. Uma solução melhor seria fazer com que a eliminação dos laços infinitos nestas regras fosse passada para o escritor da gramática, e não construída dentro do interpretador.

Como avaliação final, e não discutindo os aspectos da implementação como foi feito acima, deve ser apontado que as assunções de Marcus não foram ainda provadas, e mesmo o desempenho de uma gramática particular em PARSIFAL (contendo algumas regras ad hoc) sobre um conjunto relativamente restrito de sentenças de teste certamente não substancia a hipótese determinística. No que diz respeito à implementação em si, passando por cima dos problemas apontados acima, uma melhoria a ser considerada é a conexão do fluxo de processamento diretamente a regras de estrutura de frase, tal que exista uma necessidade menor de ativação e desativação explícita de pacotes e que certas rotinas construtoras de estrutura sejam rodadas automaticamente /CHA 83/.

Além disso, o uso de "traços" para nodos é excessivamente cru, não existindo uma abordagem uniforme para a inerência de traços entre nodos (esta inerência é assumida fortemente no modelo desenvolvido neste trabalho). No entanto, o maior problema para PARSIFAL, do ponto de vista prático, é sua dependência ao processamento determinístico.

Nos anos 70 aparece o formalismo conhecido como Rede de Transição Aumentada (ATN), abrindo a Idade de Ferro do processamento automático. Esta idéia de colocar uma gramática como uma rede de transição aparece em /CON 63/ /THO 68/ /BOB 69/, na forma clássica para a literatura em /WOO 70/ e na forma standard em /BAT 78/.

A noção de ATN pode ser derivada da noção de uma Rede de Transição Recursiva (RTN), a qual é uma versão do esquema de reconhecimento para linguagens livres do contexto. Numa RTN, os símbolos da gramática são vistos como nomes de transição. Além disso, o controle é tal que, num processo de encaixamento, o estado corrente da máquina é colocado numa pilha e assim este estado pode ser retomado quando o controle retorna.

Uma ATN é uma RTN com a capacidade de pospor decisões até que o contexto em torno do símbolo seja conhecido: isso é feito pela adição de memória às transições, de modo a estocar e recuperar as condições suplementares de alguma transição anterior. Na nomenclatura ATN, isto envolve adicionar um conjunto de "registradores" e algumas funções pré-definidas para manipulá-los como SETR, GETR, etc.

Com o acréscimo dos "registradores", é perdida a propriedade (verdadeira para as linguagens livres do contexto) de usar automaticamente a construção da árvore sintática junto com o processamento da sentença de entrada. Para construir esta estrutura, uma ATN possui então uma função BUILDQ que se encarrega disto, de modo a permitir a construção da árvore sintagmática (na verdade uma lista) a partir dos literais e dos conteúdos dos "registradores". Outra função necessária é POP, a qual tem por finalidade retornar o valor de algum processo anterior ao que está sendo processado.

O formalismo ATN foi atrativo para o campo dos construtores de processadores porque ele explica o processamento da esquerda-para-direita, encoraja o escritor da gramática a pensar suas regras de um modo procedural (separando o interpretador e a gramática), oferece um modo procedural de ligar a "estrutura superficial" da rede com a "estrutura profunda" de Chomsky (através dos "registradores"). Esta última característica serviu para /WOO 70/ dizer que o formalismo ATN possuía o mesmo poder que as gramáticas transformacionais.

No entanto alguns problemas linguísticos existem no formalismo ATN, pois ele falha em capturar certas generalizações linguísticas: a mais conhecida aqui é a relação de conjunção. Além disso, como sistema programável, é reconhecido que ele sofre da falta de variáveis propriamente escopadas.

A tradição de um processador com preocupação semântica aparece nos anos 70, inaugurando a Idade de Cobre do processamento automático de linguagens naturais. Deve ser enfatizado entretanto que nestes processadores as regularidades sintáticas sempre foram expressas, mesmo com a utilização de modos não convencionais (o convencional sendo baseado em categorias sintáticas). Por outro lado, o modelo linguístico que subjaz a esta tradição é o da Gramática de Casos, através por exemplo da teoria da "análise conceitual" /BIR 79/ e da utilização do conceito de "espera semântica" /RIE 75/.

Deve ser referido ab initium que uma grande variação ocorre nestes sistemas no que diz respeito ao relacionamento entre o processamento semântico e o sintático. De um modo geral, o relacionamento entre o sintático e o semântico no processador pode ser classificado em tres tipos:

- primeiro ocorre a construção da estrutura sintática para a sentença e depois há a conversão desta estrutura numa representação semântica;

- não existe distinção entre sintaxe e semântica e a representação semântica é construída diretamente da sentença de entrada;

- ocorre a passagem de informações da estrutura sintática para a representação semântica durante o processamento, através do uso de estruturas sintáticas parciais.

O primeiro tipo de relacionamento foi usado no sistema de perguntas e respostas LUNAR /WOO 72/. Neste caso, o processador constrói uma estrutura sintática similar à "estrutura profunda" de /CHO 65/ e certas regras de interpretação semântica do tipo se-então convertem esta estrutura em funções que acessam uma base de dados.

O segundo tipo expressa a visão de que a idéia dos anos 60 de construir um processador puramente sintático /THO 68/ é substituída pelo desejo de construir processadores onde a distinção entre a sintaxe e a semântica é perdida; são exemplos os modelos de "dependência conceitual" /SCH 75/ /RIE 75/, onde o processamento é centrado no léxico (cada palavra possui um conjunto de condições associadas a ela).

O terceiro tipo envolve a idéia de passar informações de estruturas sintáticas parciais para a representação semântica. Ela aparece no programa SHRDLU /WIN 72/. A idéia chave aqui é associar uma lista de marcadores semânticos a cada estrutura sintática construída durante o processamento e checar a consistência da estrutura construída.

A idéia desenvolvida neste trabalho é diferente dos tres relacionamentos referidos acima entre a sintaxe e a semântica, pois ela assume que a árvore de representação sintática é construída conjuntamente com a representação semântica, devido a hipótese assumida de uma regra semântica para cada regra sintática (Hipótese Regra-para-Regra), como é clássico para os sistemas lógicos. Isto é mais desenvolvido no interior deste trabalho.

Nos anos 80, um desenvolvimento teórico importante foi o aparecimento da proposta de Gazdar de reavaliação do argumento chomskyano de defesa das gramáticas transformacionais. Chomsky /CHO 57/ /CHO 65/ declarou que as gramáticas livres do contexto eram inadequadas para as linguagens naturais. No entanto deve ser lembrado que nos anos 60 Harman /HAR 63/ já havia criticado este argumento e mesmo proposto uma implementação em COMIT de uma gramática livre do contexto que atingia um fragmento do Inglês idêntico ao atingido por uma gramática transformacional escrita por Chomsky (isto é discutido em detalhe mais adiante). De qualquer forma, é importante observar que a posição de Gazdar hoje não é isolada no campo da Linguística Teórica; além disso, pessoas importantes da área da Computação (como Joshi) aceitam esta idéia. A defesa mais geral da utilização de gramáticas livres de contexto é encontrada hoje na utilização de argumentos de decidibilidade, processabilidade e mesmo capacidade para aprender. Os aspectos mais formais desta proposta que estão ligados à Computação são discutidos no capítulo seguinte. No que diz respeito aos aspectos propriamente linguísticos, deve ser referido que os argumentos linguísticos contra as gramáticas livres de contexto foram baseados em cinco fenômenos:

- a) orações comparativas /CHO 63:378-9/,
- b) expansão decimal do pi /ELS 78:43-4/,
- c) argumento com "respectivamente" /LAN 77:4-5/,
- d) orações subordinadas em Alemão /HUY 76/,
- e) incorporação de substantivos em Mohawk /POS 64/.

A primeira assunção é mostrada ser falsa empiricamente e formalmente /PUL 82/. A segunda assunção depende de uma confusão entre gramática e aritmética. A terceira assunção está baseada num argumento empírico falso e os fatos são mais relevantes no campo da Semântica e da Pragmática /MOS 85/. Os dois últimos argumentos são refutados em /PUL 82/.

A idéia básica do projeto "gramática livre do contexto" é que a captura de generalizações significativas é uma matéria ligada à notação. Porém, as gramáticas livres de contexto, tomadas como uma classe de objetos matemáticos, possuem propriedades independentes das notações que podem ser usadas para defini-las, pois elas determinam um certo conjunto de árvores sintagmáticas, elas estabelecem certas relações de equivalência, etc. Uma analogia com a lógica pode ser feita aqui: a operação de implicação material é verdadeira, por exemplo, se o antecedente é verdadeiro e o conseqüente é verdadeiro, quer a implicação material seja notada como uma flecha, um gancho ou a quinta letra do alfabeto grego, e quer este símbolo seja posicionado no uso prefixado, infixado ou posfixado.

Ao lado disso, deve ser observado que nos últimos 25 anos o "projeto transformacional" desenvolveu uma série de notações linguisticamente úteis e assim algumas delas podem ser usadas no "projeto gramáticas livres do contexto". São exemplos disso os símbolos complexos e as meta-regras, sendo que o formalismo das meta-regras pode ser encontrado no trabalho de van Wijngaarden /VAN 69/ para a sintaxe do Algol. O trabalho aqui apresentado explora estas notações, adaptando-as ao processamento de um interpretador Prolog, deixando as coisas dentro do campo das gramáticas livres de contexto.

Do lado semântico, o trabalho aqui desenvolvido explora o principal desenvolvimento formal no campo da Linguística Teórica - que foi o trabalho de Montague. Deve ser lembrado, quanto a este trabalho formal, que após Montague os linguistas teóricos podem ser divididos no grupo que dispôs a semântica dentro da sintaxe (uma prática que teve seu apogeu na Semântica Gerativa) e no grupo que seguiu o projeto chomskyano; porém a maquinaria sofisticada desenvolvida por Montague tornou viável que a semântica pudesse ser desenvolvida num outro contexto.

Assim Cooper /COO 75/ mostrou que as ambiguidades de escopo dos quantificadores podem ser tratadas semanticamente, sem necessidade de regras de movimento de quantificadores na sintaxe; Dowty /DOW 78/ mostrou que as propriedades semânticas das construções do tipo passivas, dativas com objeto não especificado, "raising" e "equi" podem ser retiradas da semântica, sem necessidade de operações sintáticas de movimento e apagamento de locução nominal; Cooper /COO 79/ mostrou que a coordenação pode ser tratada por teorias semânticas de categorias relacionadas; McCloskey /McC 79/ mostrou que a distinção entre estrutura superficial e estrutura profunda é irrelevante para fornecer uma semântica das orações relativas e das perguntas tipo wh; Klein /KLE 80/ mostrou que os significados das construções comparativas podem ser derivados de suas formas superficiais usando gramáticas livres de contexto.

Com isto parece então haver razões para acreditar que as gramáticas livres de contexto podem suportar teorias semânticas de linguagem natural tão bem como as abordagens transformacionais, com a vantagem de usar uma sintaxe de um nível só. Alguns problemas para a sintaxe da estrutura superficial no entanto já foram detectados, em especial o problema da ligação com pronomes /ENG 82/. Outros problemas são mais específicos a certas linguagens naturais, como a existência de múltiplas dependências do tipo wh em linguagens escandinavas e a ligação de variáveis que isto acarreta /MAL 82/. Para estes casos, uma proposta promissora pode ser a utilização de gramáticas indexadas, definidas em /AHO 88/, pois estas gramáticas podem usar símbolos complexos e intrinsecamente não possuem um limite finito no número de símbolos complexos distintos que elas podem usar. Outra proposta para a resolução destes problemas envolve o uso das gramáticas do tipo TAG, envolvendo o acréscimo à maquinaria livre de contexto de adjunções nas árvores /JOS 82/; as gramáticas TAGs geram linguagens que incluem as linguagens geradas pelas gramáticas livre de contexto, mas que estão incluídas pelas linguagens indexadas. A esse respeito, /JOS 85/ mostra a seguinte relação de inclusão:

CFL < TAL < IL < CSL

onde "<" denota a relação de inclusão imprópria, CFL denota as linguagens geradas pelas gramáticas livres de contexto, TAL denota as linguagens geradas pelas gramáticas de adjunção de árvore, IL denota as linguagens geradas pelas gramáticas indexadas, CSL denota as linguagens geradas pelas gramáticas sensíveis ao contexto.

No que diz respeito propriamente ao tratamento computacional das idéias de Montague, devem ser lembrados aqui uma série de trabalhos recentes, como /FRI 78/ /GUN 80/ /HOB 78/ /JAN 77/ /ROO 81/ /BEA 79/ /EJE 80/ /FOD 83/ /EVA 85/ /KIL 84/ /JOS 82/ /ROB 82/ /SCH 82/ /SCH 82a/.

#### 1.4. ERQLODS e processamento automático de linguagem natural

A identificação de teorias lógicas formais com programas de computador decorre da identificação da computação com o processo controlado de dedução. O atingimento desta última identificação é realizável computacionalmente hoje ao nível de um subconjunto da lógica de predicados de primeira ordem, chamado de cláusulas de Horn (ou cláusulas definidas). A linguagem Prolog é a realização das cláusulas definidas como uma linguagem de programação.

O projeto de construção da linguagem Prolog está intimamente ligado à utilização de Prolog para o processamento automático de linguagem natural, dentro do Groupe d'Intelligence Artificielle da Université de Marseille (Colmerauer, 1975). A utilização da lógica formal para o processamento de linguagem natural pode ser visto contudo sob dois aspectos: a) o uso da lógica como um instrumento para construir um programa que processe a linguagem natural, e b) o uso da lógica para a construção da teoria que descreve a linguagem natural (por exemplo, o uso de cláusulas definidas para descrever a sintaxe e a semântica do Português). Em geral estes dois aspectos tendem a ser confundidos na literatura sobre processadores de linguagem natural em Prolog, mas eles podem ser claramente distintos: sob o segundo aspecto, a gramática de uma linguagem natural é vista como uma lógica e não simplesmente como implementável em lógica (como é colocado no primeiro aspecto).

Uma origem importante do casamento do processamento em lógica com o processamento de linguagem natural está na elaboração dos sistemas-q /CDL 70/. Neste trabalho, Colmerauer discute o problema da reagrupação da totalidade dos elementos de uma sentença por um processador: como cada grupo de elementos de uma locução pode se combinar de modos diferentes com os outros grupos e formar novos grupos, o problema possui uma natureza combinatória. A proposta de Colmerauer aqui é de representar esta variedade de agrupamentos de um modo econômico (grafo orientado), sendo que cada flecha é denominada por uma expressão com parenteses (representando uma árvore). O sistema-q proposto por Colmerauer é em suma um sistema composto por um conjunto de regras livres de contexto que permitem transformar este grafo num outro grafo, correspondendo esta transformação a uma análise ou síntese de uma sentença, ou mesmo outra manipulação formal. Uma das originalidades do sistema-q é seu aspecto reversível: o mesmo sistema-q pode ser usado para descrever uma transformação e também sua transformação inversa. Além disso, é possível encadear vários sistemas-q de modo que cada um pegue como dados os resultados do sistema-q precedente.

Algum tempo depois, Colmerauer /COL 75/ propõe uma axiomatização em lógica de primeira ordem do modo de tratar a associatividade da concatenação sobre cadeias, dispondo em Prolog de facilidades obtidas com os sistemas-q; deve ser observado aqui um salto importante no sentido da utilização de uma maquinaria de dedução automática mais poderosa. A gramática G assim construída (Gramática de Metamorfose) pode ser vista como uma quintupla

$\langle F, Vt, Vn, Vs, - \rangle$

onde F é o conjunto dos símbolos funcionais contendo "." (construtor de árvore) e "nil";

Vt é o vocabulário terminal, sendo que  $Vt \subset HCF$ , onde "C" é o símbolo para "está contido em" e HCF designa o conjunto dos termos sem variáveis (Universo de Herbrand);

Vn é o vocabulário não terminal, sendo que  $Vn \subset HCF$ , que a intersecção entre Vn e Vt é nula e que  $V = Vt \cup Vn$ ;

Vs  $\subset Vn$  e Vs é o vocabulário não terminal de saída;

-> é uma relação de reescrita sobre  $V^*$ , com a restrição de que  $x \rightarrow y$  implica que  $x \neq \text{nil}$ .

A linguagem engendrada por G é o conjunto das cadeias sobre Vt tal que

$L(G) = \{t \text{ pertence a } Vt^* \text{ tal que } s \text{ pertence a } Vs \text{ e } s \xrightarrow{-} t\}$ .

Colmerauer mostra ainda que para cada Gramática de Metamorfose G existe uma gramática G' sob a forma normal, tal que para todo x pertencente a Vn e t pertencente a  $Vt^*$  vale

$x \xrightarrow{-} t$  sse a  $\xrightarrow{-} t$ .

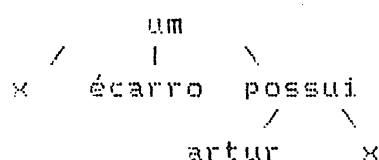
Alguns anos após, Colmerauer /COL 79/ propõe um modelo em lógica para o tratamento de linguagem natural que assume algumas hipóteses. A primeira estabelece que para cada verbo, adjetivo e nome comum é associada uma predicado com n-argumentos do tipo

éfilhode(pedro, paulo).

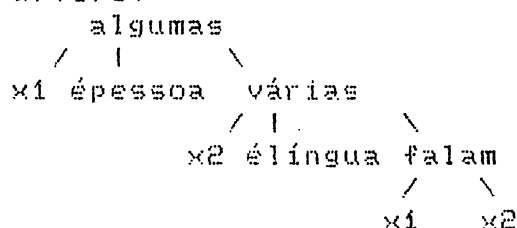
A segunda hipótese assume que a cada artigo corresponde um quantificador q de tres ramos do tipo

$q(x, f1, f2)$

lido como "para qx tal que f1, vale f2" e representado na forma de uma árvore da seguinte forma para o exemplo "Artur possui um carro":



A terceira hipótese assume que a quantificação introduzida pelo artigo do sujeito do verbo domina as quantificações introduzidas pelo complemento do verbo. Assim para a sentença "Algumas pessoas falam várias línguas" existe a árvore:





Ainda na terceira hipótese, numa construção de substantivo comum e seu complemento, Colmerauer postula que a quantificação introduzida pelo artigo do complemento domina aquela que é introduzida pelo artigo do nome. Assim "Pedro dá uma bala a cada criança" tem a seguinte representação:

$cada(x2, \acute{e}crian\c{c}a(x), uma(x1, \acute{e}bala(x1), d\acute{a}(pedro, x1, x2)))$   
 numa notação tipo Prolog.

A terceira hipótese estabelece ainda que num verbo, adjetivo ou substantivo com dois complementos, a quantificação se faz na ordem inversa da ordem de aparição natural deles. Assim "o voto de cada eleitor" é representado como

$cada(x1, \acute{e}eleitor(x1), o\_voto)$ .

sendo que  $o\_voto$  deve ser representado por  $o(x2, voto(x2), \_)$ , onde "\_" está no lugar do predicado da sentença.

A quarta hipótese diz que a negação se traduz pelo operador "não" situado sobre a quantificação do sujeito. Assim a sentença "Todo velho não ganha uma pensão" é representada como:

$n\tilde{a}o(todo(x1, \acute{e}velho(x1), uma(x2, \acute{e}pens\~{a}o(x2), ganha(x1, x2))))$ .

A quinta hipótese assume que toda oração relativa deve ser tratada como um enunciado ordinário: o pronome relativo deve ser substituído por uma variável adequada e todo o resto deve ser ligado à tradução do nome pela conjunção "e". Assim "O gato que varias pessoas amam é amarelo" tem como representação:

$o(x2, e(\acute{e}gato(x2, v\~{a}rias(x1, \acute{e}pessoa(x1), amam(x1, x2))), \acute{e}amarelo(x2)))$ .

Em relação a estas cinco hipóteses, o apresentado neste trabalho tem um tratamento mais homogêneo, seguindo o estabelecido em /MON 73/, referido geralmente como FTQ.

Colmerauer propõe o seguinte esquema geral de tradução para a lógica de primeira ordem ("==>" deve ser lido como "traduz-se em"):

substantivo próprio ==> constante (em Montague é diferente)  
 verbo, substantivo comum, adjetivo ==> símbolo relacional  
 (em Montague é diferente)

$e ==> \&$

$n\tilde{a}o ==> \sim$

quantificador ==> quantificador (x, e1, e2)

e, em relação especificamente aos quantificadores, Colmerauer propõe o seguinte esquema de tradução:

um (x, e1, e2) ==>  $(\exists x) (e1 \& e2)$

algum (x, e1, e2) ==>  $(\forall x) (e1 \rightarrow \sim e2)$

cada (x, e1, e2) ==>  $(\forall x) (e1 \rightarrow e2)$ .

Assim "Pedro odeia todo homem que não navega" possui como tradução inicial (segundo as cinco assunções acima):

$cada(x, e(\acute{e}homem(x), n\tilde{a}o(navega(x))), odeia(pedro, x))$

que é traduzido para a lógica de primeira ordem como

$(\forall x) (\rightarrow (\& (\acute{e}homem(x), \sim(navega(x))), odeia(pedro, x)))$

isto é:  $(\forall x) (\acute{e}homem(x) \& \sim(navega(x)) \rightarrow odeia(pedro, x))$ .

A sexta assunção de Colmerauer estabelece que numa sentença dada, cada enunciado possui tres valores verdade: verdadeiro, falso, indefinido.

A sétima assunção estabelece que as propriedades n-árias introduzidas pelos verbos, nomes comuns e adjetivos aplicam-se sobre conjuntos de indivíduos: uma propriedade sobre um indivíduo é uma propriedade sobre um conjunto de indivíduos de cardinalidade um. Isto também é discutido neste trabalho.

No que se refere mais especificamente a Prolog, Colmerauer /COL 75/ e Kowalski /KOW 79/ colocaram a importante idéia de traduzir o formalismo das gramáticas livres de contexto na lógica de predicados de primeira ordem. O núcleo desta idéia é constituído por um método que se originou no sistema-q de Colmerauer /COL 70/ e que expressa regras livres de contexto através de declarações lógicas de um tipo restrito, conhecidas como cláusulas definidas ou cláusulas de Horn. O problema do reconhecimento (ou do processamento) de uma cadeia de uma linguagem é então transformado no problema de provar que um certo teorema segue dos axiomas de cláusulas definidas que descrevem esta linguagem.

No que diz respeito ao lado prático, Colmerauer desenvolveu a idéia de que uma coleção de cláusulas definidas pode ser considerada como um programa; assim a dedução automática pode exibir todas as características associadas com a computação efetiva, tal que a dedução é aí perseguida num modo dirigido pelo alvo. Uma realização prática deste conceito de programação em lógica foi desenvolvido por Colmerauer na forma da linguagem Prolog, sendo que esta linguagem está baseada num procedimento de prova bastante simples e eficiente /ROU 75/. Assim se a gramática livre de contexto for expressa em cláusulas definidas e executada como um programa Prolog, o programa funciona como um eficiente processador do tipo "top-down" para a linguagem que a gramática livre de contexto descreve.

Colmerauer, no entanto, propõe uma extensão da classe das gramáticas livres de contexto, chamando esta extensão de Gramáticas de Cláusulas Definidas (DCG). A diferença principal entre uma Gramática de Metamorfose e uma DCG é que o lado esquerdo de uma regra DCG é sempre um único símbolo não terminal, enquanto que o lado esquerdo de uma regra de uma Gramática de Metamorfose pode ser uma sequência de não terminais, sujeita a certas restrições. Assim uma DCG pode ser vista como uma gramática livre de contexto com termos lógicos no lugar dos clássicos símbolos gramaticais atômicos; quanto a isso, uma gramática de metamorfose pode ser vista como uma gramática tipo  $\emptyset$  com termos lógicos para os símbolos gramaticais.

As DCGs são um tipo especial das Gramáticas de Metamorfose /COL 75/, porém, na visão de Colmerauer, as DCGs são gramáticas dependentes de contexto, podendo gerar árvores durante o processamento num modo mais restrito devido à estrutura recursiva da gramática (porque estas árvores podem prover uma representação do significado das cadeias) /PER 80/.

O problema da dependência de contexto está ligado a colocação de que, sob uma certa forma, uma DCG pode ser vista como uma gramática que consiste num número infinito de regras livres de contexto; por outro lado, uma DCG estende o conceito de gramática livre de contexto, porque uma DCG pode colocar argumentos em não terminais.

Numa DCG, uma regra livre do contexto é do tipo  
 não\_terminal -> corpo  
 onde "corpo" é uma sequência de um ou mais itens separados por vírgula; cada item é ou um símbolo não terminal ou uma sequência de símbolos terminais. A regra pode ser lida como "o corpo é uma forma possível para uma locução do tipo não\_terminal". No que diz respeito a Prolog, um símbolo não terminal é escrito como um átomo Prolog, enquanto uma sequência de terminais é escrita como uma lista Prolog - onde um terminal pode ser um termo Prolog.

Um exemplo de DCG poderia ser:

```

sentença -> loc_nom, loc_verb.
loc_nom -> det, subst, or_relat.
loc_nom -> subst.
loc_verb -> verbo_transit, loc_nom.
loc_verb -> verbo_intransit.
or_relat -> [que], loc_verb.
or_relat -> [].
det -> [cada].
subst -> [homem].
subst -> [mulher].
subst -> [pedro].
subst -> [maria].
verbo_transit -> [ama].
verbo_intransit -> [vive].

```

a qual gera sentenças do tipo "João ama Maria" e "cada homem ama cada mulher".

Numa notação em cláusulas Prolog isto fica:

```

sentença (S0, S) :- loc_nom (S0, S1), loc_verb (S1, S).
loc_nom (S0, S) :- det (S0, S1), subst (S1, S2),
                  or_relat (S2, S).
loc_nom (S0, S) :- subst (S0, S).
loc_verb (S0, S) :- verbo_transit (S0, S1), loc_nom (S1, S).
loc_verb (S0, S) :- verbo_intransit (S0, S).
or_relat (S0, S) :- conecta (S0, que, S1), loc_verb (S1, S).
or_relat (S0, S).

```

onde cada não terminal da gramática livre de contexto é associado a um predicado com dois argumentos, os quais representam respectivamente o ponto inicial e o ponto final na cadeia da locução denominada pelo não terminal. Por outro lado, a representação dos símbolos terminais envolve o predicado "conecta", onde

```
conecta (S1, T, S2)
```

significa que "o símbolo terminal T existe entre os pontos S1 e S2 na cadeia"; assim as outras regras são:

```

det (S0, S) :- conecta (S0, cada, S).
subst (S0, S) :- conecta (S0, homem, S).
subst (S0, S) :- conecta (S0, mulher, S).
subst (S0, S) :- conecta (S0, pedro, S).
subst (S0, S) :- conecta (S0, maria, S).
verbo_transit (S0, S) :- conecta (S0, ama, S).
verbo_intransit (S0, S) :- conecta (S0, vive, S).

```

É importante observar que a representação de uma gramática livre de contexto na forma de uma cláusula Prolog é independente dos dados, no sentido de que a representação da cadeia a ser processada é conhecida somente pelo predicado "conecta" e o alvo a ser provado. Além disso o predicado "conecta" tem a seguinte forma clausal

```
conecta ([Cabeça | Cauda], Cabeça, Cauda).
```

isto é: a posição na cadeia denominada pela lista que tem como cabeça o que está em Cabeça e como cauda o que está em Cauda, esta posição é conectada pelo símbolo Cabeça à posição da cadeia denominada por Cauda.

O objetivo de provar a sentença original é expresso por:

```
?- sentença ([cada, homem, que, vive, ama, maria], []).
```

O formalismo apresentado pode contudo ser simplificado, tal que as chamadas do predicado "conecta" sejam feitas no tempo de compilação e não no tempo de processamento: a cláusula

```
or_relat (S0, S) :- conecta (S0, que, S1), loc_nom (S1, S).
```

pode ser reescrita como

```
or_relat ([que | S1], S) :- loc_verb (S1, S).
```

Num nível mais geral, pode ser observado que a notação em DCG estende a notação das gramáticas livre de contexto porque cada não terminal é colocado como um termo composto (em oposição a um átomo na gramática livre de contexto), como

```
loc_nom (X, S).
```

Além disso, Colmerauer faz com que no lado direito da regra, em adição aos não terminais e às listas dos terminais, possam ocorrer também sequências de chamadas de procedimentos, escritos entre colchetes, como

```
subst (N) -> [Palavra], (raiz (Palavra, N), é_subst (N)).
```

sendo que esta última generalização pode ser escrita como

```
subst (N, S0, S) :- conecta (S0, Palavra, S),  
raiz (Palavra, N), é_subst (N).
```

No que se refere aos argumentos dos símbolos não terminais, a proposta de Colmerauer é que eles sejam usados para progressivamente construir estruturas sintáticas durante o processo de unificação. Assim a gramática livre de contexto mostrada antes é modificada, tal que para cada locução seja gerada uma árvore. Para isso, para cada regra anterior da forma

```
categoria (argumentos) -> [palavra]
```

é escrita a regra geral

```
categoria (argumentos) -> [palavra],  
(cat (palavra, argumentos)).
```

onde "cat" é definido por

```
cat (palavra, argumentos).
```

para cada palavra do léxico.

As regras modificadas são então:

```
sentença (s(LN, LV)) -> loc_nom (LN), loc_verb (LV).
```

```
loc_nom (ln(Det, Nome, Rel)) -> det (Det), subst (Nome),  
or_relat (Rel).
```

```
loc_nom (ln(Nome)) -> subst (Nome).
```

```
loc_verb (lv(VT, LN)) -> verbo_transit (VT), loc_nom (LN).
```

```
loc_verb (lv(VI)) -> verbo_intransit (VI).
```

```
or_relat (rel(que, LV)) -> [que], loc_verb (LV).
```

```
or_relat (rel(nil)) -> [].
```

```
det (det(W)) -> [W], (é_det (W)).
```

```
subst (subst (W)) -> [W], {é_subst (W)}.
verbo_transit (vt(W)) -> [W], {é_subst (W)}.
verbo_intransit (vi(W)) -> [W], {é_verbo_intransit (W)}.
e além disso são necessárias as cláusulas do tipo
é_det (cada).
é_subst (homem).
é_subst (pedro).
é_subst (maria).
é_verbo_transit (ama).
é_verbo_intransit (vive).
```

Colmerauer propõe aqui que os argumentos dos símbolos não terminais de uma DCG podem ser usados não só para a construção de estruturas sintáticas, mas também para carregar e testar "informação contextual". Na gramática acima, isto seria feito, por exemplo, pela colocação da concordância do número (singular, plural) entre certos determinantes, os substantivos e os verbos. Isto seria feito na gramática seguinte:

```
sentença (s(LN, LV)) -> loc_nom (N, LN), loc_verb (N, LV).
loc_nom (N, ln(Det, Nome, Rel)) -> det (N, Det),
                                   subst (N, Nome),
                                   or_relat (N, Rel).
loc_nom (singular, ln (Nome)) -> subst (Nome).

loc_verb (N, lv(VT, LN)) -> verbo_transit (N, VT),
                             loc_nom (N1, LN).
loc_verb (N, lv(VI)) -> verbo_intransit (N, VI).
or_relat (N, rel(que, LV)) -> [que], loc_verb (N, LV).
or_relat (N, rel(nil)) -> [].
```

e para o dicionário vale

```
det (N, det(W)) -> [W], {é_det (W, N)}.
subst (N, n(Raiz)) -> [W], {é_subst (W, N, Raiz)}.
subst (subst(W)) -> [W], {é_subst (W, N, Raiz)}.
verbo_transit (N, vt(Raiz)) -> [W],
                                {é_verbo_transit (W, N, Raiz)}.
verbo_intransit (N, vi(Raiz)) -> [W],
                                {é_verbo_intransit (W, N, Raiz)}.
```

O dicionário associado é:

```
é_det (cada, singular).
é_subst (homem, singular, homem).
é_subst (pedro, singular, pedro).
é_subst (maria, singular, maria).
é_verbo_transit (ama, singular, amar).
é_verbo_intransit (vive, singular, viver).
```

Nesta proposta, as estruturas (árvores) sintáticas são construídas aos pedaços, onde as partes não especificadas das estruturas correspondem a variáveis. A estrutura contudo deve estar completa quando o processamento terminar: as variáveis (os "buracos" na estrutura) estarão preenchidas pelo mecanismo de unificação.

A posição defendida neste trabalho aceita em linhas gerais o formalismo das DCG, postulando fortemente entretanto que as "dependências contextuais" colocadas por Colmerauer na verdade não tornam a gramática dependente de contexto, porque um conjunto determinado de "traços" sintáticos (ou mesmo semânticos) alocados a uma categoria sintática tradicional implica meramente na constituição de um "símbolo complexo". E este símbolo complexo pode ser visto como uma forma diferente de anotar um símbolo na gramática livre de contexto tradicional: como o número dos traços é finito e como o valor destes traços é finito, para cada regra sintática onde cada símbolo gramatical é um termo lógico em DCG, é possível escrever um número finito de regras livres de contexto tal que para cada símbolo complexo do tipo

```
cat (valor_k_do_traço1, valor_l_do_traço2, ...,
      valorm_do_traçon)
```

de DCG seja escrito o símbolo atômico

```
CATi
```

de uma gramática livre de contexto. Assim uma DCG é equivalente a uma gramática livre de contexto.

Um trabalho sobre processamento de linguagem natural em Prolog, importante quanto a aspectos formais, é o de Pereira /PER 83/. Este trabalho assume a idéia principal das DCGs (de que um símbolo gramatical pode ser expresso por um termo lógico), mas propõe um sistema um pouco diferente de lógica para o tratamento da semântica, chamado de cláusulas definidas de mundo fechado (DCW), cuja maior justificativa está na representação do significado da negação e da quantificação. O problema, para /PER 83/, seria a representação semântica em DCG de sentenças do tipo

Existe algum oceano que banha todos os países europeus?  
a qual teria como tradução semântica

```
resposta (sim) <== (E O) (oceano(O) &
      (V P) (país(P) & europeu(P) -> banha(O, P))).
```

A forma geral das fórmulas que são mapeadas em DCG é

```
literal <== condição
```

onde "condição" pode conter quantificadores e conectivos, todas as variáveis livres em "literal" e em "condição" são universalmente quantificadas, e "<==" é colocado em vez de ":-".

Pereira assume a "condição do mundo fechado", que pode informalmente ser assim expressa: "pode ser inferido não-P se P não for provado". Além disso, os operadores utilizáveis do lado direito de uma cláusula DCW são:

p & q : conjunção (vale se p e q valem)

existe(x, p) : quantificador existencial (vale se existe uma instância y de x tal que o objetivo obtido pela substituição de todas as ocorrências de x em p valem)

\+ p : não provabilidade (vale se p é não provado).

Outros operadores são definidos a partir daí:

todos (x, p ==> q) =def \+ existe(x, p & \+ q)

lido como "não está provado que exista x tal que p valha e q não esteja provado",

numerode (x, p, n) =def existe (s, conjuntode (x, p, s)) &  
card (s, n) & n>0

onde "card (s, n)" diz que a cardinalidade do conjunto finito "s" é "n".

Deve ser observado aí que os operadores "\+" e "conjuntode" são diferentes dos conetivos e quantificadores usuais da lógica de primeira ordem, porque as suas definições são dadas em termos de provabilidade e não em termos de verdade. Isto é consequência da assunção do mundo fechado, e torna o sistema total não monotônico.

O papel dos quantificadores existenciais no lado direito das cláusulas DCW é tal que se "existe" ocorre fora do escopo de um operador não-monotônico, ele pode ser eliminado devido à equivalência

$$p \langle == (Ex) q \equiv (Vx) p \langle == q \equiv p \langle == q$$

que vale para o renomeio de x para evitar problemas com variáveis que proventura existam em p. Por outro lado, se "existe" ocorre dentro de "conjuntode" e de "\+", a subfórmula

existe (x, p)

pode ser substituída por

p (Y1, ..., Yn)

onde Y1, ..., Yn são as variáveis livres em p e o predicado p é definido por

p (Y1, ..., Yn) \langle = p

isto querendo dizer que o único modo de provar o objetivo "existe" é realizar a prova de uma instância de seu escopo p.

Armado desta maquinaria, Pereira /PER 83/ propõe a seguinte tradução dos determinantes segundo o modelo DCW:

um, algum, o: existe (X, R & S)

não : \+ existe (X, R & S)

cada, todo : \+ existe (X, R & \+ S)

os : existe (S, conjuntode (X, R, S) & S)

um, dois, ..., numeral (N) : numerode (X, R & S, N)

e para os pronomes interrogativos vale:

qual : resposta (X) \langle == R & S

quantos : resposta (N) \langle == numerode (X, R & S, N)

onde R é o range da variável e S é o escopo da variável.

Assim /PER 83/ traduz "Alguns pássaros migram" como

existe (X, passaro(X) & migra (X))

e "Não existe rio na Antártica" é traduzido como

\+ existe (X, rio(X) & em(X, antártica);

por outro lado, "Quantos países exportam gasolina?" teria como tradução

resposta (N) \langle == numerode (X, país(X) &

exporta (X, gasolina), N).

Uma contribuição importante de /PER 83/ diz respeito ao tratamento da extraposição de locução nominal, o que é feito dentro do formalismo XG (Gramática de Extraposição). Esta maquinaria é discutido neste trabalho na secção que trata de orações relativas, sendo que aqui um formalismo mais simples (tipo DCG) é proposto.

## 2 QUESTÕES FORMAIS

### 2.1 Gramáticas transformacionais linguagens recursivamente numeráveis reconhecimento e aprendizagem

#### 2.1.1 Gramáticas transformacionais e linguagens r. e.

Putnam /PUT 61/ observa que as teorias antigas das gramáticas transformacionais possuem gramáticas que poderiam gerar linguagens do tipo r. e., porque elas envolvem regras que fazem apagamentos e substituições no curso de uma derivação. Uma adjunção (à esquerda ou à direita) ou uma permutação não podem fazer com que uma gramática gere um conjunto não recursivo. Isto porque se  $ti$  e  $ti+i$  são as saídas de dois níveis sucessivos de uma derivação, tal que  $ti+i$  vem da aplicação de uma regra de permuta ou de uma regra de adjunção sobre  $ti$ , então o número de símbolos terminais de  $ti+i$  será ao menos igual ao número de símbolos terminais de  $ti$ . Esta propriedade (de que as saídas dos níveis sucessivos numa derivação não encurtam em comprimento) é a definição básica que caracteriza a classe das gramáticas dependentes de contexto /HOP 69/. Assim, só a aplicação de regras que reduzem o comprimento (apagamento e substituição) podem fazer com que uma gramática gere uma linguagem não sensível ao contexto e talvez não recursiva, no sentido de r. e.

A Teoria Standard, apresentada em /CHO 65/, assume que a gramática possui um componente da base, consistindo de uma gramática de estrutura de frase com um conjunto não ordenado de regras de reescrita do tipo  $A \rightarrow w / B \_ C$  (isto é: o símbolo  $A$  pode ser reescrito como a cadeia não nula  $w$ , se  $A$  aparece entre  $B$  e  $C$ ). Estas regras permitem derivar uma cadeia de símbolos terminais a partir de uma cadeia inicial  $\#S\#$  e assinalar um ou mais nodos nomeados para cada cadeia derivada, mostrando sua composição como uma sequência de frases.

O conjunto resultante de nodos nomeados (marcadores frasais) serve como entrada a um componente transformacional da gramática, o qual consiste de uma sequência linear de transformações gramaticais, cada uma das quais converte um marcador frasal em outro marcador frasal, na forma sumarizada abaixo.

Uma transformação gramatical consiste de uma condição estrutural e um conjunto de transformações elementares. O marcador frasal é primeiro fatorado numa sequência de sub-partes, tal que a fatoração satisfaça à condição estrutural da transformação; esta fatoração é chamada de análise própria do marcador frasal da transformação. Se o marcador frasal de entrada não possui análise própria, a transformação fornece como saída o próprio marcador frasal de entrada. Porém se existe uma análise própria do marcador frasal de entrada, as transformações elementares são aplicadas a seus fatores.



Cada transformação elementar é ou o apagamento de uma sequência de fatores, ou a substituição de uma sequência de fatores por outro, ou a adjunção de uma sequência de fatores à direita (ou à esquerda) de um outro. Além disso, a condição estrutural e o conjunto de transformações elementares devem cumprir uma Condição de Recuperabilidade dos Apagamentos para ser uma transformação.

Dado um marcador frasal produzido pela base, as transformações são aplicadas ciclicamente às orações do marcador frasal, começando com a oração encaixada mais profunda (isto é a sentença que termina com um Js mais a esquerda no nodo denominado). Aí, de acordo com o princípio do ciclo transformacional, todas as transformações são aplicadas em ordem a esta oração, e, após o término do ciclo, as transformações são aplicadas de novo à oração terminada com o próximo Js à direita. Este processo é repetido até que o ciclo tenha operado sobre a oração mais externa e o marcador frasal resultante é a estrutura superficial se ela preenche a condição de boa formação: ela não pode conter nenhuma ocorrência do símbolo de limite #.

Um marcador frasal P produzido pela base é dito ser a estrutura profunda subjacente à estrutura superficial Q se P pode ser convertido pelo componente transformacional em Q. Além disso, o par (P, Q) é uma descrição estrutural gerada pela gramática, e é assinalada pela gramática a uma sequência de símbolos terminais que resulta do apagamento dos marcadores frasais em Q (sobra só a fronteira da árvore). O conjunto das cadeias de símbolos terminais ao qual a gramática assinala descrições estruturais é dito a linguagem gerada pela gramática, e as cadeias na linguagem são ditas sentenças.

Deve ser observado que nem todo marcador frasal gerado pela base é uma estrutura profunda, pois as regras de base podem introduzir num marcador frasal símbolos de limite # que não podem ser apagados no curso da derivação transformacional. Então é dito que o componente transformacional filtra os marcadores frasais incorretos do conjunto dos marcadores frasais produzidos pela base; as estruturas profundas são precisamente os marcadores frasais produzidos pela base e que não são filtrados pelo componente transformacional.

Uma maior discussão sobre a Teoria Standard das gramáticas transformacionais, bem como as reelaborações posteriores (Teoria Standard Extendida e Teoria Standard Extendida Revisada) está em /MOS 84/ /MOS 84a/ /MOS 84b/ /MOS 84c/ /MOS 84d/ /MOS 84e/ /MOS 84f/ /MOS 84g/ /MOS 84h/ /MOS 84i/ /MOS 84j/. Uma formalização trivial da Teoria Standard está em /GIN 69/.

O desejo de restringir a capacidade das gramáticas transformacionais ao conjunto das linguagens recursivas foi o responsável pela proposta da Condição de Recuperabilidade dos Apagamentos, citada acima, a qual foi adicionada às gramáticas transformacionais nos anos sessenta /CHO 65/. Porém Peters e Ritchie /PET 69/ /PET 73/ mostram que a incorporação desta condição no componente transformacional da gramática de /CHOM 65/, colocando o componente da base com regras sensíveis ao contexto, não altera a capacidade gerativa fraca das gramáticas transformacionais: cada linguagem r. e. pode ser gerada por uma gramática transformacional, e vice-versa. A prova deste teorema pega meramente uma gramática de estrutura de frase sensível ao contexto com uma transformação que apaga um símbolo terminal específico (satisfazendo a condição referida).

Como conclusão, a Condição de Recuperabilidade dos Apagamentos não é suficiente para garantir que as gramáticas transformacionais gerem somente conjuntos recursivos. Mesmo se a base é formada por regras livres de contexto, ou só por duas regras lineares, Peters e Ritchie /PET 73/ mostram que a mesma conclusão se aplica.

Além disso, Lasnik e Kupin /LAS 79/, assumindo a Teoria Standard Extendida /CHO 72/, mostram que a prova de Peters e Ritchie /PET 73/ ainda vale. Por outro lado, Janssen, Kok e Meertens /JAN 77/ conseguem provar que um subconjunto das gramáticas transformacionais (dentro da Teoria Standard) é recursivo, assumindo para realizar esta prova que existe um modo de enumeração de todas as sentenças sobre o alfabeto da linguagem e que a teoria linguística provê um método para decidir se uma dada sentença pode ou não ser gerada por uma dada gramática.

Como conclusão, deve ter ficado claro que o componente transformacional numa gramática transformacional é bastante poderoso. Na prova da identidade entre as linguagens r. e. e as linguagens geradas pelas gramáticas transformacionais, Peters e Ritchie /PET 73/ /PET 71/ mostram que esta identidade vale mesmo se a base quer for composta por regras dependentes de contexto, quer se ela for mais restrita na forma das regras, valendo as seguintes cinco condições de equivalência para um alfabeto  $\{a_1, a_2, \dots, a_n\}$ :

- (i) L é r. e.;
- (ii) L é gerada por uma gramática transformacional com regras da base que sejam livres de contexto;
- (iii) L é gerada por uma gramática transformacional com regras da base que sejam minimamente lineares;
- (iv) L é gerada por uma gramática transformacional com regras da base que sejam lineares só para um lado;
- (v) L é gerada por uma gramática transformacional com regras da base que sejam do tipo  $S \rightarrow S\#, S \rightarrow a_1 \dots a_n b\#$ , onde b denota o símbolo branco e # denota o símbolo de limite, tal que b e # não estão em  $\{a_1, \dots, a_n\}$ .

Isto mostra que o componente transformacional é uma maquinaria tão poderosa que ele pode mapear conjuntos altamente restritos (gerados por uma base muito empobrecida) em linguagens r. e. arbitrárias.

Em /PET 73a/ existe a discussão da classe C de linguagens geradas por uma base com regras livres do contexto e com filtros, onde uma gramática é filtrada localmente se cada derivação transformacional na gramática é tal que, se a saída de um ciclo contém #, então a árvore final da derivação contém também #. As gramáticas que não possuem nenhum predicado de filtro formam uma subclasse de C. Peters e Ritchie /PET 73a/ mostram que:

- (i) C contém todas as linguagens livres de contexto;
- (ii) C não contém nenhuma linguagem sensível ao contexto;
- (iii) C contém uma linguagem não recursiva, tal que ela é gerada por uma gramática que não usa filtros.

Então, mesmo restringindo o uso de filtros deste modo e mesmo usando bases com regras livres de contexto, ainda assim algumas linguagens r. e. não recursivas são geradas por gramáticas transformacionais.

## 2.1.2 Gramáticas transformacionais e reconhecimento de sentenças

Se as gramáticas transformacionais geram conjuntos r. e., então existem cadeias para as quais é impossível decidir se elas são ou não membros de uma certa linguagem natural (gerada pela maquinaria transformacional). Isto implica em que um falante-ouvinte de uma linguagem natural seria incapaz, em princípio, de determinar a gramaticalidade ou não de algumas cadeias de palavras. Porém a idéia intuitiva é que as linguagens naturais não funcionam como conjuntos não recursivos para os falantes-ouvintes.

A assunção aqui é que o falante-ouvinte de uma linguagem natural tem uma só gramática em sua mente e que a gramaticalidade é decidida de um modo análogo ao modo como uma Máquina de Turing decide o problema de se uma cadeia é membro de uma linguagem gerada por ela. Isso quer dizer que a pessoa usaria sua gramática internalizada para gerar uma lista de sentenças gramaticais de sua linguagem e então avaliaria a cadeia em questão contra os itens sucessivos de sua lista. No entanto, poderia ser o caso que os procedimentos mentais humanos usados para avaliar a gramaticalidade fossem estratégias heurísticas não algorítmicas, tal que estas estratégias apenas usassem informação contida na própria cadeia (superficial) ouvida ou lida, sendo que não seria usada a gramática internalizada para gerar o conjunto de sentenças de uma linguagem. Neste caso, seria concebível que houvessem cadeias indecidíveis do ponto de vista do processamento heurístico empregado, mesmo que estas cadeias fossem decidíveis do ponto de vista da gramática internalizada.

Um contra-argumento, lembrado por Levelt /LEV 74/, para esta "solução" é que uma linguagem indecidível (caso das linguagens geradas pelas gramáticas transformacionais) não é aprendível quando a habilidade para aprender é tomada como ligada a um procedimento efetivo. Um outro contra-argumento é que, tudo sendo igual, as linguagens naturais devem ser recursivas e não r. e., porque as primeiras formam uma subclasse própria das últimas (argumento da restritividade). Um último contra-argumento é que a

"solução" apresentada separa reconhecimento de geração.

Uma outra tentativa de "solução" aparece em /PET 73/, a partir da definição de uma função de ciclos (dita  $C_g$ ) para uma gramática  $G$ , tal que  $C_g$  é definida sobre as cadeias superficiais  $x$  da seguinte forma:

Se  $x$  não pertence a  $L(G)$ , então  $C_g(x) = 0$ ; em caso contrário  $C_g(x)$  vale a menor parcela de  $s$  tal que  $x$  tenha alguma árvore inicial com  $s$  ciclos.

Isto é:  $C_g$  assinala à cadeia  $x$  o valor 0 se  $x$  não está na linguagem gerada por  $G$ ; por outro lado, se  $x$  está em  $L(G)$ , são avaliadas todas as árvores iniciais possíveis que  $G$  assinala a  $x$  e é decidido pela que possuir o número total menor de ciclos. Este número é o valor de  $C_g$ .

Definindo  $C_g$  assim, Peters e Ritchie /PET 73/ provam que as seguintes tres condições são equivalentes para uma gramática transformacional  $G$ :

- (i)  $L(G)$  é decidível;
- (ii)  $C_g$  é uma função recursiva;
- (iii)  $C_g$  é limitada por uma função recursiva.

Ou seja: uma  $G$  cuja função cíclica é limitada por uma função recursiva deve ser uma linguagem recursiva (decidível).

Daí eles colocam que uma gramática transformacional escrita para qualquer linguagem natural deve ter uma função cíclica, que seria limitada por uma função exponencial da forma  $K$  elevado ao comprimento de  $x$ , onde  $K$  é uma constante. Como as funções exponenciais são recursivas e como  $C_g$  é limitado por uma função recursiva, as linguagens geradas por esta gramática seriam conjuntos recursivos. Porém existem dois problemas sérios:

(i) não está provado que cada linguagem natural possua uma função cíclica, o que implicaria na possibilidade de uma pessoa poder contar o número de ciclos de todas as árvores iniciais possíveis de cada cadeia que pertence à linguagem;

(ii) não está provado que cada linguagem natural pode ser gerada por alguma gramática transformacional.

Lapointe /LAP 77/ sugere algumas outras restrições sobre o relacionamento das regras de base com os apagamentos feitos pelas transformações, tal que uma estrutura como

$[s \# [s \# \dots [s \# xyz \# ]s \dots \#]s \#]s$

não poderia ser gerada pela base. No caso geral, é colocada a restrição de que, para cada  $x$  pertencente a  $L(G)$ , cada subciclo mínimo numa árvore inicial para  $x$  deve conter ao menos um item lexical. Discutindo sua proposta, Lapointe /LAP 77/ propõe a condição de Petrick /PET 65/, a qual, por seu turno, coloca que os apagamentos na gramática possuem a "propriedade de sobrevivência", tal que, se  $x$  é o domínio de entrada de um ciclo numa derivação e  $y$  é a saída deste ciclo, então  $y$  deve conter mais nodos terminais do que  $z$ , onde  $z$  é qualquer subparte de  $x$ . Lapointe /LAP 77/ então prova que a propriedade de sobrevivência garante que só conjuntos recursivos elementares são gerados pela gramática. Porém existe uma classe de sentenças cujas derivações não possuem a propriedade de sobrevivência. Além disso, o problema de se uma gramática possui ou não esta propriedade de sobrevivência é em geral um problema recursivamente não resolúvel, porque esta propriedade não regula a forma ou a

aplicação de regras, mas olha para toda a derivação sem especificar como as regras particulares devem interagir com estruturas particulares para que a propriedade valha.

Outra proposta /HIR 76/ é que uma regra de apagamento R só poderia apagar x na árvore t sse existisse um termo x<sub>1</sub> na descrição estrutural de R que analisasse uma categoria não cíclica em t, tal que x é irmão de B. Assim ficaria prevenido a aplicação de uma regra de apagamento sobre x em

[c ... [c ... [x ... ]x ... ]c ... ]c

[c ... [c [x ... ]x [c ... ]c ]c ... ]c porque x não possui nenhum irmão não cíclico. Ross /ROS 69/, no entanto, apresenta regras que fogem a isso.

Em relação às gramáticas transformacionais posteriores à Teoria Standard, como a Teoria Standard Extendida /CHO 77/ /CHO 77a/, a convenção X-barra garante que para cada x pertencente a L(G), cada subciclo mínimo numa árvore inicial para x deve conter ao menos um item lexical. Além disso, a estrita ciclicidade é garantida porque LN e S<sub>1</sub> definem ciclos, as árvores iniciais são totalmente especificadas por materiais lexicais e os apagamentos são acompanhados por:

- (i) substituição transformacional de "elementos ocios";
- (ii) interpretação semântica após cada ciclo;
- (iii) apagamento dos elementos ocios após todas as transformações terem se aplicado.

Porém para estas gramáticas gerarem conjuntos recursivos não r. e., algumas condições de apagamento devem existir:

- (i) se x é um termo afetado por uma regra R que analisa a árvore t, então um termo x<sub>1</sub> adjacente a x na descrição estrutural de R deve analisar um nodo não vazio em t;
- (ii) uma regra de apagamento R só pode apagar x na árvore t sse existe um termo x<sub>1</sub> na descrição estrutural de R que analise uma categoria não cíclica B em t, tal que x é um irmão de B /LAP 77/.

O problema é que a Teoria Standard Extendida inclui também inserções lexicais opcionais, implicando isso que existem árvores iniciais não totalmente especificadas por material lexical. Além disso, a teoria inclui transformações que se aplicam ciclicamente de modo estrito, de novo com ciclos S<sub>1</sub> e LN. Outro problema é que a teoria comporta regras de movimento de locução que deixam um traço fonologicamente nulo na posição do material movido. E a teoria envolve a existência de um conjunto de filtros que se aplicam após todas as transformações cíclicas terem se aplicado aos elementos ocios e aos traços fonologicamente vazios. E mais: a teoria ainda comporta regras de apagamento que apagam material lexical, elementos ocios e traços fonologicamente nulos. E, por último, a teoria assume várias condições de aplicação de transformação e de filtros /CHO 82/. Assim não é claro que a atual formulação da Teoria Standard Extendida Revisada gere só linguagens recursivas de um tipo apropriado, pois que ela envolve várias formas de apagamento.

### 2.1.3 Gramáticas transformacionais e capacidade de aprendizagem

Peters e Ritchie /PET 73/ /PET 73a/ demonstraram que cada conjunto recursivamente enumerável tem uma gramática transformacional nos moldes do modelo apresentado em /CHO 65/. Apesar disto, Chomsky /CHO 80/ diz que, dada a finitude do número de gramáticas transformacionais possíveis, não é verdade que a falta de recursividade numa linguagem torne-a não aprendível. Em primeiro lugar, a seguir, é refutada a hipótese da finitude e depois é refutada a falácia do argumento de Chomsky, mesmo aceitando a hipótese da finitude.

Assim, considerando em primeiro lugar a hipótese da finitude, Chomsky /CHO 82/ declarou que as teorias das gramáticas transformacionais seriam restritas o suficiente para permitir um número finito de gramáticas. O mesmo foi colocado em /LAS 81/ e em /WEX 81/. Contra esta hipótese existem contudo vários argumentos:

(i) não existe uma prova de que exista um número finito de gramáticas de linguagens naturais;

(ii) não existe uma definição acabada acerca das representações no componente semântico ("forma lógica") nas teorias das gramáticas transformacionais;

(iii) o número de traços diacríticos nas regras fonológicas não está determinado ainda e são empregados muitas vezes coeficientes numéricos /LAD 71/;

(iv) o conjunto das categorias sintáticas necessariamente não é finito /LAP 80/;

(v) a variação dentro do sistema lexical no que diz respeito a traços fonológicos, morfológicos, sintáticos e semânticos necessariamente não é finita.

(vi) nem todas as linguagens naturais possuem um léxico finito, pois a capacidade de formação de palavras a partir de outras é imensa /LAN 81/.

Pinker /PIN 82/ procurou oferecer, por outro lado, uma prova da finitude, no contexto das gramáticas lexicais funcionais de Kaplan e Bresnan /KAP 82/, procurando calcular a cardinalidade exata de um conjunto de gramáticas em função de certas constantes empíricas. Porém ele apela para a existência de um vocabulário universal não terminal finito.

No que diz respeito à questão da possibilidade de uma entidade (máquina ou homem) aprender uma linguagem natural que possua uma gramática transformacional, através da mera exposição a ela, Chomsky /CHO 80/ assume que, como existiriam 'n' gramáticas possíveis (hipótese da finitude), existiriam um conjunto de sentenças tal que uma investigação sistemática deste conjunto seria suficiente para distinguir as 'n' gramáticas possíveis.

Quanto a isso, se este procedimento de decisão for um procedimento que determine se uma cadeia arbitrária pertence ou não a uma linguagem natural dada, certamente este procedimento existe para cadeias menores do que, por exemplo, cem palavras; o procedimento neste caso é finito e poderia ser a simples inspeção. Porém se este procedimento for um procedimento que gera a gramática correta, aparecem problemas bastante sérios, analisados a seguir.

A idéia é que, para um conjunto finito de gramáticas que geram distintas linguagens, existe um conjunto finito de cadeias do conjunto das sentenças que pode ser usado como chave para decidir se uma gramática é correta para uma dada linguagem. Assim se  $S$  é o menor conjunto de sentenças que, para cada par de gramáticas  $\langle G_1, G_2 \rangle$ , contenha uma cadeia (tão pequena quanto possível) que é gerada por  $G_1$  e não por  $G_2$ , então  $S$  é finito, porque existe um número finito de pares de gramáticas:  $n$  ao quadrado menos  $n$ , para  $n$  gramáticas. Assumindo crucialmente que exista uma tabela que mostre, para cada par de gramáticas, qual a sentença de  $S$  que permite uma escolha entre elas e qual a escolha correta, coletando um corpus finito  $C$  da linguagem  $L$  contendo todas as sentenças de  $k$  palavras ou menos,  $C$  irá incluir a intersecção de  $L$  com  $S$ . Agora tirando de  $C$  todas as sentenças exceto as que são chaves, para cada sentença chave é possível olhar a gramática que ela afasta e então afastar esta gramática. Continuando, todas as gramáticas seriam então afastadas, exceto uma.

O importante é que este "exercício" não possui relevância para o problema da aquisição de uma linguagem natural, por uma máquina ou por um homem. Deve ser observado em primeiro lugar que este procedimento abstrato não faz uso de fatos estruturais sobre a linguagem que está sendo adquirida, pois o processo de aprendizagem não é cumulativo, ou dizendo de outro modo: o algoritmo apresentado não é um procedimento com convergência /PET 81/. Por outro lado, se a classe das gramáticas possíveis fosse finita, as pessoas ou as máquinas não poderiam fazer uso deste procedimento se o conjunto de gramáticas fosse muito grande. Considerando a demonstração de Pinker /PIN 82/ que o número de gramáticas é recursivamente enumerável, que para cada linguagem recursivamente enumerável existe uma gramática transformacional, que para gramáticas fora da classe das recursivas não existe algoritmo de reconhecimento, que para a reconhecimento de gramáticas fora da classe das gramáticas indexadas existe a demanda (mas não necessariamente) de um tempo exponencial, parece haver boas razões para afastar as gramáticas transformacionais.

## 2.2 Motivações teóricas para o formalismo proposto

As gramáticas transformacionais para linguagens naturais empregam uma maquinaria muito complexa, envolvendo símbolos complexos, regras da base, regras de inserção lexical, regras de redundância lexical, regras de movimento, procedimentos de co-indexação, convenções de ligação, filtros locais e não locais, convenções de marcação de caso, traços de percolação, condições sobre aplicação de regras de movimento, etc /MOS 84i/ /MOS 84j/. E as propriedades matemáticas desta maquinaria não são inteiramente conhecidas, como ficou exposto acima. Não é sabido, por exemplo, se a questão da não gramaticalidade com relação a tais gramáticas é decidível. Além disso, existem poucos trabalhos sérios sobre as restrições da classe das gramáticas possíveis para linguagens naturais /JAN 77a/ /LAF 77/ /PEL 80/ /WAS 78/, tal que a questão de se uma variante de gramática transformacional é mais restritiva do que outra ainda não está plenamente respondida.

Este trabalho assume que um modo forte de restringir um componente de uma gramática é simplesmente a eliminação deste componente (como é defendido em /GAZ 82/). Aqui é eliminado totalmente o componente transformacional, porém muitos recursos usados no campo das gramáticas transformacionais permanecem sendo usados, como é exposto a seguir. Uma outra restrição assumida aqui é que a língua portuguesa deve ser um tipo de linguagem livre de contexto e outra restrição adicional é que a gramática da língua portuguesa deve ser tal que gere apenas uma linguagem livre de contexto (o Português). Estas duas últimas restrições devem aqui ser claramente distinguidas, porque uma linguagem livre do contexto pode ser gerada por uma gramática dependente de contexto, ou mesmo por uma gramática de definição recursiva.

Uma primeira motivação meta-teórica para assumir isso é que as propriedades formais das linguagens livres de contexto e das gramáticas livres de contexto são relativamente bem conhecidas /BOD 73/ /AHO 72/ no campo da Teoria das Linguagens Formais.

Uma segunda motivação está ligada à questão da capacidade de aprender uma linguagem natural. Em relação a isso, dados os resultados de Peters e Ritchie /PET 73/, discutidos acima, e a definição técnica de Gold /GOL 67/ sobre a capacidade de aprender, se as linguagens naturais fossem r. e., elas não poderiam ser aprendidas. A suposição clássica feita aqui sobre este aprender é que uma criança nasce com uma gramática  $G_0$ , a qual, em contato com algumas sentenças da linguagem natural, transforma-se em  $G_1$ ; esta  $G_1$ , em contato com mais sentenças desta linguagem natural, transforma-se em  $G_2$ , e assim sucessivamente, ocorrendo um processo de convergência para a gramática  $G$ , que é a gramática estável do adulto. Peters /PET 81/ prova que este procedimento de convergência é recursivamente computável, pertencendo à classe sigma-2 da hierarquia aritmética.



Devido a inaceitabilidade da conclusão de que as linguagens naturais não podem ser aprendidas, uma saída (do tipo extensional) é reduzir a classe permitida das gramáticas; outra saída (do tipo intensional) é considerar uma definição alternativa para a capacidade para aprender.

A primeira saída envolve a hipótese da finitude para as linguagens naturais, hipótese esta que foi refutada acima. Por outro lado, a discussão sobre a saída intensional (isto é: mudar a definição de capacidade para aprender), deve levar em conta um conceito de capacidade para aprender feito em termos da possibilidade de definir um algoritmo que possa mapear conjuntos finitos de pares (cadeia, julgamento de gramaticalidade) em gramáticas.

Em relação a isso, Gold /GOL 67/ mostra que é possível construir este algoritmo para linguagens livres do contexto. Como corolário disso, é possível em princípio construir algoritmos que mapeiam conjuntos finitos de pares (árvore, julgamento de gramaticalidade) em gramáticas, quando as árvores envolvidas são do tipo gramática de estrutura de frase livre do contexto. Isto é verdade porque a dendrolinguagem de uma gramática de estrutura de frase livre do contexto é por si só uma linguagem livre do contexto (isto é discutido noutro local deste trabalho).

No que tange à aprendizagem de linguagens naturais, muito poucas teorias matematicamente precisas existem hoje sobre o processo de aquisição da sintaxe de uma linguagem natural /AND 77/ /POW 78/ /WOL 80/. Entretanto, na literatura sobre Teoria das Linguagens Formais, existe um conjunto relativamente extenso de trabalhos sobre o que é conhecido como "inferência gramatical" a partir de informação sobre exemplos finitos de sentenças de uma linguagem, sendo que muitos destes trabalhos se referem a gramáticas de estrutura de frase para linguagens tipo 2. Assim, por exemplo, existe um algoritmo que realiza o mapeamento da fronteira para a raiz da árvore, usando um autômato de estado finito sobre árvore /FUB 75/. Como, para cada gramática de estrutura de frase livre do contexto, existe um autômato finito de árvore que admite exatamente o conjunto das árvores induzidas pela gramática /THA 73/, dado o autômato de árvore para um conjunto de árvores livres do contexto, é possível construir uma gramática de estrutura de frase livre do contexto para este conjunto de árvores. Deve ser lembrado aqui que este algoritmo não pode ser aplicado a linguagens não livres do contexto, porque estas linguagens não podem ser analisadas por autômato de estado finito sobre árvore da fronteira para a raiz /THA 73/.

A argumentação formal obviamente não pode substituir os estudos sobre a aquisição da sintaxe de linguagens naturais, porém uma base sólida para o desenvolvimento de uma teoria formal sobre a aquisição de uma linguagem natural deve envolver necessariamente o conhecimento exposto acima /PIN 79/.

Uma terceira motivação para gramáticas de estrutura de frase livres do contexto está ligada ao fácil processamento de uma sentença de linguagem natural em humanos /CLA 77/. Isto não pode ser ignorado, porque existem classes concebíveis de linguagens que não são processáveis pelos critérios práticos, devendo ser aqui lembrado que as gramáticas transformacionais não possuem tempo de processamento conhecido. Contudo, se as linguagens naturais são linguagens do tipo 2, isto pode ser o início de uma explicação para o que foi referido sobre a facilidade de processamento, em humanos, das linguagens naturais. Devem ser lembrados os resultados da Teoria das Linguagens Formais que mostram que as sentenças de uma linguagem livre de contexto são provavelmente computáveis num tempo que é no máximo proporcional a menos do que o cubo do comprimento da sentença /VAL 75/. /EAR 70/ mostra que certas linguagens livre de contexto, mesmo ambíguas, são processadas provavelmente num tempo linear.

Por último, deve ser lembrado aqui a questão da ambiguidade para as linguagens tipo 2. Um argumento muitas vezes usado na literatura linguística é que uma linguagem natural ambígua (como o Português) não pode ser uma linguagem livre do contexto determinística /CHO 63/; este argumento é claramente falso, porque a noção de ambiguidade deve ser distinguida da noção de ambiguidade inerente. Assim, uma linguagem inerentemente ambígua é uma linguagem tal que todas as suas gramáticas são ambíguas. Deve ser lembrado, porém, que uma gramática LR não é nunca ambígua e que o conjunto das gramáticas LR caracteriza exatamente o conjunto das linguagens livres do contexto determinísticas. Logo nenhuma linguagem inerentemente ambígua é uma linguagem livre do contexto determinística.

Aqui não é aceito que uma linguagem natural seja inerentemente ambígua; o que é muitas vezes colocado no campo das gramáticas transformacionais é que uma gramática descritivamente adequada para uma linguagem natural (no caso, o Português) deve ter ambiguidade no seu componente semântico. A posição defendida aqui não procura restringir o campo de atuação da ambiguidade, isto porque uma linguagem livre do contexto determinística pode ter uma gramática ambígua, porque todas as linguagens possuem gramáticas ambíguas de algum tipo.

Como comentário final, deve ser respondido aqui ao argumento de que as regras do componente lexical não são exclusivamente livres do contexto /CHO 65/. Em primeiro lugar deve ser lembrado o que foi colocado acima de que não está provado que todas as linguagens naturais possuem um léxico finito. Em segundo lugar, Langendoen /LAN 81/ mostrou que as gramáticas de formação de palavra conhecidas hoje ficam dentro do escopo das gramáticas livres de contexto. A única exceção conhecida até hoje é a gramática de formação de palavras (proposta por Culy /CUL 85/) para a língua Bambara, língua esta que apresenta o fenômeno da reduplicação de afixos; a gramática proposta para isso leva o problema para o campo de uma linguagem indexical. Em terceiro lugar, Koskenniemi /KOS 83/ implementou um transdutor de estado finito (um tipo de autômato de estado finito que inspeciona dois ou mais símbolos por vez) para formalizar

sistemas morfológicos de linguagens naturais; ideias semelhantes ocorreram com Johnson /JOH 70/ e Kaplan e Kay /KAP 81/ para o campo da fonologia de linguagens naturais. No modelo de /KOS 83/, todas as regras da gramática se aplicam simultaneamente e cada regra é compilada num transdutor de estado finito, tal que os vários transdutores funcionam em paralelo. Um campo interessante de investigação pode ser criado aqui, na medida em que for lembrado que as máquinas de estado finito processam árvores de formas particulares (por exemplo, com ramificação uniforme à direita) ou árvores com limites superiores finitos (por exemplo, não mais do que três encaixamentos), segundo o Teorema de Bar-Hillel, Shamir e Ferles /BAR 65/. Langendoen /LAN 81/, Carden /CAR 83/ e Church /CHU 83/ procuraram mostrar que as estruturas de árvores envolvidas nas descrições de palavras de linguagens naturais e mesmo de unidades fonológicas caem em uma destas categorias.

### 2.3 Formalismos do modelo proposto

Para os limites deste trabalho, a proposta de um formalismo está limitada pela adequacidade matemática e pela adequacidade notacional.

A adequacidade matemática envolve obviamente questões ligadas à estrutura lógico-matemática (sintática) interna da linguagem formal proposta, bem como problemas de adequação dos objetos formais caracterizados pela notação, sob a semântica usada, a propriedades manifestadas nos objetos reais modelizados.

Por outro lado, existe a necessidade de uma adequacidade notacional que envolva a corretude da notação para o domínio que ela descreve, pois alguns tipos de notação parecem ajustar-se mais aos fatos de um certo domínio, enquanto que outras notações (equivalentes em algum sentido às primeiras) não. Por exemplo uma notação tipo as usualmente usadas para definir Máquinas de Turing certamente não são tão adequadas para a sintaxe de uma linguagem natural como as notações das gramáticas transformacionais. Um outro exemplo forte é que uma gramática de estrutura de frase livre de contexto para a linguagem finita

$\{w\#w : w \text{ é um } (a, b)^* \text{ e } |w| = 24\}$

requer o uso de mais do que 16 milhões de símbolos, enquanto que uma gramática estratificada requer o uso de menos do que 50 símbolos /BOR 83/.

#### 2.3.1 Símbolos complexos

Harman /HAR 63/ foi um dos primeiros a colocar que gramáticas de estrutura de frase livres do contexto poderiam ter a mesma capacidade gerativa fraca que as gramáticas transformacionais. Para tentar provar isso, Harman propõe uma gramática de estrutura de frase livre do contexto com símbolos complexos, implementada em COMIT, que gera exatamente o conjunto das sentenças gerado por uma gramática transformacional descrita em /CHO 62/.

A idéia central de Harman é que a quantidade de informação da gramática não precisa ficar restrita às categorias gramaticais tradicionais, mas que uma categoria elementar como locução nominal pode possuir vários subscritos, tais como

locução\_nominal/sujeito, participio  
onde a denominação antes da barra indica a categoria elementar e o que está depois da barra indica que a locução nominal é sujeito de um participio.

Ele mostra ainda que isto não altera a capacidade gerativa fraca das gramáticas de estrutura de frase livres do contexto, pois estas gramáticas com símbolos complexos geram somente linguagens livres do contexto. Além disso, assumindo que duas gramáticas que induzem descrições estruturais isomórficas são fortemente equivalentes, o uso dos símbolos complexos não possui nenhum efeito sobre a capacidade gerativa forte das gramáticas de estrutura de frase livres do contexto; isto é assim porque cada gramática que usa símbolos complexos gera um conjunto de árvores que é isomórfico ao conjunto de árvores gerado por alguma gramática de estrutura de frase livre do contexto que não usa símbolos complexos, e o mesmo vale reciprocamente.

Como comentário, deve ser lembrado que a proposta de Harman é bastante avançada na medida em que ele refere (mas não implementa) que uma regra poderia reescrever mais de um símbolo de cada vez (idéia esta originária de /POS 21. Uma outra crítica que ele faz a Chomsky é que o "critério de simplicidade" para a escolha da gramática é bastante subjetivo.

A gramática de Harman tem a forma:

S -> S1/NUMERO\_SG

S -> S1/NUMERO\_PL

S1 -> S2/MODO\_ATIVO

S1 -> S2/MODO\_PASSIVO

S2 -> LN/CASO\_NOM, NAO\_WH + LV/TIPO\_DECL, NAO\_WH

S2 -> LV/TIPO\_QUEST, NAO\_WH + LN/TIPO\_NOM, NAO\_WH

S2 -> WH\_PALAVRA + LV/TIPO\_DECL, NAO\_WH

S2 -> WH\_PALAVRA + LV/TIPO\_DECL, NAO\_WH

LN/WH -> @

LN/NAO\_WH -> DET + NOME

WH\_PALAVRA -> que

VERBO\_FINITIVO/AUX\_MODAL -> pode

VERBO\_FINITIVO/AUX\_NAO, MODO\_ATIVO, NEMBRO\_SG ->

VERBO\_FINITIVO/AUX\_SER, MODO\_ATIVO

VERBO\_INFINITIVO/AUX\_SER -> ser

VERBO\_PART\_PASS/AUX\_SER -> sido

DET -> o

NOME/CASO\_AGENT, NUMERO\_SG -> homem

etc.

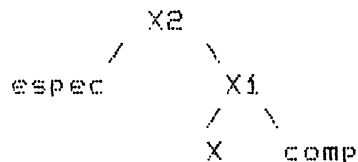
Para a construção de símbolos complexos é assumido aqui o sistema X-barra de Jackendoff /JAC 77/, discutido para o Português em /MOS 84f/ /MOS 85/. Nessa proposta, se X é um símbolo que denota uma categoria lexical, X-uma barra (aqui X1) denota o nodo que imediatamente domina X. O esquema de regra que expressa a existência de material à direita do núcleo de uma locução é:

X1 -> X comp

onde comp (complemento) é uma abreviação para alguma sequência de nodos. O material que precede o núcleo de uma locução categorial maior deve ser introduzido por uma regra do tipo:

X2 -> espec X1

onde X2 designa o nodo que está dois nodos acima do nodo lexical, e espec designa o especificador de X1. Assim



Mosca /MOS 84f/ defende que cada categoria lexical X define um conjunto de categorias sintáticas X1, X2, ..., Xk (as supercategorias de X), as quais são relacionadas por regras da forma

Xn -> ... Xn-1 ...

onde Xn-1 é o núcleo de Xn, o que segue em linhas gerais /JAC 77/. Além disso, as regras da gramática envolvem traços sintáticos complexos do tipo [a F1, b F2, ...], onde a vale + e b vale -, e Fi vale uma categoria sintática. /MOS 84f/ assume, seguindo /JAC 77/, o seguinte modelo para definição das categorias lexicais principais:

N = [+ N, + V]

V = [- N, + V]

Adj = [+ N, - V]

Prep = [- N, - V]

A noção de símbolo complexo, como mostrado em /MOS 85/, ultrapassa os limites da sintaxe X-barra, na medida em que um símbolo complexo é formado da seguinte forma: um predicado denota a categoria sintática elementar, o primeiro argumento é um número inteiro referente ao número de barras no sistema X-barra, e os outros argumentos referem-se a subcategorias e informações morfo-sintáticas, como por exemplo

n(2, nome\_proprio, masculino, singular)

v(1, primeira\_pessoa, singular, transitivo\_direto).

e como consequência, as estruturas sintáticas são árvores de estrutura de frase, cujos nodos são denominados por categorias sintáticas do tipo símbolos complexos.

Formalmente, as categorias sintáticas passam a ser definidas como conjuntos de especificações de traços, tal que uma especificação de traço é um par ordenado consistindo de um traço (por exemplo, caso) e o valor do traço, que pode ser atômico (por exemplo, agentivo) ou mesmo uma categoria sintática (no caso de uma categoria dependente de outra). Assim sendo, a estrutura interna das categorias fica restrita a simples restrições de co-ocorrências de traços, as quais são condições booleanas simples que restringem as combinações permitidas de especificação de traços. Os traços que possuem como valores categorias sintáticas foram discutidos acima; além disso pode ser suposto um traço "buraco" que assume como valor o nome da categoria sintática ausente na construção; para os efeitos de notação, deve ser observado que a próxima seção que trata do fenômeno das dependências não ligadas, utiliza  $x/y$  em vez de  $x(\dots, y, \dots)$ .

Assumindo a teoria das funções parciais sobre as categorias sintáticas, é então possível estabelecer a seguinte definição de extensão (como uma generalização da relação "é um superconjunto de"):

- uma categoria  $C_2$  é uma extensão de uma categoria  $C_1$  sse:
- (i) cada especificação de traços de  $C_1$  cujo valor é um átomo está também em  $C_2$ ;
  - (ii) para cada especificação de traço de  $C_1$  cujo valor é uma categoria, o valor do traço de  $C_2$  é uma extensão do valor em  $C_1$ .

Esta definição recursiva diz que todas as especificações cujos valores são átomos numa categoria estão também em todas as extensões desta categoria. Ela afirma ainda que se uma categoria especifica um valor  $v$  para algum traço com valor de categoria, então qualquer extensão desta categoria especifica um valor para o mesmo traço tal que este valor é uma extensão de  $v$ . Deve ser observado que uma extensão de uma categoria  $C$  pode conter, por esta definição, uma especificação para um traço cujo valor é uma categoria tal que este valor não esteja especificado em  $C$ . Além disso deve ser notado que esta definição de extensão permite estabelecer um ordenamento parcial entre as categorias.

Outra operação sobre categorias é a operação de unificação (análoga a de união entre conjuntos, exceto que o resultado deve ser uma função). A definição é:

- a unificação de um conjunto de categorias  $K$  é a menor categoria que é uma extensão de cada membro de  $K$ , se esta categoria existe; em caso contrário, a unificação de  $K$  é indefinida.

No caso de resultado indefinido, a especificação dos traços das categorias de  $K$  é contraditória. Além disso, a noção de unificação é equivalente da noção de limite mínimo superior na Teoria dos Reticulados.

Outra operação é a de generalização, que é análoga à operação de intersecção entre conjuntos. Sua definição é:

a generalização de um conjunto K de categorias é a menor categoria que contém:

- (i) a intersecção das categorias em K, e
- (ii) o conjunto das especificações de traços cujos valores são categorias, onde cada um destes valores é a generalização do conjunto dos valores assinalados ao traço pelas categorias em K.

É assumido aqui uma convenção sobre instanciação de traços, referida como Convenção de Traços do Núcleo em /MOS 85/. Ela diz que:

numa regra de forma  $D \rightarrow \dots d \dots$ , onde d é o núcleo de D, d pega todos os traços associados com D.

Por outro lado, a definição de núcleo de uma locução é a seguinte, nos moldes do sistema X-barras (onde as categorias são analisáveis com os traços +/- N e +/- V):

numa regra da forma  $D \rightarrow \dots d \dots$ , d é o núcleo de D sse:

- (i) D tem i barras e é analisado como p N e q V;
- (ii) d tem j barras e é analisado como p N e q V;
- (iii)  $j \leq i$ ; (" $\leq$ " : igual ou menor)
- (iv) não existe  $d_1$  (diferente de d), introduzido por esta regra, tal que  $d_1$  tem n barras e é analisado como p N e q V, onde  $n \leq j$ .

O uso da noção de símbolo complexo permite escrever uma regra do tipo:

$v(2, a) \rightarrow v(1, a), n(3, b)$

que segue a Convenção de Traço de Núcleo, e regras do tipo:

$n(3, b) \rightarrow det(3, b), n(2, b)$

onde b varia sobre certas combinações permitidas de traços de concordância, de modo a estabelecer concordância destes traços entre as categorias. Deve ser observado que o emprego dos símbolos complexos nestas regras não afeta o tipo da gramática, porque cada esquema de regra mostrado acima pode ser expandido num conjunto finito de regras de estrutura de frase livres do contexto, tal que a gramática resultante seja simplesmente uma gramática de estrutura de frase livre do contexto.

### 2.3.2 Dependência não ligada e categorias derivadas

Na abordagem estruturalista dos anos quarenta e cinquenta /HAR 51/ /FRI 52/, foi estabelecido uma heurística para responder a pergunta se dois constituintes como

- (1) o rato de Mimi
- (2) o gato que Pedro alimenta

pertencem a mesma categoria; esta heurística envolve a comparação da distribuição dos dois constituintes numa certa quantidade de posições sintáticas e o julgamento de se as expressões resultantes são gramaticais ou não. Se a ocorrência de um leva à gramaticalidade da expressão e a ocorrência do outro acarreta a não gramaticalidade, então eles não pertencem a mesma categoria.

Os dois exemplos acima, apesar de parecerem possuir muito pouco em comum, apresentam a seguinte distribuição:

----- bateu em Maria	[1, 2]
Sultão pegou -----	[1, 2]
Sultão deu um osso para ---	[1, 2]
O melhor animal é -----	[1, 2]
Paulo mostrou a Maria ---	[1, 2]
Sultão pegou Malhado ----	[*1, *2]
Sultão ---- Malhado no jardim	[*1, *2]
Sultão é muito ---- agradável	[*1, *2]

Uma pergunta poderia ser feita sobre se  
 (3) o gato Felix destruiu este canil  
 (4) o gato Felix destruiu

pertencem a mesma categoria sintática ou não. A resposta, nos moldes estruturalistas e também nos moldes gerativistas, é certamente não, porque não faz sentido para eles responder a pergunta de qual a categoria que "O gato Felix destruiu" pertence. Porém em sentenças como "Este canil, o gato Felix destruiu" e "O gato Felix destruiu, e Pedro reparou, este canil", a construção "o gato Felix destruiu" ocorre. Analisando a distribuição de "O gato Felix destruiu este canil" e "O gato Felix destruiu", ela aparece como abaixo:

Maria reconstruiu a casa e -----	[3, ?4]
Se ----- então Sultão ficou sem casa	[3, ?4]
Maria reconstruiu mais canis do que --	[*3, 4]
Maria reconstruiu a casa que ---	[*3, 4]
Pedro perguntou que canil ----	[*3, 4]

Porém os defensores das gramáticas transformacionais acreditam que os constituintes (3) e (4) são membros de uma mesma categoria sintática (sentença). /HAR 62/ todavia mantém que estes dois constituintes não pertencem a mesma categoria sintática; para ele (3) pertence à categoria sentença e (4) pertence a uma categoria diferente chamada sentença sem uma locução nominal. A idéia exposta aqui é em muito uma ramificação da idéia de Harris, envolvendo uma redefinição da noção de categoria sintática.

Assim, se  $V_n$  é o conjunto dos símbolos categoriais básicos (informalmente: o conjunto de todos os símbolos não terminais usados rotineiramente), é possível definir um conjunto  $D(V_n)$  das categorias derivadas como:

$D(V_n) = \{A/B \text{ tal que } A, B \text{ pertencem a } V_n\}$ .

Assim, supondo contrafactualmente que S e LN sejam as únicas categorias básicas então o conjunto das categorias derivadas seria {S/S, S/LN, LN/LN, LN/S}.

A notação A/B deve ser vista aqui como uma reminiscência da notação empregada nas gramáticas categoriais. Ela deve ser aqui assim interpretada: um nodo chamado A/B domina sub-árvores idênticas àquelas que podem ser dominadas por A, exceto que em alguma sub-árvore canônica do tipo A/B ocorre um nodo da forma B/B dominando um elemento (fonologicamente) nulo ou uma cadeia vazia. Intuitivamente o nodo A/B denomina um nodo do tipo A que possui um "buraco" /MOS 82/ do tipo B. Assim S/LN é uma sentença que possui uma LN em algum local fora de S/LN.



A estas categorias "barradas" /MOS 85/ se aplica um conjunto de regras, tal que estas regras expandem uma categoria derivada como a correspondente regra básica faz para com a categoria básica, exceto que uma das categorias dominadas fica pareada com uma marca que indica "buraco". A utilidade destas categorias na árvore sintagmática é fazer então com que a informação do buraco possa ser carregada para cima na árvore.

Assim seja  $G$  o conjunto das regras básicas (o conjunto das regras que não envolvem dependências ligadas). Para uma categoria sintática  $B$ , existe algum subconjunto do conjunto dos símbolos não terminais  $V_n$ , tal que cada elemento deste subconjunto possa dominar  $B$  de acordo com as regras de  $B$ . Seja este subconjunto denotado por  $V_b$  tal que  $V_b$  pertença a  $V_n$ .

Desta forma para uma categoria  $B$ , onde  $B$  pertence a  $V_n$ , é possível definir um conjunto finito  $D(B, G)$  de regras derivadas ("regras com categorias barradas" de IMOS 85i) como:

$$D(B, G) = \{ \langle A/B \rangle L_1 \dots L_i/B \dots L_n \mid \text{tal que} \\ \langle A \rangle L_1 \dots L_i \dots L_n \text{ pertence a } G \ \& \ i = \langle i \rangle n \\ \& \ A, L_i \text{ pertencem a } V_b \},$$

onde  $\langle X \rangle K \dots M$  diz que o nodo  $X$  domina (ou seja, é analisado) como  $K \dots M$ .

Um exemplo de aplicação acima pode ser feito sobre a gramática  $G$  abaixo:

$\{ \langle S \rangle LN LV \}, \{ \langle LV \rangle V LV_1 \}, \{ \langle LV \rangle V LN \}, \{ \langle LP \rangle P LN \}, \\ \{ \langle S_1 \rangle \text{ que } S \}, \{ \langle LV \rangle V S_1 \}, \{ \langle LV \rangle V LN LP \}, \{ \langle LN \rangle LN LP \}$

e então  $D(LN, G)$  teria como componentes:

$$\{ \langle S/LN \rangle LN/LN LV \}, \{ \langle S/LN \rangle LN LV/LN \}, \\ \{ \langle LV/LN \rangle V LV_1/LN \}, \{ \langle LV/LN \rangle V LN/LN \}, \{ \langle LV/LN \rangle V S_1/LN \}, \\ \{ \langle LV/LN \rangle V LN/LN LP \}, \dots \}$$

Estas regras derivadas não possuem nenhuma propriedade especial do ponto de vista semântico ou lexical. Isto é: todas as regras derivadas podem possuir o mesmo número da regra básica correspondente (o ordenamento das regras é parcial), as mesmas propriedades de subcategorização e as mesmas traduções semânticas que as regras básicas das quais elas derivam.

É possível agora formalizar algumas restrições sobre a geração de regras derivadas (ditas "restrições de ilhas" em /MOS 85/, tal que certos tipos de regras derivadas não sejam empregados por uma certa linguagem. Assim por exemplo no Português não é possível fazer extraposição de uma locução nominal que está diretamente dominada por uma locução preposicional, isto é:

$$* \langle LP/LN \rangle \dots \}$$

Outro exemplo vem de Ross /ROS 67/ que apresenta a Condição de Ramificação a Esquerda, pela qual nenhuma  $LN$  que seja um constituinte de uma grande  $LN$  pode ser reordenada para fora desta  $LN$  grande. Isto pode ser colocado como:

$$* \langle LN/LN \rangle LN/LN \dots \}$$

Uma versão generalizada desta condição, oriunda de /GAZ 82/ e discutida em /MOS 85/, estabelece que todas as dependências de LN no ramo esquerdo de um constituinte devem ser bloqueadas; assim:

\* [ $\langle A/B \rangle T/B \dots$  ]

onde A e T são nomes de nodos e  $B = LN$ .

Esta restrição pode ser incorporada à definição de  $D(B, G)$  pela estipulação de que:

" ( $i = 1 \ \& \ B = LN$ ).

Em adição às regras derivadas, é necessário possuir regras de ligação (que podem ser pensadas como um subconjunto das regras básicas) para eliminar categorias. Elas possuem a seguinte forma geral:

$[\langle A/A \rangle t]$  onde A pertence a  $V_n$  e t é um elemento vazio e a interpretação é

$h\langle A \rangle$  onde  $h\langle a \rangle$  é uma variável distinguida que atinge as denotações do tipo A.

### 2.3.3 Definição e propriedades formais de meta-regras

As gramáticas possuem regras que geram/interpretam sentenças; as meta-regras são declarações sobre estas regras, na medida em que são procedimentos que podem ser usados para gerar regras da gramática ou para codificar certas relações entre elas (como redundância em suas formas). Formalmente, uma meta-regra é um par ordenado de regras  $\langle A, B \rangle$ , as vezes escrito  $A \Rightarrow B$ , e interpretado como: se a gramática possui uma regra da forma A, esta gramática possui a regra da forma B.

É possível, a partir disso, fornecer uma definição indutiva da gramática pela definição de um subconjunto das regras (dito subconjunto das regras básicas) e uma lista de meta-regras; o conjunto total das regras é derivado pela aplicação das meta-regras às regras básicas e então recursivamente aos produtos de tais aplicações. Para os limites do que aqui é discutido, é assumido que as regras básicas são todas livres de contexto.

Uma gramática com regras básicas livres de contexto e com meta-regras pode ser então formalizada como uma sêxtupla

$G = \langle V_t, V_n, S, \rightarrow, \Rightarrow, V_v \rangle$

onde

- (i)  $V_t$  denota o conjunto finito dos símbolos terminais;
- (ii)  $V_n$  denota o conjunto finito dos símbolos não terminais, tal que  $V_t$  e  $V_n$  são disjuntos;
- (iii) S é um  $V_n$  especial;
- (iv)  $V_v$  denota um conjunto finito de variáveis essenciais, tal que  $V_v$  é disjunto com  $[V_t \cup V_n]$ ;
- (v)  $\rightarrow$  denota um subconjunto finito de  $V_n \times (V_t \cup V_n)^*$ ;
- (vi)  $\Rightarrow$  denota um subconjunto finito de  $\langle V_n \times (V_t \cup V_n \cup V_v)^* \rangle \times (V_n \times (V_t \cup V_n \cup V_v)^*)$ , tal que, se  $\langle A, a \rangle, \langle B, b \rangle$  pertence a  $\Rightarrow$ , então cada membro de  $V_v$  ocorre um número igual de vezes em a e b, e tem ao menos uma ocorrência em cada um.

Por definição

$\langle\langle A, a \rangle = \rangle \langle B, b \rangle$  sse, para algum par  $\langle\langle A, x \rangle, \langle B, y \rangle$  pertencente a  $=$ , existe um inteiro positivo  $n$ , existem as cadeias  $w_{11}, \dots, w_{1n}, w_{21}, \dots, w_{2n}, \dots, w_{m1}, \dots, w_{mn}$  pertencentes a  $(V_t \cup V_n)^*$  e existem os símbolos  $u_1, \dots, u_{n-1}, v_1, \dots, v_{n-1}$  pertencentes a  $V_v$ , e valem as igualdades:

- (i)  $w_i = w_j$  quando  $u_i = v_j$ , para  $i = \langle i, j \langle n$ ;
- (ii)  $x = w_{11} u_1 \dots w_{1n-1} u_{n-1} w_{1n}$ ;
- (iii)  $a = w_{11} w_{21} \dots w_{1n-1} w_{2n-1} w_{1n}$ ;
- (iv)  $y = w_{31} v_1 \dots w_{3n-1} v_{n-1} w_{3n}$ ;
- (v)  $b = w_{31} w_{51} \dots w_{3n-1} w_{5n-1} w_{3n}$ .

Por outro lado, sejam as definições:

- (i)  $--G--$  denota o conjunto de todas as regras da gramática com meta-regras  $G$ ;
- (ii)  $|--G--$  denota uma relação sobre  $(V_t \cup V_n)^*$  que vale sobre um par  $\langle x, y \rangle$  sse  $T, U, V$  pertencem a  $(V_t \cup V_n)^*$  e  $A$  pertence a  $V_n$  tal que  $x = TAU, y = TVU$  e  $\langle A, U \rangle$  pertence ao conjunto  $--G--$ ;
- (iii)  $|--G--$  é ancestral (fechamento transitivo e reflexivo) de  $--G--$ ;
- (iv)  $\{x \text{ pertencente a } V_t^* \text{ tal que } S |--G-- x\}$  é a linguagem gerada por  $G$ .

Um problema teórico colocado pelo acréscimo de meta-regras a uma lista finita de regras (básicas) livres de contexto é se a linguagem gerada é ou não livre de contexto. Para responder a isso é necessário começar observando que uma meta-regra utiliza procedimentos que atingem um grande conjunto de regras. Isto é feito pelo emprego de variáveis nas meta-regras. Estas variáveis são de duas categorias. A primeira consiste de "variáveis abreviatórias" que se aplicam a conjuntos finitos de valores admissíveis. A utilidade destas variáveis é a de permitir a expressão de generalizações linguisticamente importantes. Do lado formal, elas poderiam ser eliminadas da gramática, na medida em que cada meta-regra com estas variáveis pode ser substituída por um conjunto finito de meta-regras obtido pela instanciação dos valores admissíveis para estas "variáveis abreviatórias". Assim elas não afetam nem o conjunto das cadeias geradas nem o conjunto das árvores sintagmáticas que podem ser assinaladas às cadeias geradas pelas gramáticas com meta-regras.

O segundo tipo de variáveis, ditas "variáveis essenciais", está ligado a cadeias de símbolos terminais ou não terminais. Assim, por exemplo a meta-regra

$$LV(a) \rightarrow XAY \quad = \quad LV(a) \rightarrow X A(a, \text{si\_próprio}) Y$$

onde  $K(1)$  indica que a categoria  $K$  possui os traços  $1$ ,

possui as variáveis essenciais  $X$  e  $Y$  e uma variável abreviatória  $A$ ; esta meta-regra seria tal que  $A$  atingisse o conjunto das locuções nominais e das locuções preposicionais, e "a" atingisse traços de especificação de concordância. Esta meta-regra assim geraria regras de  $LV$  com constituintes reflexivizados e controlados pelo sujeito, na medida em que a gramática possuía a regra

$$S(a) \rightarrow LN(a) LV(a)$$

e além disso a variável a sobre traços de concordância determinaria também o pronome reflexivo adequado.

O uso destas variáveis essenciais pode não só criar conjuntos infinitos de regras, mas também pode gerar linguagens não livres do contexto, como é sugerido mas não provado em /GAZ 82/. O teorema a seguir mostra que uma gramática com meta-regras com variáveis essenciais gera uma linguagem recursivamente enumerável (r. e.).

**Teorema:** Cada linguagem gerada por uma gramática com meta-regras é um conjunto recursivamente enumerável de cadeias.

**Prova:** O conjunto  $--G-->$  de todas as regras de G é r. e. Começando do conjunto finito das regras básicas e aplicando iterativamente o conjunto finito das meta-regras de G, como  $--G-->$  é uma r. e., pode ser deduzido que  $|-G-->$  é também r. e. Logo  $|-*G-->$  é também r. e., porque as relações r. e. são fechadas sob a operação que pega seus ancestrais. Daí então  $\{x \text{ tal que } S |-*G-> x\}$  é um conjunto r. e. de cadeias, porque ele é definível por quantificação existencial de uma relação r. e. (por exemplo  $(\exists y)[y=S \ \& \ y |-*G-> x]$ ). Logo  $L(G) = \bigcup_{t*} \text{intersecção } \{y \text{ tal que } S |-*G-> x\}$  é r. e., porque  $\bigcup_{t*}$  é r. e. e a classe dos conjuntos r. e. é fechada sob a intersecção.

Existe, por outro lado, uma interação entre variáveis essenciais e não terminais ditos "símbolos sem utilidade" na teoria das linguagens formais. Deve ser lembrado aqui que o teorema que prova que cada linguagem livre do contexto não vazia é gerada por uma gramática livre do contexto sem estes símbolos explica o nome "símbolos sem utilidade" /HOP 79/. Porém a denominação "símbolo sem utilidade" é enganosa quando aplicada a gramáticas com meta-regras, porque estes símbolos não são agora elimináveis. Isto é assim porque nas gramáticas com meta-regras pode ser o caso de R1 (regra básica) gerar R2 com um símbolo sem utilidade e R2 gerar R3 sem símbolo sem utilidade. Uma denominação mais adequada para estes símbolos sem utilidade talvez seja a de "categorias fantasmas".

Um exemplo de categoria fantasma poderia ser LVT (locução verbal transitiva), a qual poderia ser usada para introduzir verbos como "dizer" em sentenças como "Maria disse para João que poderia chover", através das regras:

LVT  $\rightarrow$  V S1

S1  $\rightarrow$  que S,

tal que a categoria fantasma LVT não aparecesse em nenhum lado direito de qualquer regra da gramática. Assim nenhum não terminal poderia ser usado na derivação da cadeia LVT, porém uma meta-regra como:

LVT  $\rightarrow$  V X  $\Rightarrow$  LV  $\rightarrow$  V LP X

poderia ser suposta, de modo a derivar uma regra ativa para LV a partir de LVT. Por outro lado, estas regras com LVT poderiam

também ser usadas como entrada de uma outra meta-regra que gerasse passivas com LV.

Esse exemplo mostra que as categorias fantasmas (símbolos sem utilidade) não são necessariamente sem utilidade, nem são necessariamente não terminais como o nome "categoria fantasma" sugere.

Por outro lado, uma gramática com meta-regras com uma única variável essencial gera uma linguagem recursivamente enumerável. Isto pode ser mostrado pelo exemplo abaixo. Assim seja a gramática G com uma única regra básica

$S \rightarrow abc$

e com as três meta-regras seguintes

$S \rightarrow aX \quad \Rightarrow \quad S \rightarrow Xaa$

$S \rightarrow bX \quad \Rightarrow \quad S \rightarrow Xbb$

$S \rightarrow cX \quad \Rightarrow \quad S \rightarrow Xcc$

onde X é a única variável essencial. As sentenças desta linguagem são exatamente os lados direitos das regras que compõem esta gramática; a aplicação recursiva das meta-regras gera a linguagem

$\{abc, bcaa, caabb, aabbcc, abbccaa, bbccaaa, bccaaaabb, \dots\}$

a qual não é livre do contexto, porque o resultado de sua intersecção com o conjunto regular  $a^*b^*c^*$  gera

$\{a \exp 2n \ b \exp 2n \ c \exp 2n \text{ tal que } n \geq 0\}$

que é um subconjunto infinito de

$\{a \exp n \ b \exp n \ c \exp n \text{ tal que } n \geq 0\}$

que não é uma linguagem livre do contexto. Como a classe das linguagens livres do contexto é fechada sob intersecção com a classe das linguagens regulares /HOP 79/, então a linguagem gerada pela gramática G não é livre do contexto.

Além disso, a linguagem  $\{a \exp n \ b \exp n \ c \exp n\}$  pode ser gerada pela gramática G1 abaixo, com a seguinte regra básica

$S \rightarrow AabcA$

e as seguintes meta-regras com uma única variável essencial X:

$S \rightarrow AaaX \quad \Rightarrow \quad S \rightarrow AaXa$

$S \rightarrow AabX \quad \Rightarrow \quad S \rightarrow AbXac$

$S \rightarrow AbbX \quad \Rightarrow \quad S \rightarrow AbXb$

$S \rightarrow AbcX \quad \Rightarrow \quad S \rightarrow AcXbb$

$S \rightarrow AccX \quad \Rightarrow \quad S \rightarrow AcXc$

$S \rightarrow AcAX \quad \Rightarrow \quad S \rightarrow AXccA$

$S \rightarrow AXA \quad \Rightarrow \quad S \rightarrow X$

devendo ser observado que A nunca ocorre do lado esquerdo de uma regra e logo é um símbolo fantasma.

Pensando então naquilo que é conhecido como a "hierarquia de Chomsky", é claro que as gramáticas com regras básicas livres do contexto e meta-regras geram as linguagens regulares e as linguagens livres do contexto; elas geram também algumas linguagens r. e. e, pelos exemplos discutidos no transcórper deste trabalho, elas geram também algumas linguagens sensíveis ao contexto. Restaria estudar se elas geram todas as linguagens indexadas ou não, todas as sensíveis ao contexto ou não, todas as recursivas ou não e todas as r. e. ou não. Uma hipótese a ser verificada é se as gramáticas com meta-regras distribuem-se fora do padrão "hierarquia", isto é: se as respostas às questões acima forem do tipo "geram parte das linguagens indexadas, parte das sensíveis ao contexto, parte das recursivas, parte das r. e."

Para os limites deste trabalho, uma conclusão importante a tirar é que é necessário redefinir a forma das meta-regras, no sentido de limitar seu poder gerador. Uma tentativa seria fazer com que uma meta-regra só pudesse se aplicar a regras básicas, e não a uma regra derivada de uma regra básica, fazendo isso com que o conjunto das regras da gramática fique finito. Outra tentativa seria eliminar as variáveis essenciais, de modo que a classe das linguagens atingidas pelas gramáticas com meta-regras não fosse as linguagens r. e.. As duas propostas são aceitas por este trabalho.

#### 2.3.4 Interpretação das regras de estrutura de frase

Chomsky /CHO 75/ interpreta uma regra de estrutura de frase como uma regra de reescrita que mapeia cadeias em cadeias. Assim  $S \rightarrow LN LV$  é uma função que mapeia cadeias da forma  $X-S-Y$  em cadeias da forma  $X-LN-LV-Y$ . O conjunto das cadeias derivadas compostas só por terminais é o conjunto das cadeias que aparecem no mapeamento do símbolo inicial  $S$  para as cadeias de terminais. Com algumas restrições muito simples, isto permite definir uma árvore segundo a derivação /MOS 84a/.

Um outro modo de interpretar uma regra de estrutura de frase existe e aparece em /CHO 65:215): a idéia é a construção de um sistema onde a reescrita de um símbolo é determinada não só pelo símbolo que está sendo reescrito, mas também pelo símbolo da categoria dominante (este é o conceito de admissão de um símbolo por uma regra). Assim um nodo denominado  $S$  numa árvore é admitido por uma regra tipo  $S \rightarrow LN LV$  sse este nodo domina imediatamente e exaustivamente dois nodos: o da esquerda sendo denominado  $LN$  e o da direita sendo denominado  $LV$ . Uma árvore é então analisada por uma gramática sse cada nodo não terminal é admitido por uma regra da gramática. Sob esta interpretação, as regras de estrutura de frase são condições de boa formação sobre árvores; como consequência, nenhuma noção de derivação existe e não há razão para estabelecer uma ordem para as regras.

Em sua análise do componente da base de uma gramática transformacional, McCawley /McC 68a/ sugere que o papel das regras sensíveis ao contexto no componente da base pode ser substituído por condições de admissibilidade de nodos nas árvores da base. O componente da base poderia então ser visto como um conjunto de árvores com labels, satisfazendo certas condições.

Sampson /SAM 79/ refere que a abordagem de regras de reescrita restringe a forma das regras de modo a impedir apagamentos como por exemplo:

$LV \rightarrow V LN LV$   
 $LN [+PRO, +NOM] \rightarrow e$  (e = símbolo vazio)  
 $COMP \rightarrow e$  (COMP = complementizador)

Uma outra consequência é a proibição dos apagamentos, porque as gramáticas de estrutura de frase sensíveis ao contexto com regras de apagamento, sob a interpretação da reescrita, geram sistema de reescrita não restritos (linguagens r. e.). Mas isto não vale se a interpretação tipo condição de admissibilidade

de nodo for usada sobre gramáticas de estrutura de frase livres ou dependentes do contexto /BAR 65/. O único efeito sobre a capacidade gerativa fraca de acrescentar produções vazias nas gramáticas de estrutura de frase (livres de contexto ou sensíveis ao contexto), sob a interpretação de condição de admissibilidade de nodos, é que pode ser gerada uma linguagem consistindo somente de cadeias vazias.

Peters e Ritchie /PET 69/, por seu turno, provaram que o poder gerativo fraco das gramáticas sensíveis ao contexto é o mesmo das gramáticas livres de contexto, se as regras são usadas, nos dois tipos de gramáticas, como condições de admissibilidade de nodos: elas só podem analisar linguagens livres do contexto. Então, num sentido algébrico bem definido, a condição de admissibilidade de nodos das gramáticas de estrutura de frase é mais restritiva que a interpretação como reescrita. Como não existe até hoje razão para acreditar que uma linguagem natural é estritamente sensível ao contexto (sensível ao contexto, mas não livre do contexto), pode ser assumido aqui a interpretação mais restrita das gramáticas de estrutura de frase: a interpretação da admissibilidade de nodo.

Joshi e Levy /JOS 77/ estenderam este resultado e mostraram que, se as condições de admissibilidade de nodos incluem condições booleanas de predicados de análise (condição contextual horizontal na árvore) e predicados de dominação (condição contextual vertical), a capacidade gerativa fraca das gramáticas sensíveis ao contexto é a mesma das gramáticas livres do contexto. Nesta abordagem, uma análise própria de uma árvore é uma camada sobre a árvore, tal que o conjunto das análises próprias de uma árvore  $t$ , denotado por  $Pt$ , é definido como segue:

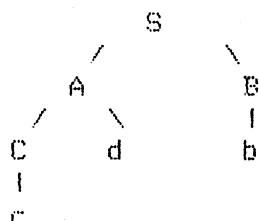
(i) se  $t$  é vazia, então  $Pt = \emptyset$ ;

(ii) se  $t = [ \langle A \rangle A_0 A_1 \dots A_n ]$ , então

$$Pt = \{ A \} \cup Pt_0 \cdot Pt_1 \dots Pt_n$$

onde  $t_0, t_1, \dots, t_n$  são árvores e  $\cdot$  denota concatenação (de conjuntos).

Assim se  $t$  é a árvore



então  $Pt = \{ S, AB, Ab, CdB, Cdb, cdB, cdb \}$ .

Uma regra (dita de restrição local) no modelo de /JOS 80/ possui a forma  $A \rightarrow w / C$ , onde  $C$  é a combinação booleana de predicados de análise própria e de predicados de dominação. O tempo de reconhecimento das linguagens geradas por estas regras é proporcional ao cubo do comprimento da cadeia, segundo um algoritmo proposto em Buneman e Levy /BUN 78/. Neste modelo é possível propor regras do tipo

$S_1 \rightarrow \text{COMP } S$

$S_1 \rightarrow S / \sim \text{DOM}(S_1, S_S)$

tal que a segunda regra assegura que nenhum  $S_1$  imediatamente dominado por  $S$  aparece sem o complementizador COMP. Joshi e Levy /JOS 77/ ainda mostram que os filtros locais de Chomsky e Lasnik /CHO 77b/ podem ser mapeados em restrições locais sobre regras e isto não acarreta nenhum efeito sobre a capacidade gerativa fraca destas gramáticas: elas permanecem só sendo capazes de gerar linguagens livres do contexto.

Deve ainda ser lembrado que a computabilidade do tipo  $n$  elevado a 3 existe para linguagens livres do contexto mesmo quando a gramática de estrutura de frase (que definiu o conjunto das árvores) emprega produções vazias /AHO 72/; Joshi e Levy /JOS 77/ mostram que isto também vale quando ela incorpora restrições locais.

## 2.4 Anexo: um comentário sobre as dendrolinguagens

Alguns desenvolvimentos na Teoria dos Autômatos colocaram uma extensão no domínio de definição dos autômatos, concebendo o processamento de árvores (em vez do tradicional processamento de cadeias). Isto permitiu enriquecer e simplificar a matéria, do mesmo modo que a ênfase sobre cadeias enriqueceu os estudos dos autômatos de números naturais. Outra consequência foi que partes importantes da Linguística Matemática puderam ser formalizadas nestes autômatos de árvore.

As propriedades algébricas dos autômatos finitos sobre árvores foram já extensivamente estudados /BRA 68/ /MEZ 67/ /THA 67/ /THA 68/ /THA 73/ /ARB 68/. A noção de conjunto reconhecível é central nestes trabalhos. Um esquema de avaliação finita (um autômato finito) é usado aí sobre uma árvore de entrada: o autômato analisa a árvore da base para o topo (raiz), classificando a árvore como aceitável ou não. O conjunto reconhecível associado com o autômato é o conjunto de todas as árvores aceitáveis. Rabin /RAB 67/, por outro lado, faz o autômato trabalhar do topo para as folhas, estudando autômatos de geração de linguagens.

Uma descrição indutiva das árvores é dada em /THA 68/ a partir da idéia de um alfabeto ordenado, isto é, intuitivamente, do conjunto das denominações que podem ocorrer numa árvore. Assim, em primeiro lugar, um nodo com  $k$  descendentes é denominado com um símbolo de ordenamento (rank)  $k$ . Se  $\langle S, r \rangle$  é um alfabeto ordenado, onde  $S$  é finito e  $[r: S \rightarrow \text{números naturais}]$  por definição, então  $S_n = (r \exp -1) \{n\}$ . Desta forma os  $S$ -termos (isto é: as árvores) são definidos recursivamente, tal que o conjunto  $F\langle \emptyset, S \rangle$  (os  $S$ -termos constantes) é o menor conjunto de cadeias tal que:

- (i)  $S_0 = C F\langle \emptyset, S \rangle$ ; (" $=C$ " : igual ou está contido)
- (ii) se  $t_0, \dots, t_{n-1}$  pertencem a  $F\langle \emptyset, S \rangle$  e  $s$  pertence a  $S_n$ , então  $s(t_0, \dots, t_{n-1})$  pertence a  $F\langle \emptyset, S \rangle$ .



A definição de um termo assim feita garante que uma única leitura possa ser feita para cada termo; por outro lado, as árvores assim geradas são finitas, com nomes nos nodos, ordenadas e com raiz, tal que não é possível que dois nodos com mesmo nome possuam um número diferente de ramos. Esta definição é equivalente a das gramáticas livres do contexto e ordenadas, onde se  $A \rightarrow w$  e se  $A \rightarrow x$ , então  $|w| = |x|$ .

Se  $S$  é um alfabeto ordenado, uma  $S$ -dendrolinguagem é um subconjunto de  $F(\emptyset, S)$ ; os conjuntos das árvores de derivação são  $S$ -dendrolinguagens, chamadas de dendrolinguagens de derivação /THA 67/. Rounds /ROU 70/ prova que cada dendrolinguagem de derivação é reconhecível e que cada dendrolinguagem reconhecível pode ser obtida de uma dendrolinguagem a partir de uma função (projeção) que renomeia nodos de árvore. Como corolário, ele mostra que o conjunto das fronteiras das árvores reconhecíveis é uma linguagem livre do contexto, e que cada linguagem livre do contexto pode ser obtida deste modo.

Thatcher /THA 69/ e Aho e Ullman /AHO 68a/, considerando a idéia de transformação de Chomsky, acrescentam a esse modelo a capacidade de realizar transformações determinísticas sob estado finito, a partir do estabelecimento de regras de estado finito que mapeiam uma árvore de entrada numa árvore de saída e que trabalham primeiro no nodo do topo da árvore de entrada e depois no segundo nível, e assim sucessivamente até os nodos inferiores. Um exemplo informal de transformação seria:

$$\begin{array}{ccc} \begin{array}{c} d + \\ / \quad \backslash \\ x_0 \quad x_1 \end{array} & \longrightarrow & \begin{array}{c} + \\ / \quad \backslash \\ dx_0 \quad dx_1 \end{array} \end{array}$$

ou seja

$$(d, +(x_0, x_1)) \rightarrow +((d, x_0), (d, x_1)).$$

Rounds /ROU 70/ mostra que o domínio do sistema que usa transformações parciais e determinísticas é um conjunto reconhecível (efetivamente obtível) e que as transformações totais e determinísticas são efetivamente fechadas sob composição (mas não as parciais). Ele mostra porém, e isto é importante, que as transformações determinísticas não preservam sempre o conjunto como reconhecível. Rounds /ROU 70/ amplia a capacidade das transformações de modo a considerar modelos não determinísticos de mapeamento (uma árvore de entrada gera um conjunto finito de árvores de saída, onde um nodo é transformado em cada estágio); as linguagens geradas por este modelo no entanto são recursivas, no caso das transformações serem lineares. Contudo /ROU 70/ mostra que estas "transformações" não conseguem refletir as transformações propostas nas gramáticas transformacionais.

### 3 UMA INTRODUÇÃO À GRAMÁTICA DE MONTAGUE

#### 3.1 Características gerais da sintaxe e da semântica

Cada teoria séria da linguagem natural possui alguma formulação explícita sobre o relacionamento entre expressões da linguagem natural e expressões de alguma linguagem interpretada, a qual, de algum modo, mostra a significação das expressões da linguagem natural. Este relacionamento pode ser chamado de relação de tradução; no que se refere a isso, algumas teorias propõem mesmo a existência de uma linguagem intermediária não ambígua entre a linguagem natural e a interpretada.

As teorias diferem sobre como esta relação é especificada. Por exemplo, na teoria exposta no *Aspects* /CHO 65/, foi assumido que a relação deve ser definida como uma relação entre a estrutura profunda da sintaxe e a representação semântica; vários problemas importantes contudo levaram à proposta de aplicação da relação a uma estrutura intermediária: a estrutura superficial com traços, gerando-se então um nível de representação chamado de "forma lógica" /CHO 75/. O comum a estas duas abordagens é a assunção de que as regras de tradução são definidas inicialmente sobre estruturas sintáticas de algum tipo. Esta assunção pode ser chamada de Hipótese Configuracional, pela qual todas as regras de tradução possuem a forma:

Dada uma estrutura sob tal forma, traduza esta estrutura numa expressão, na linguagem interpretada, sob tal forma.

Em contraste com essa Hipótese Configuracional, pode ser assumida a visão de que a sintaxe e a tradução podem seguir juntas. Ou seja: a sintaxe constrói uma estrutura sintática pela aplicação de regras sintáticas que operam sobre estruturas sintáticas, enquanto que para cada regra sintática existe uma única regra de tradução especificando a tradução da expressão resultante como uma função da tradução de suas partes. Esta assunção pode ser chamada de Hipótese Regra-para-Regra sobre a relação de tradução. Esta segunda hipótese é característica da Gramática de Montague, assim como de várias outras abordagens que caem dentro dos limites do que pode ser chamado de "Gramáticas Recursivas", isto é: gramáticas que na sua definição empregam elementos básicos da linguagem e um procedimento recursivo para geração de novos elementos desta linguagem. No caso da abordagem de Montague, o PTQ /MON 73/ descreve a relação de tradução como uma operação sobre uma estrutura, a chamada "árvore de análise". Porém, como a árvore de análise é um registro da derivação sintática, é fácil conceber neste fragmento exemplos de um procedimento que simultaneamente constrói uma estrutura sintática bem formada e sua tradução respectiva.

Uma formulação forte da Hipótese Regra-para-Regra assume que:

- para cada regra sintática existe uma única regra de tradução que especifica a tradução da saída da regra sintática como uma função das traduções das entradas desta regra;
- todas as regras possuem aplicação estritamente local na derivação, isto é: nenhuma regra tem acesso a estágios anteriores ou posteriores na derivação;

- as regras sintáticas aplicam-se a estruturas sintáticas e as regras de tradução aplicam-se a traduções; nenhum tipo de regra tem acesso às representações do outro tipo;

- as traduções das entradas de uma regra sintática devem ficar intactas na tradução de saída, exceto para mudança de variáveis (com a finalidade de evitar ligação acidental de variáveis).

A primeira assunção é simplesmente a afirmação da concepção regra-para-regra na relação de tradução. A segunda assunção é consequência desta hipótese. A terceira assunção requer um tipo de autonomia de um modo diferente do discutido em /CHO 75/. A quarta assunção é um tipo de condição de recuperabilidade, comparando com as propostas das versões antigas das gramáticas transformacionais, na relação de tradução; ela afasta por exemplo regras que mapeiam algum tipo de tradução numa constante (incluindo a nula).

Além disso, a teoria geral da semântica e a conexão entre a sintaxe e a semântica, em Montague, é tal que uma linguagem ambígua, como o Inglês ou o Português, é interpretada pela associação, a cada uma das expressões bem formadas desta linguagem natural, de um conjunto não nulo de significados; aqui é referido conjunto porque uma expressão do Português, por exemplo, pode ter mais do que uma interpretação. Este processo é feito pela geração de uma linguagem desambiguada, onde cada expressão bem formada corresponde a exatamente um significado; esta linguagem desambiguada é relacionada de algum modo às expressões bem formadas da linguagem natural ambígua. A relação entre a linguagem desambiguada e a linguagem natural fica representada mais diretamente através das árvores de análise.

### 3.2 Conceitos básicos da semântica de Montague

Se de um lado a tradição gerativo-transformacional teve sérios problemas na tratabilidade da semântica, os filósofos e lógicos tenderam a ver a semântica como perfeitamente tratável mas de um modo inadaptável a sentenças das linguagens naturais. Contra estas duas posições, Montague /MON 70/ declarou que não existia uma diferença teórica importante entre as linguagens formais e as naturais. Sua proposta envolveu trabalhar com noções tipo valores verdade, teoria dos modelos e mundos possíveis. Estas tres idéias são discutidas a seguir.

A proposta envolvendo valores verdade é a de que conhecer o significado é conhecer em que mundo esta sentença é verdadeira, isto é: dar o significado de uma sentença é especificar suas condições de verdade, ou seja, dar as condições necessárias e suficientes para a verdade desta sentença. Isto envolve adotar uma teoria de correspondência para a verdade, como fez Tarski.

Esta teoria concebe a diferença entre sentenças (entidades linguísticas) e estados de existência (configurações de objetos no mundo). Afastando já um possível paradoxo inicial, pode ser constatado que uma certa linguagem natural (a língua portuguesa, por exemplo) pode ser escolhida como linguagem objeto e também como a meta-linguagem, no sentido em que a língua portuguesa pode ser usada para falar sobre a língua portuguesa; isto porém não implica, como é evidente, em nenhum vício de circularidade. Retornando agora ao problema da correspondência entre sentenças e estados de existência, como existem muitas sentenças não sinônimas do Português (é bem provável aqui um número infinito), esta proposta de semântica deve especificar um número muito grande (talvez infinito) de condições de verdade. Isto sugere então usar uma maquinaria recursiva.

Aceitando a idéia de definições recursivas, o problema fica então o de especificar os pedaços que devem entrar na especificação recursiva dos estados de existência. Como primeira aproximação, pode ser assumido que o mundo contém vários tipos de objetos (entidades) e existem relações entre estes objetos e certas propriedades.

Uma estratégia para tratar com estes problemas é estabelecer que as regras que especificam condições de verdade trabalham junto com as regras sintáticas. Esta é uma característica da semântica baseada em condições de verdade, sendo conhecida como Princípio de Composicionalidade (ou Princípio de Frege /FRE 52/). A idéia é que uma sentença é composta por um número (potencialmente finito) de palavras ou morfemas - ditas expressões básicas - que são combinadas segundo regras sintáticas em expressões sucessivamente maiores. Cada regra sintática pode ser vista como uma certa declaração pela qual certas expressões de entrada combinam-se de um certo modo para produzirem uma expressão de saída.

Supondo que cada expressão básica da sintaxe é associada com algo no mundo (uma entidade, uma propriedade, uma relação), isto forma então a base da recursividade que opera tanto na sintaxe quanto na semântica. O Princípio de Frege estabelece quanto a isso que o significado de um todo é uma função do significado das partes deste todo e do modo de combinação destas partes; as partes aqui referem-se aos constituintes sintáticos das expressões em questão.

Em relação à segunda idéia, semântica baseada na teoria dos modelos, ela envolve a construção de modelos matemáticos abstratos de coisas do mundo, de modo a elaborar valores semânticos de expressões numa linguagem objeto. Vista desta forma, a teoria dos modelos é um método para expressar o programa da semântica baseada em condições de verdade.

A idéia importante aqui da teoria dos modelos é que o significado das expressões e as correlações entre as expressões e os significados podem ser atingidos pela investigação de como o significado das expressões complexas é relacionado ao significado das expressões mais simples com as quais estas expressões complexas são construídas.

O modelo começa por especificar quais os tipos de coisas que existem no mundo, assumindo uma certa ontologia; a partir daí ele especifica uma interpretação da linguagem objeto. Feito isto, pode ser percebido que certas expressões básicas do Português podem receber uma interpretação invariante de modelo para modelo (por exemplo, as expressões "e", "ou", "não", "cada"); por outro lado, uma certa classe de sentenças do Português é logicamente verdadeira, enquanto outra classe é logicamente falsa. Além disso, existem sentenças ambíguas como "Cada homem ama uma mulher". A teoria deve ser tal então que comporte todos estes casos.

Em relação à terceira idéia, a semântica de mundos possíveis, a proposta é que o significado de uma sentença depende não só do mundo como ele de fato é, mas sim do mundo como ele pode ser ou pode ter sido. Isto envolve então a noção de mundos possíveis: a noção de mundo possível envolve a especificação completa de como as coisas são ou podem ser, envolvendo isso todos os pormenores finos semanticamente relevantes. Por outro lado, um mundo possível em particular contém algo que pode afetar o valor verdade de alguma sentença. Nesta abordagem então a noção de estados de existência é uma noção mais restrita que a noção de mundo possível, porque um estado de existência pode se referir só a alguns estados de um mundo. Mas em geral uma sentença em particular é sobre uma pequena parte de todos os mundos possíveis (o mais comum é mesmo um só). Isto implica em que, em geral, um estado de existência torna uma sentença verdadeira em relação ao que é comum a todos aqueles mundos nos quais a sentença é verdadeira.

Com estas bases epistemológicas, é possível especificar uma sintaxe e uma semântica de algum subconjunto de uma linguagem natural, o que é feito a seguir.

### 3.3 Um exemplo simples

Antes de apresentar o formalismo para um subconjunto do Português, é preciso esclarecer que as idéias gerais do formalismo sintático proposto por Montague tem origem no formalismo conhecido como "gramática categorial", proposto por Ajdukiewicz em 1935 /AJD 67/ e elaborado por Bar-Hillel /BAR 64/. Uma gramática categorial é uma gramática de estrutura de frase que possui apenas regras livres do contexto, um pequeno número de categorias básicas (a proposta inicial envolveu apenas S para sentença, N para nome próprio, C para nome comum), um número infinito de categorias derivadas, um léxico finito dividido em categorias. Um exemplo de especificação para esta gramática, nos moldes do que foi originalmente proposto por Ajdukiewicz, poderia ser o seguinte:

- "uma" pertence a  $(S/(S/C))/C$
- "porca" pertence a C
- "ama" pertence a  $(S/C)/C$
- "petúnia" pertence a C
- "amarela" pertence a C/C

assumindo que um elemento da categoria A/B se concatena com um elemento da categoria B e gera um elemento da categoria A; a especificação acima formaria por exemplo "Uma porca ama uma petúnia amarela", entre outras. Este tipo de formulação de regra sintática foi contudo ampliado por Montague.

Em primeiro lugar, a proposta de Montague deve ser vista dentro do campo da tradição lógica. Neste campo é costume especificar a sintaxe não por uma gramática de estrutura de frase, mas por uma definição recursiva de expressões bem formadas. Isto é feito pela explicitação de uma lista de expressões básicas divididas em várias categorias; há aqui contudo uma correspondência bastante clara para com os símbolos terminais e as categorias lexicais de uma gramática de estrutura de frase. Além disso é feita a explicitação de um conjunto de regras de formação que estabelecem como as expressões de várias categorias são combinadas em outras expressões mais complexas. Estas regras costumam ter a seguinte forma: dadas as expressões  $a, b, c, \dots, m$  pertencentes respectivamente às categorias  $C_a, C_b, C_c, \dots, C_m$ , estas expressões podem ser combinadas de um certo modo (estabelecido pela regra) de modo a gerarem uma expressão da categoria  $C_v$ . Isto corresponde à uma regra em gramática de frase do tipo  $C_v$  é composta pela concatenação de  $C_a, C_b, C_c, \dots, C_m$  de um modo estabelecido pela regra.

Assim seja uma linguagem  $L_0$ , cuja sintaxe possui o seguinte conjunto de expressões básicas e o seguinte conjunto de categorias sintáticas:

Categorias	Expressões básicas
Nomes	$d, n, j, m$
Predicados c/1 argumento	$M, B$
Predicados c/2 argumentos	$K, L$

As regras de formação serão de dois tipos: as regras que combinam predicados com um número apropriado de nomes para produção de sentenças atômicas e regras que formam uma sentença de uma ou mais outras sentenças. Elas são as seguintes:

- se  $x$  é um predicado com um argumento e  $y$  é um nome, então  $x(y)$  é uma sentença;
- se  $x$  é um predicado com dois argumentos e  $y$  e  $z$  são nomes, então  $x(y, z)$  é uma sentença;
- se  $x$  é uma sentença, então  $\sim x$  é uma sentença;
- se  $x$  e  $y$  são sentenças, então  $[x \ \& \ y]$  é uma sentença;
- se  $x$  e  $y$  são sentenças, então  $[x \ \vee \ y]$  é uma sentença;
- se  $x$  e  $y$  são sentenças, então  $[x \ \rightarrow \ y]$  é uma sentença;
- se  $x$  e  $y$  são sentenças, então  $[x \ \leftrightarrow \ y]$  é uma sentença.

Neste caso, a língua portuguesa é a meta-linguagem, onde  $x, y, z$ , etc são variáveis desta meta-linguagem.

Em relação à semântica, é assumido que os valores semânticos envolvem uma variedade de tipos e que, em geral, os membros de categorias sintáticas diferentes tomam valores semânticos de tipos diferentes. Assim:

Nome	Valor semântico
$d$	Donald Knuth
$n$	Zohar Manna
$j$	Jeffrey Ullman
$m$	Marvin Minsky

Poderia ser assumido que certos indivíduos não possuíssem nomes na linguagem  $L_0$ , ou que alguns indivíduos tivessem dois nomes, etc. A idéia básica é a existência de uma função matemática de nomes da linguagem para o conjunto de indivíduos do domínio do discurso. Esta função no caso de  $L_0$  deve, em princípio, ser total, no sentido de que nenhum nome fique sem seu par.

Por outro lado, o valor semântico do predicado  $B$  é definido aqui como o conjunto dos indivíduos sobre os quais  $B$  é verdadeiro. É assim para todos os outros predicados. Assim sendo, a semântica fica como extensional, no sentido de que o tratamento semântico de um predicado envolve somente sua extensão. Isto é, uma sentença com  $M(n)$  é verdadeira se e somente se o indivíduo denotado pelo nome  $n$  (no caso Zohar Manna) for um membro do conjunto dos indivíduos denotados pelo predicado  $M$ :  $M(n)$  é verdadeira se e somente se  $n'$  pertence a  $M'$ , onde  $x'$  deve ser lido como a denotação de  $x$ . Esta mesma convenção se aplica também a predicados binários.

Em relação ao número de regras semânticas, é assumido que o componente semântico se submete ao Princípio da Composicionalidade: existe uma correspondência um-para-um entre as regras sintáticas e as regras semânticas.

As expressões básicas da semântica de  $L_0$  são:

$d'$  = Donald Knuth,  
 $n'$  = Zohar Manna,  
 $j'$  = Jeffrey Ullman,  
 $m'$  = Marvin Minsky,  
 $M'$  = conjunto das pessoas carecas,  
 $K'$  = conjunto das pessoas que amam,  
 $B'$  = conjunto das pessoas tal que a primeira é amiga da segunda,  
 $L'$  = conjunto das pessoas tal que a primeira conhece a segunda.

As regras semânticas de  $L_0$  são:

- se  $x$  é um predicado unário e  $y$  é um nome, então  $x(y)$  é verdade se e somente se  $y'$  pertence a  $x'$ ;
- se  $x$  é um predicado binário e  $y$  e  $z$  são nomes, então  $x(y,z)$  é verdade se e somente se  $\langle y', z' \rangle$  pertence a  $x'$ ;
- se  $x$  é uma sentença, então " $x$  é verdade se e somente  $x$  não é verdade";
- se  $x$  e  $y$  são sentenças, então  $[x \ \& \ y]$  é verdade se e somente se ambos  $x$  e  $y$  são verdade;
- se  $x$  e  $y$  são sentenças, então  $[x \ \vee \ y]$  é verdade se e somente se ou  $x$  é verdade ou  $y$  é verdade;
- se  $x$  e  $y$  são sentenças, então  $[x \ \rightarrow \ y]$  é verdade se e somente se ou  $x$  é falso ou  $y$  é verdade;
- se  $x$  e  $y$  são sentenças, então  $[x \ \leftrightarrow \ y]$  é verdade se e somente se ambos são verdade ou ambos são falsos.

Este exemplo muito simples mostra então que:

- existe um conjunto de coisas que recebem valor verdade (isto envolve um conjunto de indivíduos, um conjunto de valores verdade e várias funções);
- há a especificação, para cada categoria sintática, de um tipo de valor semântico (as sentenças recebem valores verdade);
- existe um conjunto de regras semânticas que especificam como o valor semântico de uma expressão complexa é determinado em

função dos valores semânticos de seus componentes;

- existe um assinalamento específico de um valor semântico de um tipo apropriado para cada uma das expressões básicas.

Uma regra de formação sintática possui a forma: se  $a, b, c, \dots, m$  são categorias sintáticas e se  $f$  é uma função de formação, então  $v$  é uma categoria sintática tal que  $v = f(a, b, c, \dots, m)$ . A regra semântica correspondente possui a forma: se  $a', b', c', \dots, m'$  são as correspondentes interpretações semânticas de  $a, b, \dots, m$  e se  $g$  é um funtor semântico, então  $v' = (f(a, b, \dots, m))'$  e isto é igual a  $g(a', b', \dots, m')$ .

Além disso:

- para cada regra de formação sintática existe uma regra de interpretação semântica (isomorfismo entre a sintaxe e a semântica);

- todas as locuções de uma certa categoria sintática recebem interpretações de um mesmo tipo lógico;

- o operador tipo  $f$  gera cadeias; ele é um operador de composição (concatenação e substituição de variáveis);

- o operador tipo  $g$  é um conectivo (como será visto a seguir, pode ser também quantificador, constante lógica, aplicação de argumento a função, operador de intensão, operador de extensão).

### 3.4 Verdade relativa a um modelo

As três idéias discutidas antes no que diz respeito à semântica foram: valor verdade, teoria dos modelos e mundo possível. Aqui será discutida uma quarta idéia também essencial ao tipo de semântica aqui apresentado.

A idéia é que uma determinada sentença assume o valor semântico (verdadeiro ou falso) determinado por suas propriedades estruturais e por certos fatos do mundo. Assim um exemplo tipo "Zohar Manna tosse" assume o valor semântico  $i$  (verdade) devido às interações dos seguintes resultados:

- "Zohar Manna" existe no mundo em questão, tendo então o valor semântico  $z$ ;

- "tosse" tem como valor semântico uma função de indivíduos para valores verdade, tal que mapeia Zohar Manna em  $i$ ;

- "Zohar Manna" é um nome e "tosse" é um predicado com um argumento, e o valor verdade de uma sentença composta por um nome mais um predicado unário é determinado pela aplicação da função (que é o valor semântico do predicado unário) ao argumento (que é o valor semântico do nome).

Formalmente um modelo deste tipo pode ser visto como um par ordenado  $\langle A, \Phi \rangle$  onde  $A$  é um conjunto de indivíduos e  $\Phi$  é uma função que assinala valores semânticos de um certo tipo a expressões básicas. Todo o resto é tomado como uma parte fixa da semântica da linguagem.

As várias escolhas de um modelo são feitas para representar os vários modos que afetam o mapeamento de expressões básicas a coisas em mundos, enquanto o resto fixo representa a contribuição a valores semânticos feitas pela própria teoria.



No que tange a esta proposta, existem algumas vantagens da noção "verdade relativa a um modelo" sobre a noção simples de verdade. O lógico ou o linguista poderiam não ter, por exemplo, um grande interesse sobre questões do tipo "quais indivíduos pertencem a um certo conjunto e quais os que não pertencem" (embora isto interesse bastante por exemplo a um usuário de um banco de dados); porém questões mais interessantes para os primeiros certamente seriam as ligadas a uma formulação semântica explícita que envolva a idéia de que se certas denotações básicas são especificadas de tal modo, então uma definição precisa é feita automaticamente para a verdade das sentenças de uma linguagem.

As preocupações relativas a este automatismo, levam a formular o seguinte conjunto de definições para uma linguagem L:

- uma sentença de L é válida sse (se e somente se) é verdade em relação a cada mundo possível de L;

- uma sentença de L é contraditória sse é falsa em relação a cada mundo possível de L;

- duas sentenças de L são logicamente equivalentes sse a primeira é verdade nos mesmos modelos em que a segunda é verdade e em nenhum outro;

- uma sentença s de L é uma consequência lógica de um conjunto T de sentenças de L (isto é, T implica logicamente s), sse cada modelo no qual todas as sentenças de T são verdade é um modelo no qual s é verdade também.

Embora esta definição de consequência lógica envolva uma relação entre um conjunto T de sentenças e uma sentença s, pode ser obviamente considerada uma implicação entre um par de sentenças como um caso especial da definição, no qual T contém uma só sentença. A equivalência lógica fica então como uma implicação lógica entre um par de sentenças.

Uma vantagem importante desta abordagem é então a caracterização dos conceitos de validade e de implicação em termos de dedução e de regras de inferência, existindo mesmo uma correspondência entre as definições em termos de mundos possíveis e as definições em termos de dedução. Assim a definição de sentença válida em termos de dedução seria: uma sentença de L é um teorema de L sse existe uma prova desta sentença somente a partir dos axiomas de L. A definição de sentença contraditória seria: uma sentença de L é uma contradição sse sua negação é um teorema de L. A definição de sentença que é uma implicação lógica seria: uma sentença s de L é dedutível (ou pode ser provada) de um conjunto T de sentenças sse existe uma sequência de sentenças de L tal que cada uma ou seja um axioma ou pertença a T ou siga de alguma das sentenças precedentes na sequência através de regras de inferência, e s seja a última sentença da sequência. A definição de sentença logicamente equivalente seria: duas sentenças de L são logicamente equivalentes sse cada uma é dedutível da outra.

Esta possibilidade de dar definições correspondentes em semântica dos modelos e em sistemas dedutivos tem um enorme significado para a linguística, na medida em que uma axiomatização puder ser feita para uma linguagem formalizada adequada às representações semânticas de linguagens naturais. Lakoff /LAK 72/, em relação a isto, coloca que esta linguagem

poderia dar mesmo conta das inferências corretas feitas nas linguagens naturais.

Um outro ponto extremamente importante em relação às correspondências entre a semântica dos modelos e os sistemas dedutivos é que a definição de verdade em relação a um modelo tem a vantagem de capturar as definições de verdade lógica, de implicação lógica, etc., e ao mesmo tempo capturar a intuição que o falante/ouvinte tem sobre a relação que a sua linguagem natural possui com o mundo; em relação a isto, os sistemas dedutivos satisfazem só o primeiro destes objetivos, e não o segundo.

Uma outra vantagem para preferir o método semântico sobre o dedutivo é que certas lógicas não podem receber uma definição axiomática de validade e de implicação, porém uma definição destas lógicas em teoria dos modelos pode ser feita; por exemplo foi provado por Henkin /HEN 50/ que o conjunto das sentenças válidas da lógica de segunda ordem não pode ser axiomatizada por métodos finitos, enquanto que a definição semântica da validade e da implicação deste conjunto pode ser feita.

Uma questão, atinente às relações entre o método semântico e o método axiomático, diz bem respeito ao trabalho aqui desenvolvido: pode ser perguntado se existe uma axiomatização do fragmento da língua inglesa descrito no PTQ /MON 73/. A esse respeito ainda hoje não existe uma prova definitiva, embora Gallin /GAL 75/ conjecture que esta axiomatização exista. O problema a resolver, do ponto de vista do autor deste trabalho, é que uma linguagem natural envolve esquemas com construções sintáticas e itens lexicais cujas análises semânticas se ressentem de uma axiomatização completa. Assim por exemplo uma sentença tipo "Não existe modo para que todos os meninos levem meninas diferentes para a festa" envolve necessariamente quantificação sobre funções arbitrárias de indivíduos para indivíduos, representando modos de parear um menino com uma menina que ele leva para a festa. Como a lógica de segunda ordem é semidecidível, isto acarreta alguns problemas à primeira vista sérios para o método da axiomatização.

### 3.5 Expandindo L<sub>0</sub>

Uma questão relativa à linguagem L diz respeito à adição de variáveis individuais e de quantificadores sobre variáveis no componente sintático. Isto acarreta uma distinção entre "fórmula" e "sentença", de tal modo que uma sentença seja uma fórmula que não contenha nenhuma ocorrência livre de variáveis; ao lado disso, uma fórmula pode ou não conter ocorrências livres de variáveis. No que diz respeito à interpretação de L, este acréscimo envolve a noção de "satisfação de uma fórmula pelo assinalamento de objetos a variáveis"; ou seja, para usar uma noção equivalente encontrada no PTQ: a verdade de uma fórmula com respeito a um assinalamento de valores a variáveis. Em relação a isso, tradicionalmente é dito que uma fórmula com variáveis livres não pode ser verdadeira ou falsa, porque as variáveis assim não podem denotar indivíduos particulares: as fórmulas

assim não fazem nenhuma asserção real até que as variáveis tenham sido quantificadas ou substituídas por nomes no curso de uma dedução. A tradição tem usado mesmo o termo "função proposicional" /RUS 05/ para estas fórmulas, de tal modo que elas podem ser vistas não como proposições mas como funções que geram proposições quando supridas com indivíduos como argumentos. Esta abordagem culmina então por levar a uma teoria de substituição de quantificação, de tal forma que as sentenças compostas sejam construídas a partir de funções proposicionais mais simples e nem sempre a partir só de sentenças mais simples /TAR 44/, como ocorria em Lo.

Esta noção de satisfação (verdade relativa a um assinalamento a variáveis) deve ser acrescentada à maquinaria semântica, a partir da criação de uma função que assinale a cada variável da atual linguagem L algum valor de um certo domínio. Em relação a esta função, ela não precisa necessariamente ser do tipo um-para-um.

Desta forma, a noção de verdade relativa a um modelo então fica envolvendo dois estágios: a definição recursiva de fórmula verdadeira em L com respeito ao modelo M e a definição da função de assinalamento de valores a variáveis. Assim um modelo de L fica sendo um par  $\langle A, \# \rangle$  tal que A é um conjunto não vazio e  $\#$  é uma função que assinala um valor semântico a cada constante não lógica de L (isto é: nomes, predicados unários, predicados binários). O conjunto dos possíveis valores semânticos para nomes é A; o conjunto dos possíveis valores semânticos para predicados unários é  $\{1, 0\}$  expoente A; o conjunto dos possíveis valores semânticos para predicados binários é  $\{1, 0\}$  expoente A (onde  $[x$  expoente  $y]$  significa o conjunto de todas as funções de y para x). Por seu turno, um assinalamento g de valor é uma função que assinala um membro de A a cada variável de L.

Os valores semânticos das expressões básicas de L são:

- se u é uma variável individual de L, então o valor semântico de u, em relação a M e g, é  $g(u)$ ;
- se x é uma constante não lógica de L, então o valor semântico de x, em relação a M e g, é  $\#(x)$ .

As condições de verdade das fórmulas de L, em relação a M e g, são:

- se x é um predicado unário e y é um termo (constante não lógica ou variável), então o valor semântico de  $x(y)$ , em relação a M e g, é o valor semântico de x, em relação a M e g, aplicado ao valor semântico de y, este também relativo a M e g;

observação: seja  $k'$  a simplificação de "valor semântico de k, em relação a M e g"; a regra anterior afirma então que  $(x(y))' = x'(y')$ ;

- se x é um predicado binário e y e z são termos, então  $(x(y, z))' = [x'(z')] (y')$ ;

- se x é uma fórmula, então  $(\sim x)' = 1$  sse  $x' = 0$ ; de outro modo  $(\sim x)' = 0$ ; similarmente para  $[x \& y]$ , para  $[x \vee y]$ , para  $[x \rightarrow y]$  e para  $[x \leftrightarrow y]$ ;

- se x é uma fórmula e u é uma variável, então  $((\forall u)x)' = 1$  sse, para cada função de assinalamento  $g_1$  (onde  $g_1$  é como g, exceto para o indivíduo assinalado para u por g),  $x_1 = 1$  (onde  $k_1$  é o valor semântico de k, em relação a M e  $g_1$ );

- se  $x$  é uma fórmula e  $u$  é uma variável, então  $((\exists u)x)' = 1$  sse, para algum assinalamento  $g$ ,  $x+ = 1$ .

O valor semântico  $x'$  de uma expressão  $x$  depende então de função de assinalamento  $g$  somente com relação a quais os valores que  $g$  assinala às variáveis que estão livres em  $x$ . Dessa forma, deve ser observado então que, se  $g(u) = g_1(u)$  para todas as variáveis que estão livres em  $x$ , então  $x' = x_1'$ .

A definição de verdade para fórmulas de  $L$  relativa a  $M$  é:

- para cada fórmula  $x$  de  $L$ , o valor semântico de  $x$  relativo a  $M$  é igual a 1 sse  $x' = 1$  para todo assinalamento  $g$  de valor;

- para cada fórmula  $x$  de  $L$ , o valor semântico de  $x$  relativo a  $M$  é igual a 0 sse  $x' = 0$  para todo assinalamento  $g$  de valor.

Com estes acréscimos, as definições de validade, implicação, etc podem ser feitas nesta nova linguagem da mesma forma como antes.

### 3.6 Variáveis e quantificação

O PTQ de Montague possui uma maquinaria da forma referida acima para traduzir o fenômeno da quantificação em Inglês, tratando especificamente os quantificadores "every", "some", "the", "a". Se fosse usado uma lógica de predicados, a sentença "Cada estudante caminha" teria uma tradução semelhante a  $(\forall x)[E(x) \rightarrow C(x)]$ , onde  $E$  denota "estudante" e  $C$  denota "caminha"; por seu turno a sentença "Algum estudante caminha" teria como tradução  $(\exists x)[E(x) \& C(x)]$ . Deve ser observado que o caso mais simples de quantificação em Inglês, como em Português, é expresso por um determinante combinado com um nome comum formando uma locução nominal, como ocorre nos exemplos acima; deve ser observado que a formalização em lógica de predicados envolveu a transformação em predicado de um nome comum; além disso ocorreram mais dois esquemas: variáveis (que ficaram correspondendo sintaticamente a nomes) e regras de quantificação; o problema, à primeira vista, com esta formalização é que a correspondência de variáveis a nomes não parece natural.

Dowty, Wall e Peters /DOW 81/, apresentando didaticamente o PTQ, sugerem que a correspondência a variáveis numa linguagem natural é mais natural para pronomes com antecedente nominal, como é o caso em "Cada americano ama a si próprio" e em "Cada americano ama a sua mãe". Deve ser observado, quanto a isso, que estes pronomes não são todavia meros substitutos sintáticos de seus antecedentes, pois que as sentenças "Cada americano ama cada americano" e "Cada americano ama a mãe de cada americano" não possuem respectivamente os mesmos significados das sentenças anteriores. A paráfrase das primeiras sentenças é algo como "u<sub>1</sub> ama u<sub>1</sub>", se para cada valor de  $u_1$  exista um americano, e além disso a ocorrência mais à esquerda de  $u_1$  deve ser ligada a "cada americano" e a ocorrência mais à direita deve ser ligada a "a si próprio". Assumindo esta análise com o uso de variáveis, uma sentença como "Cada aluno caminha" pode ser analisada como "u<sub>1</sub> caminha", substituindo a ocorrência de  $u_1$  por "cada aluno".

A proposta do FTQ em relação a esse problema implica inicialmente na definição do seguinte conjunto de expressões básicas:

maria, pedro, ana são constantes da categoria N (nome);

u1, u2, ... são variáveis da categoria N;

homem, mulher, peixe são constantes da categoria NC (nome comum);

dorme, funga são constantes da categoria Vi (verbo intransitivo);

ama, come são constantes da categoria Vt (verbo transitivo);

e, ou são constantes lógicas da categoria Conj (conjunção).

As regras de formação sintática são:

- se x é um Vt e y é um N, então [xy] é um Vi;

- se x é um Vi e y é um N, então [yx] é uma For (fórmula);

- se x é uma Conj e y e z são For, então [yxz] é uma For;

- se x é um NC, u é uma variável e y é uma For contendo ao menos uma ocorrência de u, então y+ é uma For, onde y+ deriva de y pela substituição da ocorrência mais à esquerda de u por "algum x";

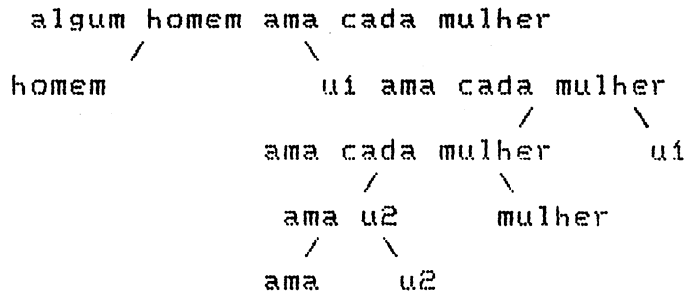
- se x é um NC, u é uma variável e y é uma For contendo ao menos uma ocorrência de u, então y+ é uma For, onde y+ deriva de y pela substituição da ocorrência mais à esquerda de u por "o x".

Esta gramática simples gera as seguintes árvores sintagmáticas:

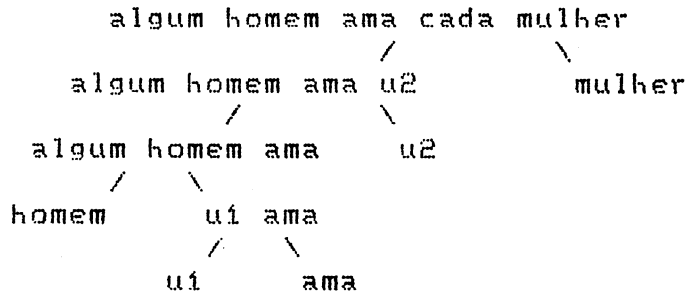


Uma consequência da análise acima é que as sentenças com "cada", "algum", "o" passam a ter um número infinito de análises sintáticas, uma usando u1, outra usando u2, outra u3, etc. Além disso, a cada análise sintática corresponde uma interpretação semântica.

Por outro lado, a fórmula "u1 ama u2" pode gerar a sentença "algum homem ama cada mulher" sob dois modos diferentes. Pela quarta regra de formação é possível gerar "u1 ama cada mulher" e então, pela quinta regra, gerar "Algum homem ama cada mulher". No entanto a aplicação destas mesmas duas regras em ordem inversa gera o mesmo resultado, gerando primeiro "algum homem ama u2" e depois a sentença em questão. A semântica, por seu turno, deve estipular duas formas diferentes de condições de verdade: a sentença derivada pelo primeiro modo é verdadeira se existe alguém que seja homem e que este alguém ame cada mulher; a condição de verdade para a segunda forma de derivação sintática corresponde a outra interpretação: para cada mulher existe algum homem ou outro homem, não havendo necessidade do mesmo homem para todas as mulheres.



1o caso



2o caso

Os valores semânticos das expressões básicas seriam:

- se  $u$  é uma variável, então  $u' = g(u)$ ;
- se  $a$  é uma constante não lógica, então  $a' = f(a)$ ;
- se  $a$  é uma constante lógica (membro de Conj), então  $a'$  pode ter as seguintes definições em tabela verdade:

$e'$ :	1 1 1	$ou'$ :	1 1 1
	0 0 0		1 0 1
	1 0 0		0 1 1
	0 1 0		0 0 0

As condições de verdade das fórmulas relativas a  $M$  e  $g$  são:

- se  $a$  é um  $Vt$  e  $b$  é um  $N$ , então  $(ab)' = a'(b')$ ;
- se  $a$  é um  $Vi$  e  $b$  é um  $N$ , então  $(ba)' = a'(b')$ ;
- se  $a$  é um Conj e  $z$  e  $w$  são For, então  $(zaw)' = [a'(z')](w')$ ;
- se  $a$  é um NC,  $u$  é uma variável e  $z$  é uma For contendo ao menos uma ocorrência de  $u$ , então  $z+$  é uma For, onde  $z+$ , etc (como acima).

### 3.7 Tipos e operador Lambda

Montague propôs um sistema para nomear as categorias sintáticas:

- a categoria dos termos (nomes e variáveis individuais) é designada pelo símbolo "e" (mnemônico de "entity");
- a categoria das fórmulas (incluindo as sentenças) é designada pelo símbolo "t" (mnemônico de "true");
- a categoria dos predicados com um argumento é designada pelo símbolo  $\langle e, t \rangle$ , na medida em que estes predicados mapeiam um elemento da categoria e num elemento da categoria t;
- a categoria dos predicados com dois argumentos é designada pelo símbolo  $\langle e, \langle e, t \rangle \rangle$ , na medida em que estes predicados mapeiam um elemento da categoria e num elemento da categoria dos predicados com um argumento.

Os termos (constantes ou variáveis) possuem como valor semântico indivíduos ou entidades no universo do discurso; as fórmulas recebem valores verdade; uma expressão das outras duas categorias toma como valor semântico algum tipo de função: a predicados de um argumento são assinalados funções de indivíduos para valores verdade (daí então a denotação  $\langle e, t \rangle$ ) e a predicados com dois argumentos são assinalados funções de indivíduos para funções de indivíduos para valores verdade (e daí a denotação  $\langle e, \langle e, t \rangle \rangle$ ).

Esta notação permite escrever regras sintáticas sobre predicados e fórmulas, tais como:

- um predicado unário ( $\langle e, t \rangle$ ) combina-se com um termo ( $\langle e \rangle$ ) e gera uma fórmula ( $\langle t \rangle$ );

- um predicado binário ( $\langle e, \langle e, t \rangle \rangle$ ) combina-se com dois termos ( $\langle e \rangle$ ,  $\langle e \rangle$ ) e gera uma fórmula ( $\langle t \rangle$ ).

Já o operador de negação "¬" combina-se sintaticamente com uma fórmula e gera uma fórmula; seu valor semântico é uma função de valores verdade em valores verdade; sua notação será então  $\langle t, t \rangle$ . Por razões similares os conectivos  $\&$ ,  $\vee$ ,  $\rightarrow$  e  $\leftarrow$  pertencerão à categoria  $\langle t, \langle t, t \rangle \rangle$ , tal que cada um deles se combina com uma expressão da categoria  $t$  e gera uma expressão da categoria  $\langle t, t \rangle$ .

Com estas especificações é possível definir uma linguagem de tipos, cuja sintaxe está baseada na Teoria Simples de Tipos de Russel, e que é muito próxima à Teoria Funcional de Tipos de Church /CHU 40/.

Esta linguagem contém constantes e variáveis em cada categoria sintática e aceita quantificação sobre variáveis de qualquer categoria. Isto é: a extensão das variáveis não atinge só indivíduos (o que caracteriza uma linguagem de primeira ordem), mas atinge também predicados (o que caracteriza uma linguagem de segunda ordem) e também qualquer categoria definida na teoria dos tipos (o que caracteriza uma linguagem de alta ordem).

Uma definição recursiva do conjunto dos tipos é a seguinte:

- e é um tipo;
- t é um tipo;
- se a e b são tipos, então  $\langle a, b \rangle$  é um tipo;
- nada mais é um tipo.

Assim é possível criar símbolos complexos como  $\langle e, \langle e, t \rangle \rangle$ ,  $\langle e, \langle e, \langle e, t \rangle \rangle \rangle$ ,  $\langle \langle \langle e, t \rangle \rangle, \langle e, t \rangle \rangle, t$ , etc.

Nesta abordagem, as categorias sintáticas desta linguagem são vistas como conjuntos de expressões. Desta forma, o que foi especificado pela definição recursiva dada acima foi uma série de nomes (labels) de expressões. Para um linguista esta diferença todavia inexistente: o símbolo LN denota quer o nome de uma categoria, quer a categoria mesma, e quer uma variável sobre alguma expressão arbitrária desta categoria.

A sintaxe desta linguagem de tipos envolve um conjunto de tipos (estabelecido pela definição recursiva acima), um conjunto de expressões básicas e regras sintáticas. As expressões básicas consistem de constantes não lógicas e variáveis; assim:

- para cada tipo x, o conjunto de constantes não lógicas do tipo x, denotado por  $Conx$ , contém constantes  $cn, x$  para cada

número natural  $n$  (isto implica em um número infinito de constantes);

- para cada tipo  $x$ , o conjunto de variáveis do tipo  $x$ , denotado por  $Var_x$ , contém variáveis  $v_n$ , para cada número natural  $n$  (isto implica em um número infinito de variáveis).

As regras sintáticas são tais que o conjunto de expressões bem formadas do tipo  $x$  é definido recursivamente como:

- para cada tipo  $x$ , cada variável e cada constante não lógica do tipo  $x$  é um membro do conjunto de expressões bem formadas do tipo  $x$  (isto é: é uma  $EBF_x$ );

- para os tipos  $x$  e  $y$ , se  $a$  é uma  $EBF\langle x, y \rangle$  e  $b$  é uma  $EBF_x$ , então  $a(b)$  é uma  $EBF_y$ ;

- se  $a$  e  $b$  pertencem a  $EBF_t$ , então as seguintes expressões  $[ \sim a ]$ ,  $[ a \& b ]$ ,  $[ a \vee b ]$ ,  $[ a \rightarrow b ]$  e  $[ a \leftrightarrow b ]$  pertencem a  $EBF_t$  (onde  $t$  denota o tipo das fórmulas);

- se  $a$  pertence a  $EBF_t$  e se  $u$  pertence a  $Var_x$  (sendo  $x$  um certo tipo), então  $(\forall u)a$  e  $(\exists u)a$  pertencem a  $EBF_t$ .

Usando  $D_a$  para nomear o conjunto das possíveis denotações das expressões da categoria sintática  $a$ , a definição recursiva de  $D_a$  é:

-  $D_e$  é  $A$  (onde  $A$  é um conjunto não vazio cujo domínio é composto por indivíduos ou entidades);

-  $D_t$  é  $\{1, 0\}$ ;

- para cada categoria sintática  $a$  e  $b$ ,  $D\langle a, b \rangle$  é o conjunto de todas as funções de  $D_a$  para  $D_b$ .

Um modelo da linguagem estabelecida é um par ordenado  $\langle A, \Phi \rangle$ , onde  $A$  é o conjunto de indivíduos ou entidades, e  $\Phi$  é uma função que assinala uma denotação  $D_a$  a cada constante não lógica do tipo  $a$ . Uma função  $g$  de assinalamento de valor a variáveis existe como uma função que assinala a cada variável  $v_n$ , a uma denotação no conjunto  $D_a$ .

A denotação de uma expressão da linguagem  $L$  relativa ao modelo  $M$  e à função  $g$  de assinalamento de variável é definida recursivamente como:

obs.:  $x'$  é o valor semântico de  $x$  em relação a  $M$  e  $g$ .

- se  $x$  é uma constante não lógica, então  $x' = \Phi(x)$ ;

- se  $x$  é uma variável, então  $x' = g(x)$ ;

- se  $x$  é uma  $EBF\langle a, b \rangle$  e  $y$  é uma  $EBF_a$ , então  $(x(y))' = x'(y')$ ;

- se  $x$  e  $y$  são  $EBF_t$ , então  $(\sim x)'$ ,  $(x \& y)'$ ,  $(x \vee y)'$ ,  $(x \rightarrow y)'$  tem as definições como anteriormente;

- se  $x$  é uma  $EBF_t$  e  $u$  é uma  $Var_k$ , então  $((\forall u)x)' = 1$  sse, para todo  $e$  pertencente a  $D_a$ ,  $x' = 1$ ;

- se  $x$  é uma  $EBF_t$  e  $u$  é uma  $Var_k$ , então  $((\exists u)x)' = 1$  sse, para algum  $e$  pertencente a  $D_a$ ,  $x' = 1$ .

obs.: a notação foi simplificada, sendo que  $g+$  não foi referido.

Montague ainda expandiu esta linguagem de tipos através da adição de um ligador de variável, dito operador lambda. Este operador é chamado também de operador de abstração, abstrator de conjunto ou mesmo só de ligador. Assim  $\lambda x [ \dots x \dots ]$  é uma expressão que especifica um conjunto; aqui  $\dots x \dots$  é uma fórmula na linguagem de tipos, a qual é ligada ao operador lambda. De uma modo mais formal, se  $h$  é uma fórmula,  $\lambda x h$  é uma função que caracteriza o conjunto especificado por  $h$  com respeito à variável  $x$ .



Uma aplicação para este formalismo ocorre para sentenças tipo "João ama alguns indivíduos"; assim o conjunto de indivíduos amados por João pode ser representado por

$\lambda x [ (A(x)) (j) ]$

onde A denota "ama" e j denota um indivíduo particular "João". O conjunto dos indivíduos que ama João seria representado por

$\lambda x [ (A(j)) (x) ]$ .

A adição deste operador obriga a acrescentar à sintaxe a seguinte definição:

- se x é uma EBFt e u é uma Vare, então  $\lambda u [x]$  é uma EBF (e,t);

e à semântica a seguinte regra:

- se x é uma EBFt e u é uma Vare, então  $(\lambda u x)'$  é uma função h de A em  $\{0,1\}$ , tal que, para todo indivíduo e no conjunto A,  $h(e)=1$  sse  $x'=1$ ; de outra forma  $h(e)=0$ .

Como Montague aceita a capacidade de realizar abstração  $\lambda$  não só sobre fórmulas, mas sobre qualquer tipo, então:

- se x é uma EBFa e u é uma Varb, então  $\lambda u [x]$  é uma EBF(b,a);

- se x é uma EBFa e u é uma Varb, então  $(\lambda u [x])'$  é a função h de Db em Da, tal que, para todo objeto k em Db,  $h(k)=x'$ .

Uma aplicação deste formalismo pode ser feita à sentença "Cada estudante caminha". A tradução usual em lógica seria  $(\forall x)[E(x) \rightarrow C(x)]$ ; porém na linguagem definida agora seria  $\lambda P (\forall x) [E(x) \rightarrow P(x)] (C)$ , onde P é a notação abreviada para  $v\emptyset, \langle e,t \rangle$ , denotando a locução verbal LV. A condição de verdade desta sentença é tal que  $(\forall x) [E(x) \rightarrow P(x)]$  seja verdadeiro. Assim a denotação de  $\lambda P (\forall x) [E(x) \rightarrow P(x)]$  é o valor semântico de "cada estudante", desde que o valor semântico da sentença da forma "cada estudante LV" for obtido pela aplicação do valor semântico de "cada estudante" ao valor semântico de LV.

Da mesma forma, o valor semântico de "algum estudante" é  $\lambda P (Ex) [E(x) \& P(x)]$  e de "nenhum estudante" tem a forma  $\lambda P \sim (Ex) [E(x) \& P(x)]$ .

Uma sentença da forma "João LV" é usualmente transcrita em lógica como  $LV(j)$ . Trazendo para a notação defendida aqui, LV denota que P é verdade sse  $P(j)$  é verdade. Assim o valor semântico da locução nominal "joão" fica sendo igual ao de  $\lambda P [P(j)]$ . Nesta visão, a propriedade de caminhar (um exemplo de LV) tem uma propriedade de segunda ordem de ser verdadeira para João. Isto equivale a dizer que João caminha, porém de uma forma mais complicada.

Assim as locuções nominais "cada homem", "algum homem" e "nenhum homem" possuem respectivamente as seguintes representações, onde Q abrevia  $v1, \langle e,t \rangle$ :

$\lambda Q [ \lambda P (\forall x) [Q(x) \rightarrow P(x)] (H) ]$

$\lambda Q [ \lambda P (Ex) [Q(x) \& P(x)] (H) ]$

$\lambda Q [ \lambda P \sim (Ex) [Q(x) \& P(x)] (H) ]$

e desta forma o valor semântico de "cada" é:

$\lambda Q [ \lambda P (\forall x) [Q(x) \rightarrow P(x)] ]$

e assim sucessivamente para "algum" e "nenhum".

Esta abstração dupla pode gerar determinantes do tipo  $\langle \langle e,t \rangle, \langle \langle e,t \rangle, t \rangle \rangle$ , o qual se combina com um substantivo do tipo

$\langle e, t \rangle$  e gera uma locução nominal do tipo  $\langle \langle e, t \rangle, t \rangle$ .

Como  $\langle a, \langle a, t \rangle \rangle$  é um tipo de relação entre duas coisas do tipo  $a$ , os determinantes são relações entre dois tipos de indivíduos. Assim "cada" é uma relação de subconjunto, isto é:  $\langle \text{cada}(Q) \rangle(F)$  é verdade sse  $Q$  é um subconjunto de  $F$ .

Uma sentença do tipo "Cada estudante caminha" pode ser representada pela fórmula:

$\lambda Q [\lambda P [(\forall x) [Q(x) \rightarrow P(x)]]] (E) (C)$

e a árvore sintagmática faria  $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$  corresponder a  $\lambda Q [\lambda P (\forall x) [Q(x) \rightarrow P(x)]]$ , e  $\langle e, t \rangle$  corresponder a  $E$ , de modo a gerar  $\langle \langle e, t \rangle, t \rangle$ . Este último seria combinado com  $\langle e, t \rangle$  (que corresponde a  $C$ ), gerando  $t$  (que corresponde a sentença).

A estrutura de  $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$  faria  $\lambda Q$  corresponder a  $\langle e, t \rangle$  e  $\lambda P (\forall x) [Q(x) \rightarrow P(x)]$  corresponder a  $\langle \langle e, t \rangle, t \rangle$ . Já o  $\langle e, t \rangle$  de  $\langle \langle e, t \rangle, t \rangle$  corresponderia a  $\lambda Q$ . E  $\langle t \rangle$  seria inicialmente fatorado em  $\langle t, t \rangle$  e  $\langle t \rangle$ , sendo que  $\langle t, t \rangle$  seria fatorado em  $\langle t \rangle$  (que corresponde a  $Q(x)$ ) e em  $\langle t, \langle t, t \rangle \rangle$  (que corresponde ao operador  $\rightarrow$ ); e  $\langle t \rangle$  seria  $P(x)$ .

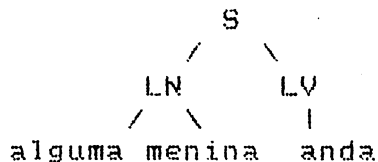
Assim o cálculo lambda pode servir para calcular o valor semântico de cada constituinte sintático de uma expressão complexa. Deve ficar claro porém que são os valores semânticos das expressões lambda (mais do que as expressões lambda em si) é que são idênticos aos valores semânticos das expressões da língua natural, no caso o Português. Desta forma, a linguagem de tipos acrescida do operador lambda não deve ser vista como um nível de representação linguística (assumindo aqui o conceito de nível das gramáticas transformacionais), mas sim como uma linguagem de tradução das expressões da linguagem natural referida.

### 3.8 Denotação de locução nominal

A idéia de denotar locuções nominais (LN) como funções de propriedades de entidades para valores verdade (ou seja: conjuntos de propriedades de entidades) foi a solução elegante de Montague para o problema da acomodação do tratamento do quantificador das linguagens formais às estruturas de constituintes encontradas na maioria das linguagens naturais. Keenan /KEE 72/ refere este problema através da discussão da diferença sintática entre uma sentença do tipo "Alguma menina anda" e  $(\exists x) [\text{menina}(x) \ \& \ \text{anda}(x)]$ . Para a abordagem lógica, esta sentença tem a seguinte árvore de análise, simplificada para exposição:

```
alguma menina anda
 /      \
alguma   menina anda
           /      \
         menina   anda
```

Porém para a abordagem das gramáticas transformacionais, a árvore tem a forma:



Na abordagem transformacional, existe alguma evidência de que "alguma menina" deva constituir um constituinte único em algum ponto da derivação, e isto fica em desacordo então com a análise semântica.

A solução proposta por Montague foi interpretar a segunda estrutura dizendo que "alguma menina" denota o conjunto das propriedades cada uma das quais é possuída por uma menina. Então a propriedade "anda" pode existir neste conjunto, no caso de pelo menos uma menina andar. A sentença total fica verdadeira assim só no caso em que o conjunto das propriedades denotadas pela LN-sujeito incluir a propriedade associada à LV; isto ocorre só no caso em que o conjunto de propriedades possuídas por ao menos uma menina inclui a propriedade de andar. Isto é de fato equivalente a dizer que a sentença é verdadeira só no caso onde ao menos uma menina anda.

O tratamento de "alguma menina" em termos de lógica intensional fica sendo então

$\lambda P (Ex) [menina(x) \rightarrow [(extensão\ de\ P)(x)]]$

onde P é uma variável sobre propriedades. Esta fórmula pode ser vista como denotando o conjunto de propriedades representado informalmente como:

$\{P: P \text{ é uma propriedade} \ \& \ (Ex) [menina(x) \ \& \ x \ \text{tem} \ P]\}$

onde o quantificador existencial foi usado como parte da caracterização do conjunto.

À primeira vista o tratamento de uma LN quantificada poderia parecer como diferente do tratamento de uma LN constituída por nome próprio, como "João". No caso da LN quantificada, a sentença é verdadeira só no caso em que a propriedade associada com a LV for um membro do conjunto das propriedades denotadas pela LN; no caso da LN com nome próprio, a sentença é verdadeira só no caso em que a entidade denotada pelo nome próprio possuir a propriedade associada com a LV.

Montague contudo propõe um tratamento único para estas duas formas de LN, fazendo com que nomes próprios também denotem conjuntos de propriedades. Assim a LN "João" denota o conjunto de propriedades possuídas por João. Isto provê um tratamento semântico unificado para as sentenças simples: uma sentença simples da forma [LN LV] é verdade só no caso onde o conjunto de propriedades denotado pela LN contiver a propriedade associada com a LV.

Em termos filosóficos, o movimento realizado por Montague representa uma solução elegante para o conflito entre o Russel moço e o velho. Isto porque Russel /RUS 03 parágrafos 60-65/ procurou especificar denotações para LNs como "todo homem", "um homem", "o homem", etc. e fez isso da seguinte forma: se a denota uma classe e a1, ..., an são seus membros, então "um a" denota a1 ou a2 ou ... an, onde 'ou' tem o significado que nenhum em particular deve ser tomado" (parágrafo 61, traduzido). Porém alguns anos depois ele /RUS 05:104/ rejeitou esta proposta

confusa e colocou que tais LNs deveriam ser interpretadas como unidades simples; assim "algum", "nenhum", etc. não são assumidos como tendo um significado isolado, mas um significado é agora assinalado para cada proposição onde eles ocorrem. Assim se C é algum predicado e x é uma variável, Russel agora diz que C(um homem) significa "[C(x) & x é homem] não é sempre falso". Esta proposta de Russel é tal que subjaz ao tratamento da quantificação nas linguagens formais modernas onde sentenças quantificadas são obtidas a partir de sentenças contendo variáveis livres; além disso esta proposta é feita em Teoria dos Conjuntos.

Montague unifica estas duas abordagens de Russel a partir de uma idéia tomada de Leibniz. De acordo com Mates /MAT 68/, Leibniz associa nomes próprios com conjuntos de atributos. Mas o uso que Leibniz faz disso é diferente da proposta de Montague: Leibniz queria ter algo associado com nomes como "Pegasus" sem ter que introduzir entidades possivelmente (mas não realmente) existentes. Então Pegasus foi associado somente com o conjunto dos atributos de Pegasus; em contraposição a isso, "Pedro" foi associado, por Leibniz, a um conjunto de atributos e a uma entidade.

Montague, por seu turno, não exitou em incluir entidades possíveis em sua ontologia, e assim usou a idéia de Leibniz para realizar esta generalização de nomes próprios para LN quantificadas. Assim fazendo, Montague usou o tratamento da quantificação de Russel como obtida através de sentenças vazias, porém ele usou a fórmula como parte da caracterização do conjunto de propriedades associadas com a LN. Então, ao usar a idéia de Leibniz, Montague torna aceitáveis ambas as sugestões de Russel.

### 3.9 Tratamento do tempo e da modalidade

Uma nova extensão da maquinaria foi feita por Montague para o tratamento do tempo dentro da linguagem intensional antes referida. A idéia é relativizar as denotações não só a um modelo e a assinalamento de valores, mas também a um momento no tempo (dado um conjunto de tempos).

Assim a definição de um modelo incluirá não só o domínio A e a função de interpretação  $\mathcal{I}$ , mas também o conjunto não vazio I dos momentos no tempo do modelo. Além disso I possui um ordenamento linear, tal que a relação  $<$  de ordenamento linear satisfaz as condições:

- $i \neq i$ ;
- $[i < j \ \& \ j < k] \rightarrow i < k$ ;
- $[i < j] \vee [i = j] \vee [j < i]$ ,

para todo i, j, k em I e a expressão  $i < j$  pode ser lida como "i é anterior a j" ou equivalentemente como "j é posterior a i".

A função  $\mathcal{I}$  que assinala uma denotação a cada constante não lógica torna-se agora uma função com dois argumentos: a constante e um membro de I. Assim para uma constante a e um certo i,  $\mathcal{I}(i, a)$  é a denotação de a no tempo i.

Desta forma um modelo temporal da linguagem L é uma quadrupla ordenada de  $\langle A, I, \langle, \$ \rangle$ , tal que A e I são conjuntos não vazios,  $\langle$  é um ordenamento linear e  $\$$  é uma função que assinala a cada par (consistindo de um membro de I e uma constante não lógica de L) uma denotação adequada. Agora a' será a simplificação de "a denotação de a relativa ao modelo M, tempo i e função g de assinalamento de valor a variáveis".

A lógica de tempo assume que o tempo presente é o ponto de referência inicial em todos os tempos; para formar uma linguagem com tempo é necessário então acrescentar o operador tempo futuro (F) e o operador tempo passado (P). Assim para a sintaxe:

- Px e Fx são fórmulas;

e para a semântica:

-  $(Fx)'=i$  sse existe algum j em I tal que  $i < j$  e  $x+=i$ ; de outra forma  $(Fx)'=\emptyset$ ;

obs.:  $x+$  e o valor semântico de x relativo a M, j, g.

-  $(Px)'=i$  sse existe algum j em I tal que  $j < i$  e  $x+=i$ ; de outra forma  $(Px)'=\emptyset$ .

É possível combinar os operadores de tempo gerando PFX (pretérito perfeito), FFX (futuro do pretérito), FFFx, etc, sendo assumido por axioma que:

FFx  $\rightarrow$  Fx

FPx  $\rightarrow$  Px

Por outro lado, é costume em lógica do tempo usar os operadores:

Gx =def  $\sim F \sim x$

Hx =def  $\sim P \sim x$

Esta lógica aceita também quantificadores como por exemplo  $(\forall x) F M(x)$ . Outros operadores acrescentados são o operador quadrado (nec) e o operador losango (pos), oriundos das lógicas modais. Assim uma expressão do tipo nec x (é necessário x) é verdade no mundo atual se x é verdade em todos os mundos possíveis; por outro lado pos x (é possível x) é verdade no mundo atual se x é verdade em ao menos um mundo possível.

Uma condição interessante, como mostraram Kripke /KRI 63/ e Hughes e Creswell /HUG 68/, é a possibilidade de aplicar o operador pos sobre um conjunto de indivíduos, trabalhando então com indivíduos possíveis que não são indivíduos reais. Um problema análogo ocorre para a lógica do tempo na distinção entre quantificador sobre indivíduos que estão no tempo presente versus quantificação sobre o conjunto de todos os indivíduos que ou existiram, ou existem ou existirão um dia /COC 65/. Montague aqui segue Dana Scott /SCD 70/, rejeitando esta maquinaria complexa, assumindo um domínio único A de quantificação para todos os mundos possíveis (ou, em lógica de tempo, para todos os tempos possíveis). Se for necessário expressar quantificação sobre os indivíduos reais, esta é feita de um modo simples pela inclusão do predicado de existência E entre as constantes não lógicas. A denotação do predicado E pode contudo variar de um mundo para o outro, o que é bastante razoável.

Assim  $(\forall u)$  nec  $x$  é lido como "cada indivíduo real é necessariamente tal que  $x$ " e pode ser expresso pela seguinte fórmula  $(\forall u) [E(u) \rightarrow \text{nec } x]$ ; por outro lado nec  $(\forall u) x$  tem como leitura "é necessário que cada indivíduo existente seja tal que  $x$ " e por seu turno pode ser expresso pela seguinte fórmula nec  $(\forall u) [E(u) \rightarrow x]$ . Além disso as fórmulas nec  $u$   $(\forall x)$  e nec  $(\forall u) x$  ficam como equivalentes, sendo lidas como "cada indivíduo real ou possível é necessariamente tal que  $x$ ".

Um outro problema interessante é o da identidade entre indivíduos em mundos diferentes. Na formulação de Montague, este problema não existe; porém na proposta de Carnap de identificar um mundo possível com um modelo possível ocorrem problemas sérios /QUI 53/ /LEW 68/.

Anteriormente foi definido um modelo para a linguagem modal como a tripla  $\langle A, I, \Phi \rangle$ , com operadores de tempo e operadores modais. A partir da discussão imediatamente anterior é necessário acrescentar a definição do conjunto  $W$  de mundos e a do conjunto  $T$  de tempos, onde  $\langle$  é ordenado sobre  $T$  (Montague escolheu aqui respectivamente  $I$  e  $J$ ). Desta forma o espaço semântico passa a possuir duas dimensões: um ponto particular neste plano possui um par de coordenadas  $\langle w, t \rangle$  para algum  $w$  em  $W$  e para algum  $t$  em  $T$ . Este par é dito um índice (Montague usa aqui a letra  $i$ ). Assim um modelo para a linguagem temporal fica sendo a quintupla  $\langle A, W, T, \langle, \Phi \rangle$ . Com este modelo,  $x'$  será lido como "a denotação de  $x$  relativa ao modelo  $M$ , ao mundo possível  $w$ , ao tempo  $t$  e função  $g$  de assinalamento de valores a variáveis".

Neste caso  $Fx$  é verdade para um índice  $\langle w, t \rangle$  somente se existir um outro índice  $\langle w, t_1 \rangle$  com  $t \langle t_1$  tal que  $x$  é verdade em  $\langle w, t_1 \rangle$ , e isso similarmente para os outros operadores de tempo. Além disso nec  $x$  é verdade num particular índice  $\langle w, t \rangle$  somente se  $x$  é verdade em todo  $\langle w_1, t_1 \rangle$ , para todo  $w_1$  em  $W$  e todo  $t_1$  em  $T$ .

O tempo é uma classe de fenômeno linguístico referido como expressão deítica pelos linguistas e como expressões indexicais pelos lógicos. O significado destas expressões de algum modo depende do contexto no qual a expressão está sendo usada. No caso do tempo verbal, existe uma dependência do tempo da enunciação da elocução.

Fillmore /FIL 75/ propõe em relação a isso que a noção de índice possa ser expandida de modo a por exemplo incluir "âncoras deíticas". Uma delas poderia ser um índice como  $\langle w, t, s \rangle$  onde  $w$  é um mundo possível,  $t$  é o tempo da elocução e  $s$  é um membro de  $A$  que representa o falante da elocução. Assim o pronome "eu" seria um termo individual relativo a um modelo, um mundo, um tempo, um falante e um assinalamento de valor, cuja regra semântica seria:

- o valor semântico de "eu" relativo a  $M, w, t, s, g$  fica sendo  $s$ .

Isto colocaria o espaço semântico como tridimensional. Mas outras coordenadas poderiam ser acrescentadas aos índices, para permitir por exemplo o tratamento de "aqui" e "ali" (relação de proximidade a um local). Lewis /LEW 70/ sugere por exemplo que um índice incluía uma coordenada de mundo possível, uma coordenada de tempo, uma coordenada do falante, uma coordenada do ouvinte

(para sentenças tipo "Vocês são porcalhões"), uma coordenada do objeto indicado (para sentenças tipo "Este porco é porcalhão") e uma coordenada do discurso feito anteriormente (para sentenças tipo "O antes mencionado porco é porcalhão"). Esta séptupla especificaria então um contexto relativo ao qual uma expressão receberia uma denotação.

Creswell /CRE 73/ propõe, ao contrário, tomar contextos como entidades primitivas e únicas, com um número ilimitado de propriedades; por exemplo a propriedade de ter um certo falante, um certo ouvinte, um certo tempo, etc. As interpretações das expressões dependeriam então não dos constituintes de um contexto (isto é: de um índice), mas sim das propriedades que o contexto tem. Por outro lado Kaplan /KAP 70/ procura mostrar as vantagens de um processo com dois estágios, tal que o primeiro mapeie os componentes adicionais do contexto a um sentido, que é então mapeado em mundos possíveis e momentos de tempo para gerar as denotações.

Este campo de problemas leva à questão da significação das expressões para o lado da pragmática /MON 68/. Em relação a isso, Montague propõe uma formalização numa simples linguagem de primeira ordem, usando a noção de índice. Thomason /THO 74/ contrapõe a isso que a pragmática deve incluir o estudo do ato de fala, da pressuposição, da implicação conversacional, etc, o que tornaria o problema bem mais complicado. Se a questão ficar restrita apenas a semântica (aceitando-se que é possível separar a semântica da pragmática), é possível contudo distinguir uma semântica com índices e uma semântica sem índices, onde o programa da semântica atinja só a um tipo de problema não referido por Thomason.

Um problema importante todavia reside na lógica intensional assim estabelecida: o da definição da denotação dos símbolos  $F$ ,  $P$ ,  $nec$ ,  $pos$  em si; o que ficou estabelecido até agora foi a denotação de expressões tipo  $Fx$ ,  $Px$ ,  $nec x$ . Isto se prende à consideração tradicional dos operadores temporais e dos modais como não fazendo parte dos operadores funcionais de verdade: O problema geral aqui é se a denotação de uma expressão pode ser explicada sempre em termos das denotações de suas partes, mesmo quando estas partes envolvem operadores temporais e modais.

Frege em 1893 trouxe este problema para o operador  $nec$ , discutindo a aplicação da Lei de Leibniz para expressões como:

- Necessariamente, a estrela da manhã é a estrela da manhã;
- Necessariamente, a estrela da manhã é a estrela da tarde.

A Lei de Leibniz estabelece que o resultado da substituição, numa fórmula, de um nome por outro nome que denote o mesmo indivíduo, fica verdadeiro sse a fórmula original for verdadeira (uma discussão aparece na Dissertação de Mestrado em Linguística de Mosca /MOS 83/). Estabelecido isto, Frege aceita que a primeira sentença acima é verdadeira ( $nec (a=a)$ ) e que a segunda não é verdadeira, devido a ser uma matéria de fato contingente e não um fato lógico.

Outro caso discutido por Frege envolve expressões com verbos epistêmicos como em:

- João acredita que a estrela da manhã é a estrela da manhã;
- João acredita que a estrela da manhã é a estrela da tarde.

Estas expressões são referidas como "construções de referência opaca", porque elas são opacas em relação à substituição de nomes por co-designativos. Elas se opõem às "construções de referência transparente", nas quais se aplica a Lei de Leibniz.

Outro problema trazido por Frege é o referente ao objeto de verbos tipo "querer", "mandar", "necessitar", como em "João quer um unicórnio". Outro caso onde também a denotação de toda a sentença parece não depender das denotações de suas partes envolve adjetivos e advérbios. No caso dos adjetivos atributivos, parece natural que a denotação de "A casa de João é azul" seja verdadeira (sob certos índices) se o objeto denotado por "a casa de João" (sob certos índices) esteja no conjunto de coisas denotadas pelo adjetivo "azul" (sob certos índices); o problema reside especificamente nos adjetivos tipo "anterior", "posterior", os quais não podem ter como denotação a intersecção de dois conjuntos: a denotação de "senador anterior" (com certos índices) requer a denotação de "senador" (com certos índices) numa coordenada de tempo anterior.

A solução clássica de Frege para estes problemas envolve a distinção entre o sentido (Sinn) de uma expressão e a sua referência ou denotação (Bedeutung). Mais tarde estes termos tornaram-se respectivamente sinônimos de intensão e extensão. O objetivo desta distinção, em Frege, é a resolução do problema dos contextos não referenciais (contextos modais, contextos de crença, etc); a proposta de Frege é que as expressões em linguagem natural possuem um tipo de ambiguidade tal que algumas vezes uma expressão possui uma denotação normal e em outras circunstâncias uma expressão denota seu sentido. O problema com esta proposta é a formalização da noção de sentido e o estabelecimento de algum algoritmo que decida quando uma expressão tem uma denotação normal e quando não.

Carnap /CAR 47/ a esse respeito sugere que o sentido (intensão) de uma expressão é uma função de estados possíveis de existência que gera, para um certo estado de existência, a denotação (extensão) da expressão. A intensão, nesta proposta, fica gerando todas as denotações da expressão. A semântica de Kripke para lógica modal, usando mundos possíveis como índices, formaliza esta idéia /KRI 63/.

No caso de uma semântica com coordenadas, como ocorre na proposta de Montague, uma intensão deve ser vista como uma função de índices para extensões. Vendo o problema mais de perto, como um nome denota um indivíduo para certo índice, a intensão de um nome deve ser uma função de índices para indivíduos: é a função que dá, para cada contexto, a pessoa denotada pelo nome no contexto. Esta intensão é o conceito do nome particular. O problema aqui é com nomes do tipo "Presidente do Brasil", o qual denota um mesmo indivíduo em tempos diferentes. Kripke /KRI 72/ discute aqui se estes nomes "ordinários" possuem uma intensão ao lado de sua extensão.



Um predicado unário denota um conjunto de indivíduos, para cada índice. Então a intensão de um predicado unário é a função que dá, para cada índice, o conjunto denotado para cada índice. Esta intensão é dita uma propriedade. Assim a propriedade correspondente a "dorme" é a função que dá, para cada situação possível, o conjunto de indivíduos que estão dormindo nesta situação.

Uma fórmula denota um valor verdade para cada índice. Logo a intensão de uma fórmula é uma função de índices para valor verdade; esta intensão é dita uma proposição. No campo da semântica dos mundos possíveis é comum uma proposição ser descrita como "o conjunto dos mundos possíveis"; neste caso, as expressões indexadas são ignoradas e, em vez de funções de índices, estas proposições passam a ser vistas como funções de mundos possíveis.

Na proposta de Montague, intensão e extensão são noções interdefiníveis relativas a um modelo; da extensão de uma expressão para todos os índices pode ser determinada a intensão, na medida em que esta extensão é vista como uma função; por outro lado, da intensão de uma expressão pode ser determinada a extensão desta expressão para um certo índice (através da aplicação da intensão, como função, a este índice). Porém conhecendo a extensão de uma expressão com respeito a um só índice em geral não é suficiente para estabelecer a intensão desta expressão, porque é preciso conhecer a extensão desta expressão para todos os outros índices: a interdefinibilidade depende então de ter todo o modelo.

Por outro lado, as variáveis também possuem intensão: a extensão de uma variável é o valor assinalado a ela pela função  $g$  e não difere de um índice para o próximo índice; logo a intensão de uma variável é uma função constante sobre índices.

Num quadro simples, isto pode assim ser simplificado:

Categorias de expressões	Extensão	Intensão
Termos individuais (constantes ou variáveis)	Indivíduos de A	Funções de índices a indivíduos de A (conceito individual)
Predicados unários	Conjunto de indivíduos em A	Funções de índices a conjuntos de indivíduos em A (propriedades de indivíduos)
Fórmulas (incluindo sentenças)	Valores verdade	Funções de índices para valores verdade (proposições)

Deve ficar claro que conceitos individuais, propriedades de indivíduos, proposições não são expressões de uma linguagem natural ou formal, nem elementos de uma abstrata estrutura profunda de uma linguagem natural ou formal; eles são apenas construtos numa teoria de conjuntos e logo são independentes de uma linguagem. Estas intensões são correlacionadas sistematicamente com expressões da linguagem natural e da linguagem formal através de regras semânticas, porém as intensões em si mesmas são objetos semânticos extra-linguísticos, construídos através de instrumentos da Teoria dos Conjuntos sobre os conjuntos básicos  $A, W, T$  e  $(\emptyset, 1)$  de algum modelo.

Outro ponto importante é que uma proposição (que é a intensão de alguma sentença de uma linguagem natural) não necessita consistir de um só mundo possível, mas ela pode se referir a um conjunto de mundos. Por exemplo, a sentença "João sonhou que ele visitou Portugal" descreve vários mundos possíveis que se relacionam com a expressão "João visitou Portugal". Esta proposição pode incluir mundos possíveis nos quais João visitou Portugal no verão e mundos nos quais ele fez a visita no inverno; pode incluir mundos nos quais ele fez uma viagem longa e mundos nos quais a visita foi curta, etc. Porém esta proposição não contém um destes mundos (potencialmente infinitos) no qual João não visitou Portugal. Além disso, quanto mais uma proposição for específica, menor o conjunto de mundos possíveis que ela contém.

Retornando ao problema da expressão  $[nec\ x]$ , e considerando a proposição de Frege de que a interpretação da expressão  $[nec\ x]$  deve envolver a intensão de  $x$  (em relação a  $M$  e a  $g$ ) e não a extensão de  $x$  (em relação a  $M, w, t, g$ ) é possível conceber duas alternativas:

- mudar as regras semânticas de tal modo que, em circunstâncias corretas, possam ser usadas intensões bem como extensões na determinação do valor semântico de uma expressão complexa;

- modificar a linguagem objeto de modo a incluir expressões que possuam como extensão (denotação) as intensões de certas outras expressões.

A última proposta segue a idéia de Frege de que expressões em contextos oblíquos denotem suas intensões; neste caso, para cada sentença  $x$  é necessário estabelecer uma segunda expressão  $x^*$  que denote a intensão de  $x$ . Então se o operador  $nec$  (necessariamente) for combinado sintaticamente com  $x^*$ , a regra semântica correspondente deve definir a extensão de  $nec\ x^*$  em termos das extensões de  $nec$  e de  $x^*$ .

A outra abordagem parece mais formalizável e foi a seguida por Montague no PTQ.

A idéia de desenvolver uma lógica que contenha expressões denotando intensões e extensões aparece em Church /CHU 51/, na forma de uma abordagem axiomática, mas não de natureza semântica. Por outro lado, Kaplan /KAP 64/ procurou construir uma análise semântica para o sistema de Church, mas ele adota a noção de intensão de Carnap (função de mundos possíveis para extensões).

Montague, por seu turno, procurou desenvolver uma linguagem intensional e sua interpretação em teoria dos modelos segue as idéias desenvolvidas por Kripke para a semântica da lógica modal; além disso Montague aplica este formalismo para a análise dos fenômenos intensionais do Inglês. Nesta lógica intensional existe um modo de formar uma expressão que denota a intensão de uma expressão:

se  $x$  é uma expressão, então  $\hat{x}$  é uma expressão que denota a intensão de  $x$  em relação a  $M$  e  $g$ .

Mas como a noção de intensão é definida para cada expressão da linguagem, a expressão  $\hat{x}$  possui ela mesma uma intensão e uma extensão: a intensão de  $\hat{x}$  é então  $\hat{\hat{x}}$ , e assim sucessivamente.

Deve ser observado que a extensão de  $\hat{x}$  em relação a  $M, w, t, g$  será sempre a mesma para todos os índices  $\langle w, t \rangle$ , isto porque existe somente uma intensão de  $x$  em relação a  $M$  e  $g$ . Isto é: a intensão desta intensão será uma função constante sobre índices, dando a cada índice o valor da intensão de  $x$  em relação a  $M$  e  $g$ . O mesmo vale para a intensão da intensão da intensão de  $x$ , que será outra função constante que dá para cada índice a função constante sobre índice, dando a cada índice o valor da intensão de  $x$  em relação a  $M$  e  $g$ . Desta forma só intensões de "primeira ordem" precisam ser desenvolvidas na interpretação de uma linguagem natural no PTQ.

Um segundo esquema semântico foi introduzido por Montague na sua linguagem intensional: o operador de extensão  $ext$  (originalmente um acento circunflexo virado), que pode ser aplicado a uma expressão  $x$  que denota uma intensão gerando uma expressão bem formada  $ext x$ .

Para um índice  $\langle w, t \rangle$  a expressão  $ext x$  denota o valor da intensão de  $x$  em relação a  $M$  e  $g$ , com  $\langle w, t \rangle$ . Isto é: se  $x$  denota uma intensão, então, para um índice  $\langle w, t \rangle$ ,  $ext x$  com relação a  $M, w, t, g$  é definido como  $x^*(\langle w, t \rangle)$ .

O efeito de combinar  $ext$  com uma expressão da forma  $\hat{x}$  é o cancelamento dos dois operadores:

a interpretação de  $[ext \hat{x}]$  com relação a  $M, w, t, g$  é a interpretação de  $x$  com relação a  $M, w, t, g$ .

### 3.10 A lógica intensional do PIQ

A lógica intensional do PTQ utiliza os recursos sintáticos e semânticos desenvolvidos até agora neste texto. Ela emprega hierarquia de tipos, quantificação de alta ordem, abstração lambda para todos os tipos, operadores de tempo, operadores modais, operador  $\hat{\phantom{x}}$  e operador  $ext$ .

A sintaxe do PTQ envolve um conjunto de tipos, um conjunto de expressões básicas e regras sintáticas.

O conjunto de tipos é definido recursivamente como:

- t é um tipo;
- e é um tipo;
- se a e b são tipos, então  $\langle a, b \rangle$  é um tipo;
- se a é um tipo, então  $\langle s, a \rangle$  é um tipo;
- nada mais é um tipo.

obs.: os tipos gerados pela penúltima definição serão usados para as intensões correspondentes a cada tipo a.

As expressões básicas são:

- para cada tipo a, existe um conjunto enumeravelmente infinito de constantes não lógicas  $c_{n,a}$ , para cada número natural n; o conjunto de todas as constantes não lógicas do tipo a é denotado por  $C_{na}$ ;

- para cada tipo a, existe um conjunto enumeravelmente infinito de variáveis  $v_{n,a}$ , para cada número natural n; o conjunto de todas as variáveis do tipo a é denotado por  $V_{na}$ .

As regras sintáticas que geram as expressões bem formadas do tipo a (EBFa) são:

- cada variável do tipo a é uma EBFa;
  - cada constante do tipo a é uma EBFa;
  - se x é uma EBFa e u é uma variável do tipo b, então  $[\lambda u x]$  é uma EBF $\langle b, a \rangle$ ;
  - se x é uma EBF $\langle a, b \rangle$  e y é uma EBFa, então  $x(y)$  é uma EBFb;
  - se x e y estão em EBFa, então  $[x = y]$  pertence a EBFt;
- (obs.: isto é novo)
- se x e y estão em EBFt, então  $\sim x$ ,  $[x \vee y]$ ,  $[x \& y]$ ,  $[x \rightarrow y]$ ,  $[x \leftrightarrow y]$  estão em EBFt;
  - se x é uma EBFt e u é uma variável de algum tipo, então  $[(\forall u) x]$  e  $[(\exists u) x]$  estão em EBFt;
  - se x é uma EBFt, então  $\text{nec } x$ ,  $\text{Fx}$ ,  $\text{Px}$  estão em EBFt;
  - se x é uma EBFa, então  $\hat{x}$  é uma EBF $\langle s, a \rangle$ ;
  - se x é uma EBF $\langle s, a \rangle$ , então  $\text{ext } a$  é uma EBFa.

A semântica da linguagem intensional, posta da forma antes apresentada, envolve um modelo, regras semânticas e definições adicionais.

Um modelo para esta linguagem intensional é uma quintupla ordenada  $\langle A, W, \$, T, \langle, F \rangle$ , tal que A, W, T são conjuntos não vazios,  $\langle$  é um ordenamento linear sobre T e  $\$$  é uma função cujo domínio é o conjunto das constantes não lógicas da linguagem.

O conjunto das possíveis denotações do tipo a é:

$D_e = A$ ;

$D_t = \{0, 1\}$ ;

$D_{\langle a, b \rangle} =$  conjunto de todas as funções de  $D_a$  para  $D_b$ ;

$D_{\langle s, a \rangle} =$  conjunto de todas as funções de  $W \times T$  para  $D_a$ , onde  $W \times T$  é o conjunto de todos os índices  $\langle w, t \rangle$ .

O conjunto de todos os sentidos do tipo a, denotado por  $S_a$ , é definido como  $D_{\langle s, a \rangle}$ . A função  $\$$  assinala a cada constante não lógica do tipo a um membro de  $S_a$ .

A palavra nova que apareceu aqui foi "sentido", porque Montague distingue entre "sentido" e "intensão". Para ele, o conjunto dos sentidos do tipo  $a$  é simplesmente o conjunto das possíveis intensões onde as intensões das expressões do tipo  $a$  são escolhidas. Isto é: todas as intensões de expressões são sentidos, mas necessariamente nem todos sentidos são intensões de alguma expressão. A função  $\$$  assinala um membro de  $S_a$  a cada constante de tipo  $a$ , assinalando a sua intensão; antes esta função  $\$$  era uma função de dois argumentos (um índice e uma expressão), mas aqui ela passa a ser uma função de um só argumento (a expressão). Isto ocorre porque o valor assinalado a cada expressão é outra função, que é a intensão; esta outra função (a intensão) é, neste caso, uma função de índices a denotações apropriadas.

Uma função  $g$  de assinalamento de valores a variáveis tem como domínio o conjunto de todas as variáveis e dá, como valor, para cada variável do tipo  $a$ , um membro de  $D_a$ . Deve ser observado, diferentemente a como vinha sendo antes, que  $g$  assinala a cada variável uma extensão, enquanto  $\$$  assinala a cada constante não lógica uma intensão (e não mais uma extensão).

Assumindo que a extensão da expressão  $x$  em relação ao modelo  $M$ ,  $w$  pertencente a  $W$ ,  $t$  pertencente a  $T$ , e com assinalamento  $g$  de valor é denotada por  $x'$ , as regras semânticas são as seguintes para uma expressão  $x$ :

- se  $x$  é uma constante não lógica, então  $x' = [\$ (x)] (\langle w, t \rangle)$ , isto é: a extensão de  $x$ , com relação a  $w$  e  $t$ , é o resultado da aplicação da intensão de  $x$  (formado por  $\$$ ), sobre o argumento  $\langle w, t \rangle$ ;

- se  $x$  é uma variável, então  $x' = g(x)$ ;

- se  $x$  é uma EBF $a$  e  $u$  é uma variável do tipo  $b$ , então  $(\lambda u x)'$  é a função  $h$  com domínio  $D_b$ , tal que, para cada objeto  $k$  neste domínio,  $h(k) = x+$ ;

obs.:  $x+$  é a interpretação de  $x$  com relação a  $M$ ,  $w$ ,  $t$ ,  $g_1$ , onde  $g_1$  é uma função de assinalamento (de valor) igual a  $g$ , exceto com a diferença de que  $g_1(u) = x$ ;

- se  $x$  é uma EBF $\langle a, b \rangle$  e  $y$  é uma EBF $a$ , então  $(x(y))'$  é  $x'(y')$ ;

- se  $x$  e  $y$  são EBF $a$ , então  $(x=y)'$  = 1 sse  $x'$  é o mesmo que  $y'$  (a condição é a identidade das denotações das duas expressões de mesmo tipo);

- se  $x$  é uma EBF $t$ , então  $(\sim x)'$  = 1 sse  $x' = \emptyset$ , e  $(\sim x)'$  = 0 de outra forma;

- se  $x$  e  $y$  são EBF $t$ , então  $(x \& y)'$  = 1 sse ambos  $x'$  e  $y'$  são 1;

obs.: definições similares para os conetivos  $[x \vee y]$ ,  $[x \rightarrow y]$  e  $[x \leftrightarrow y]$ ;

- se  $x$  é uma EBF $t$  e  $u$  é uma variável, então  $((\forall u) x)'$  = 1 sse  $x+ = 1$  para todo  $g_1$ ;

- se  $x$  é uma EBF $t$  e  $u$  é uma variável, então  $((\exists u) x)'$  = 1 sse  $x+ = 1$  para algum  $g_1$ ;

- se  $x$  é uma EBF $t$ , então  $(nec x)'$  = 1 sse a interpretação de  $x$  com relação a  $M$ ,  $w_1$ ,  $t_1$ ,  $g$  for 1, para todo  $w_1$  em  $W$  e para todo  $t_1$  em  $T$ ;

- se  $x$  é uma EBft, então  $(Fx)'$  = 1 sse a interpretação de  $x$  com relação a  $M$ ,  $w$ ,  $t_i$ ,  $g$  for igual a 1, para algum  $t_i$  em  $T$ , tal que  $t < t_i$ ;
  - se  $x$  é uma EBft, então  $(Px)'$  = 1 sse a interpretação de  $x$  com relação a  $M$ ,  $w$ ,  $t_i$ ,  $g$  for igual a 1, para algum  $t_i$  em  $T$ , tal que  $t_i < t$ ;
  - se  $x$  é uma EBfa, então  $(\hat{x})'$  é a função  $h$  com domínio  $W \times T$  tal que, para todo  $\langle w_i, t_i \rangle$  pertencente a  $W \times T$ ,  $h(\langle w_i, t_i \rangle)$  seja a interpretação de  $x$  com relação a  $M$ ,  $w_i$ ,  $t_i$ ,  $g$ ;
  - se  $x$  é uma EBF $\langle s, a \rangle$ , então  $(\text{ext } x)'$  é igual a  $x'(\langle w, t \rangle)$ .
- Outras definições são:
- se  $x$  é uma EBft, então  $x$  é verdade com relação a  $M$  e  $\langle w, t \rangle$  sse  $x' = 1$  para todo assinalamento  $g$  de valor;
  - se  $x$  é uma expressão, então a intensão de  $x$ , com relação a  $M$  e  $g$ , é a função  $h$  com domínio  $W \times T$ , tal que para todo  $\langle w, t \rangle$  em  $W \times T$ ,  $h(\langle w, t \rangle) = x'$ .

Recordando que o operador = estabelece a identidade das denotações de duas expressões de mesmo tipo, pode ser estabelecido por exemplo que dois nomes possuem a mesma extensão, ou que dois predicados possuem a mesma extensão. A identidade entre duas intensões como  $\hat{x} = \hat{y}$  só ocorre entretanto de elas forem a mesma função, isto é: se elas gerarem o mesmo valor para cada índice  $W \times T$ . Assim sendo,  $\hat{x} = \hat{y}$  é mais forte que  $x = y$ .

No caso do operador = atingir duas expressões que são fórmulas (isto é, membros de EBft),  $x = y$  diz que  $x$  e  $y$  denotam o mesmo valor verdade, de tal forma que  $x = y$  é logicamente equivalente a fórmula  $x \leftrightarrow y$ . Como observação, deve ser notado que todos os operadores lógicos  $\&$ ,  $\vee$ ,  $\sim$ ,  $\rightarrow$ ,  $\leftrightarrow$ , nec,  $\forall$ , podem ser definidos em termos de =,  $\lambda$  e  $\hat{\phantom{x}}$ , com a ajuda de variáveis de vários tipos /MON 70a/.

Algumas intensões agora podem ser consideradas. Assim como indivíduos são denotados por expressões do tipo  $e$ , os conceitos individuais são denotados por expressões do tipo  $\langle s, e \rangle$ ; expressões deste tipo incluem constantes, variáveis e expressões bem formadas pela prefixação do operador de intensão  $\hat{\phantom{x}}$  sobre expressões do tipo  $e$ .

Por outro lado, como os predicados de um argumento são do tipo  $\langle e, t \rangle$ , as expressões do tipo  $\langle s, \langle e, t \rangle \rangle$  denotam propriedades de indivíduos. Como norma geral, uma expressão do tipo  $\langle s, \langle a, t \rangle \rangle$  denota uma propriedade de algum tipo de coisas que são denotadas por expressões do tipo  $a$ ; assim  $\langle s, \langle \langle s, t \rangle, t \rangle \rangle$  denota propriedades de proposições, e  $\langle s, \langle \langle s, \langle e, t \rangle \rangle, t \rangle \rangle$  denota propriedades de propriedades de indivíduos.

Por último, como  $\langle e, \langle e, t \rangle \rangle$  (correspondente a predicados binários) denota relações entre indivíduos, a noção intensional correspondente a esta relação é dita relação-em-intensão entre indivíduos e seu objeto semântico é denotado por expressões do tipo  $\langle s, \langle e, \langle e, t \rangle \rangle \rangle$ . Generalizando, as denotações das expressões do tipo  $\langle s, \langle a, \langle b, t \rangle \rangle \rangle$  são relações-em-intensão entre certas expressões denotadas pelo tipo  $b$  e certas expressões denotadas pelo tipo  $a$ . Assim as expressões do tipo  $\langle s, \langle \langle s, t \rangle, \langle e, t \rangle \rangle \rangle$  denotam relações-em-intensão entre indivíduos e propriedades e expressões do tipo  $\langle s, \langle \langle s, \langle e, t \rangle \rangle, \langle \langle e, t \rangle, t \rangle \rangle \rangle$

denotam relações-em-intensão entre conjuntos de indivíduos e propriedades de indivíduos.

Deve ser observado que *s* sozinho não constitui um tipo: isto quer dizer que semanticamente nenhuma expressão da linguagem intensional denota índices diretamente, mas sim somente pode denotar funções de índices para algo. A idéia subjacente é que uma linguagem natural não faz referência explícita a índices /GAL 75/.

Por outro lado, se *b* é do tipo *e*, a expressão  $\hat{b}$  (que é do tipo  $\langle s, e \rangle$ ) pode ser combinada com *a* (que é do tipo  $\langle s, e \rangle$ ), formando a expressão  $a(\hat{b})$  que estabelece que o conceito correspondente a *b* está na extensão de *a*. Além disso, se *a* é do tipo  $\langle s, e \rangle$ , a expressão  $\text{ext } a$  (que é do tipo *e*) pode ser combinada com *b* (do tipo  $\langle e, t \rangle$ ), formando a expressão  $a(\text{ext } b)$  que estabelece que a extensão do conceito *b* está na extensão de *a*.

E ainda: se *a* é do tipo  $\langle s, \langle e, t \rangle \rangle$  (denotando uma propriedade de indivíduos) e se *b* é do tipo *e* (denotando um indivíduo), é possível fazer a asserção de que "*b* tem como propriedade *a*". Isto pode ser feito, lembrando que uma propriedade é uma função de índices para funções de indivíduos para valores verdade, usando a asserção "*b* está na extensão da propriedade *a* com tal índice", isto é:  $\text{ext } a(b)$ . No PTQ esta última expressão é reescrita como  $a(b)$ .

Em relação a sentenças da forma "Necessariamente *x*", a palavra "necessariamente" é tratada aqui como um funtor *nec* ou quadrado que se combina com  $\hat{x}$ . Assim *nec* é tratada não como uma expressão do tipo  $\langle t, t \rangle$ , mas como uma expressão do tipo  $\langle \langle s, t \rangle, t \rangle$ , isto é: uma função de proposições a valores verdade. Além disso, a regra semântica para *nec* com os índices  $\langle w, t \rangle$  é tal que *nec* é uma função de proposições a valores verdade cujo valor é 1 quando aplicada a uma proposição *p* somente no caso em que *p* mapeia cada índice  $\langle w, t \rangle$  em 1; de outro modo irá mapear em 0.

Em relação aos contextos oblíquos formados por adjetivos como "anterior", a extensão da locução "senador anterior" fica definida em termos de suas partes, através da combinação de "anterior" não com uma expressão que denota um predicado, mas com uma expressão que denota uma propriedade. Estes adjetivos são então expressões do tipo  $\langle \langle s, \langle e, t \rangle \rangle, \langle e, t \rangle \rangle$ . Assim anterior é um funtor que acessa não só a extensão corrente do predicado com um argumento ao qual "anterior" se combina, mas também acessa a extensão deste predicado com outros índices; em especial, com índices numa coordenada de tempo anterior. Isto é: anterior é uma função que toma como entrada uma propriedade (de ser senador) e gera o conjunto de indivíduos que não são agora, mas foram, a extensão desta propriedade (ou seja: o conjunto dos senadores anteriores). Correspondentemente a intensão de "anterior" é a função que indica como isto é feito através de qualquer índice.

Em relação aos verbos tipo "acreditar", Montague concebe estes verbos como denotando uma relação entre indivíduos e proposições. Assim a constante *Acr* é tratada como uma expressão do tipo  $\langle \langle s, t \rangle, \langle e, t \rangle \rangle$ . Uma sentença como "João acredita que Maria é careca" pode ser traduzida nesta lógica intensional como:

(Acr (^[Careca(m)]))(j).

obs.: deve ficar claro que [(a(b)) (c)], [a (b) (c)] e [a(c,b)] são notações alternativas; no exemplo acima, (Acr(^[Careca(m)]))(j) pode ser rescrito como Acr(j,^[Careca(m)]).

Resta a questão da substituição de nomes. Em primeiro lugar, quanto a isso, se a interpretação de Careca(m) em relação a M, w1, t1, g for igual a interpretação de Careca(n) em relação a M, w1, t1, g, então a interpretação de m em relação a M, w1, t1, g fica igual à interpretação de n em relação a M, w1, t1, g (a Lei de Leibniz vale). Porém disto não segue que a proposição denotada por ^Careca(m) seja idêntica a proposição denotada por ^Careca(n) com certos índices, pois pode ser o caso que, para um índice <w, t2>, a fórmula Careca(m) seja falsa e Careca(n) seja verdadeira. Um exemplo mais concreto disso é fazer m = maria e n = Miss Brasil, e assumir que Maria é Miss Brasil e também assumir que a sentença "João acredita que Miss Brasil é careca" é verdade enquanto que a sentença "João acredita que Maria é careca" é falsa. Estas sentenças são semelhantes aos exemplos de Frege com "estrela da manhã" e "estrela da tarde" /FRE 52/.

Assim, somente no caso de expressões sem contexto oblíquo é verdade que  $a=b \rightarrow [x \rightarrow x a/b]$ , onde  $[x a/b]$  é o resultado da substituição por b de toda ocorrência de a em x. No caso da lógica intensional desenvolvida por Montague, isso só é verdade quando x está sob o escopo dos operadores  $\hat{\quad}$ , nec, F e P. Além disso, uma restrição importante a esta fórmula é que deve ser excluída toda ligação da variável (isto é: a variável deve ser livre). Outra versão restrita da fórmula acima ocorre no caso onde a e b não somente possuem a mesma extensão ao índice corrente, mas possuem também a mesma intensão; neste caso, se duas expressões do mesmo tipo possuem a mesma intensão, elas podem livremente substituírem-se num contexto sintático, preservando a extensão e a intensão da expressão original.

A sentença "João acredita que Miss Brasil é careca" pode ser entendida de dois modos. Um modo é a asserção de que João tem uma crença sobre um certo indivíduo que parece ser Miss Brasil ao índice corrente, sendo que esta crença é que tal indivíduo é careca. Esta leitura é chamada de "leitura de re", pois Miss Brasil é a descrição que o falante tem sobre a pessoa em questão. O outro modo de leitura (de dicto) é a asserção de que João acredita que qualquer um indivíduo que seja chamado de "Miss Brasil" é careca; isto descreve a crença de João, mesmo que ele não conheça a quem o nome "Miss Brasil" denote realmente. Deve ser notado que  $Acr(joão, ^[Careca(m)])$  formaliza a segunda leitura (de dicto). Já a primeira leitura (de re) pode ser formalizada na lógica intensional de Montague através do operador lambda:  $lambda x [Acr(joão, ^[Careca(x)])] (m)$ ; esta formulação diz que um indivíduo, denotado por m, e tal que João acredita na proposição que este indivíduo é careca. Se a conversão lambda for aplicada à última fórmula acima, resulta a fórmula  $Acr(joão, ^[Careca(m)])$ ; assim o princio da conversão lambda deve ser restrito na linguagem intensional proposta por Montague, devendo ficar sujeito a mesma restrição que a Lei de Leibniz. Desta forma,  $[(lambda u [x]) (a) (\rightarrow) x u/a]$  somente se nenhuma



variável livre em  $a$  tornar-se ligada quando  $a$  for substituído por  $u$  em  $x$  e se  $u$  não estiver sob o escopo de  $\hat{\phantom{a}}$ ,  $\text{nec}$ ,  $P$ ,  $F$  em  $x$ . Como antes, a restrição pode ser eliminada sobre os operadores intensionais quando  $a$  tem uma intensão constante, isto é: quando  $a$  é  $\hat{b}$ ; ou quando  $a$  for uma variável. Isto porque uma variável, uma expressão denotando uma intensão, uma fórmula  $\text{nec } x$ , uma fórmula  $Px$  e uma fórmula  $Fx$  possuem extensões que não variam sob índices. No caso específico de  $F$  e  $P$ , o tempo não pode ter um início ou um fim, isto é: se  $t$  pertence a  $T$  então existem  $t_1$  e  $t_2$  tal que  $t_1 < t < t_2$ . Numa linguagem mais de natureza lógica, pode ser dito então que nestas condições  $x$  e  $a$  são modalmente fechados.

### 3.11 Alguns problemas na semântica das crenças

Quine /QUI 60/ colocou problemas importantes sobre o significado da quantificação em contextos intensionais; porém ele trabalhou com uma lógica modal anterior a Kripke. Montague, por seu turno, propôs-se a tratar a quantificação destes contextos de um modo adequado às linguagens naturais, trabalhando com verbos de atitude proposicional (como "acreditar"), operadores modais e adjetivos intensionais, de modo que as críticas de Quine aqui não valem. Em especial, Montague /MON 60/ acredita que uma análise semântica extensional de um predicado como "é vermelho" não é por si só adequada para analisar a locução "a propriedade de ser vermelho", porque o predicado extensional "é vermelho" se aplica só aos indivíduos correntes, enquanto que a propriedade de ser vermelho denota algo muito mais complexo /CRE 75/.

Existem contudo alguns problemas com a abordagem de Montague, especialmente no que diz respeito à análise de proposições (como conjuntos de índices ou como conjuntos de mundos possíveis) como objetos de crença, isto é: como objetos de verbos como "acreditar", "conhecer", "desejar", etc. Assim se duas sentenças  $x$  e  $y$  possuem a mesma intensão, então  $\text{Acr}(a, \hat{x})$  implica em  $\text{Acr}(a, \hat{y})$  para uma pessoa (denotada por  $a$ ) que acredita; isto é um problema porque existem casos onde uma pessoa não sabe que uma sentença é igual a outra e então ela acredita numa e não acredita na outra. Um caso interessante é o das sentenças necessariamente verdadeiras: pelo modelo, se uma pessoa acredita numa sentença necessariamente verdadeira, então ela acredita em todas as sentenças necessariamente verdadeiras. Isto faz com que esta pessoa não possa "descobrir" uma nova sentença necessariamente verdadeira. Este problema pode ser afastado se toda preocupação de modelagem psicológica é afastada (ver uma discussão em /MOS 83/).

Outro problema aparece se nomes próprios forem vistos como designadores rígidos /PUT 73/ /DON 74/, isto é: possuindo a mesma denotação em todos os mundos possíveis e em todos os tempos; ou seja: a intensão de um nome fica como uma função constante. Desta forma a verdade da sentença "Os antigos acreditavam que Hesperus era Hesperus" implica a verdade da sentença "Os antigos acreditavam que Hesperus era Phosphorus", porque os nomes Hesperus e Phosphorus nomeiam a mesma coisa e as sentenças "Hesperus é Hesperus" e "Phosphorus é Phosphorus" possuem a mesma intensão. Kripke e Putnam estenderam esta proposição a nomes de coisas naturais, como compostos químicos, espécies biológicas, etc. O problema aqui é que nenhuma pessoa conhece totalmente as intensões de todas as expressões de uma linguagem natural. A única saída aqui é que a semântica de Montague se aplique não a semântica que um falante/ouvinte tem na sua cabeça, mas se aplique a um falante/ouvinte idealizado que conheça toda uma linguagem natural.

Outro problema semelhante é o dos "mundos impossíveis" /CRE 73/: a questão é a aplicação a mundos onde  $x$  e " $x$ " não possuem o mesmo valor. Isto pode ser afastado se for aceito trabalhar só com lógica binária e se a lógica intensional, relacionada com a linguagem natural, também for binária.

Outro conjunto de problemas refere-se à idéia de isomorfismo intensional de Carnap. Isto aparece pelo tratamento feito por Lewis /LEW 70/ a alguns conceitos da semântica; assim ele define intensão como foi feito antes, mas acrescenta uma outra noção técnica que é "meaning". Para ele, num "meaning" existe a intensão da sentença total e a intensão de cada um de seus subconstituintes arranjados em correspondência à estrutura sintática da sentença. Assim fazendo, os nós superiores dos significados de duas sentenças logicamente equivalentes podem ser iguais mas os significados dos nós de baixo podem não ser iguais; isto é: sentenças equivalentes podem ter "meanings" diferentes.

Por outro lado, Lewis assume que o objeto de crença (para os verbos que envolvem atitudes proposicionais) é um "meaning" e não uma intensão. Sua proposta contudo apresenta problemas, pois não é possível distinguir com ela um par de sentenças onde as orações encaixadas possuam a mesma estrutura sintática, mas difiram só no que diz respeito a palavras sinônimas; por outro lado, há problemas sérios com a quantificação em contextos de crença como mostrou /BIG 78/.

Outra abordagem para a semântica de crenças considera o objeto da crença como sentença (ou nome de sentença), e não como intensão. Alguns argumentos contra esta teoria quotacional aparecem em /BIG 78a/ e em /PAT 73/.

#### 4. TRATAMENTO LÓGICO DA SENTENÇA DECLARATIVA, DA COORDENAÇÃO, DA ORAÇÃO RELATIVA E DAS PERGUNTAS

##### 4.1 Tratamento da sentença declarativa

Aqui são usadas as noções de símbolo complexo e categorias derivadas apresentadas no capítulo sobre questões formais. Cada símbolo complexo é um termo lógico, cujo predicado indica a categoria sintática principal, cujo primeiro argumento indica o nível desta categoria no sistema X-barra e cujos outros argumentos são traços especificados. Vale aqui a Convenção de Traços do Núcleo, apresentada antes:

se uma regra tem a forma  $D \rightarrow \dots d \dots$  onde  $d$  é o núcleo de  $D$ , então  $d$  carrega todos os traços associados com  $D$ , a qual é seguida na escrita da gramática.

Um exemplo de categoria sintática é então

$s(\text{decl}, \text{impass})$

que indica que esta categoria é composta pelas sentenças do tipo declarativo e do tipo impessoal. Outro exemplo de categoria sintática é

$v(3, \text{impass}, 3\text{psing})$

que indica que esta categoria é composta pelos verbos com três barras, do tipo impessoal e da terceira pessoa do singular.

A notação usada para as regras sintáticas deste trabalho seguem a proposta da DCG de Colmerauer /COL 75/, discutida antes, sendo que os argumentos que indicam o início e o fim da cadeia em questão não são indicados por comodidade. Assim

$s(\text{decl}, \text{impass}) :- v(3, \text{impass}, 3\text{psing}), \text{adv}(3, \text{sent})$

indica que uma sentença do tipo declarativo e impessoal pode ser analisada como um verbo três barras, do tipo impessoal e na terceira pessoa do singular, seguido por um advérbio três barras do tipo sentencial.

Em relação à interpretação semântica, a notação utilizada não segue, por comodidade, a notação em linguagem Prolog, visando uma maior facilidade de apresentação: a notação segue o padrão do PTQ /MON 73/. São usados contudo os símbolos "int  $x$ " para indicar a intensão de  $x$ , "ext  $x$ " para indicar a extensão de  $x$ , " $x$ " para indicar a interpretação semântica de  $x$ , "nec  $x$ " para indicar que o operador quadrado (necessidade) se aplica sobre a expressão  $x$ , sendo que qualquer outra notação diferente do PTQ é explicitada no texto.

Além disso, é utilizada muitas vezes a regra de aplicação funcional do PTQ /MON 73/

se  $a$  traduz-se em  $a'$  e se  $b$  traduz-se em  $b'$ , então  $F_i(a, b)$  traduz-se em  $a'(int\ b')$

devendo ser observado aí que " $a$ " e " $b$ " neste trabalho são símbolos complexos, tal que é notado

$(a(\dots))' (int\ (b(\dots)))'$ .

As sentenças declarativas com verbo impessoal colocam o verbo na terceira pessoa do singular. Elas cumprem a regra sintática (onde não aparecem os argumentos tradicionais das DCG que apontam posições na cadeia de entrada):

s(decl, impess) :- v(3, impess, 3psing).

cuja interpretação semântica é

(v(3, impess, 3psing))'

devendo aqui ser observado que, para o Português, a interpretação semântica de v(3, impess, 3psing) é igual a de uma sentença, isto é, vale <t>.

No caso de combinação com advérbio sentencial, valem as regras (o advérbio pode estar na frente ou depois do verbo):

s(decl, impess) :- v(3, impess, 3psing), adv(3, sent).

s(decl, impess) :- adv(3, sent), v(3, impess, 3psing).

que possuem a mesma interpretação semântica

(adv(3, sent))' (int (v(3, impess, 3psing)))'

pois há aqui apenas uma aplicação funcional da interpretação do advérbio sobre a intensão da interpretação do verbo. Deve ser observado que os advérbios sentenciais são do tipo <t, t> e que essa proposta é análoga à regra 9 do PTQ /MON 73/:

S9: se a pertence a P<t, t> e se b pertence a P<t>, então F6(a, b) = ab pertence a P<t>;

T9: se a pertence a P<t, t> e se b pertence a P<t>, e se a traduz-se como a' e b traduz-se como b', então F6(a, b) traduz-se como a'(int b').

Para as sentenças com verbos pessoais, valem as regras sintáticas

s(decl, pess, sing) :- n(3, sing), v(3, pess, sing).

s(decl, pess, pl) :- n(3, pl), v(3, pess, pl).

onde os traços "sing" (singular) e "pl" (plural) asseguram a concordância da locução nominal sujeito com a locução verbal predicado; este traço (sing ou pl) é passado, pela Convenção de Traços do Núcleo, para o verbo núcleo da locução verbal. A interpretação semântica correspondente a estas duas regras sintáticas é

(v(3, pess, ...))' (int (n(3, ...)))'. (onde int = ?)

No caso de combinação das sentenças dos verbos pessoais com advérbios sentenciais, valem as regras sintáticas

s(decl, pess, ...) :- adv(3), n(3, ...), v(3, pess, ...).

s(decl, pess, ...) :- n(3, ...), adv(3), v(3, pess, ...).

s(decl, pess, ...) :- n(3, ...), v(3, ...), adv(3).

onde todos "..." valem "sing" (ou "pl"), sendo que a interpretação semântica vale

(adv(3))' (int (v(3, pess, ...))' (int (n(3, ...)))' ) ).

Os respectivos tratamentos da locução nominal, da locução verbal e da locução adverbial são feitos a parte.

O conjunto destas regras gera as seguintes interpretações semânticas:

chove' para a sentença "Chove.",

possivelmente' (int chova') para a sentença "Possivelmente chova.",

certamente' (int (ama\_maria' (int pedro'))) para a sentença "Pedro certamente ama Maria", onde ama\_maria é analisado como depois e mostrado.

## 4.2 Tratamento da coordenação

Como primeira colocação, deve ser levado em conta que as Gramáticas Transformacionais nunca conseguiram dar um tratamento correto ao problema da coordenação /CHO 57/ /CHO 65/ /CHO 80/. Por outro lado, o formalismo ATN, como apresentado em /WOO 70/ /WOO 73/, tem sérios problemas para capturar as várias estruturas coordenadas. Para observar mais de perto isto, pode ser considerada a gramática ATN para uma simples locução nominal do tipo

LN → Det N Adj\* LP\*.

A gramática ATN para esta regra é:

(LN/

(cat det ... (to LN/det))

(jump LN/det ... ))

(LN/det

(cat n ... )

(cat adj ... (to LN/det))

(LN/pop

(push lp/... (to LN/pop))

(pop (LN/build) ... ))

e para tornar explícitas todas as construções coordenadas de locução nominal, derivadas de uma simples especificação de locução nominal, é necessário adicionar novos arcos na rede, tais como:

(LN/... )

(LN/det

(cat n ... )

(cat adj ... )

(wrđ e (to LN/ ... ))

que atinge por exemplo a construção "o gato faminto e fraco",

(LN/pop

(push lp/ ... )

(wrđ e (to LN/ ... ))

(wrđ e (to LN/det ))

(wrđ e (to LN/pop ))

(pop (LN/build) ... ))

onde a terceira linha atinge "os gatos e os cachorros", a quarta linha atinge "os gatos e os cachorros famintos", e a quinta linha atinge "os gatos com fome e os cachorros com sede".

De um modo geral, então, é necessário um arco "wrđ e" para fazer com que o interpretador ATN vá de cada estado anterior na rede para o estado que o precede no mesmo nível. Isto complica, por exemplo, a formalização em ATN de certos tipos de construção coordenada, em especial as construções com buracos como em "João planejou e Pedro escreveu o programa".

Algumas soluções foram propostas para resolver este problema. Assim Blackwell /BLA 81/ propõe que um único arco "wrd e" pegue o interpretador do estado final do primeiro elemento coordenado para o estado inicial do segundo elemento coordenado, de tal modo que ele esteja pronto para analisar o segundo argumento da estrutura coordenada na segunda passagem pela rede. Isto requer que o processamento seja suspenso com buracos e que estes buracos sejam preenchidos: o processamento a cada nível pode ser interrompido pela palavra "e" e, depois da estrutura coordenada ter sido analisada, os apagamentos em cada um dos constituintes unidos por "e" são vistos. Blackwell /BLA 81/ na verdade implementa, quanto a isso, duas restrições ao formalismo ATN: a restrição da direcionalidade e a restrição da periferia.

A primeira restrição implementada assume a Assunção da Redução na Coordenação, pela qual: o apagamento nos constituintes unidos por "e" deve ser feito somente sob a condição de identidade nas estruturas coordenadas. A restrição com isto estabelece, na implementação, que os elementos dos arcos direitos são apagados para trás, e que os elementos dos arcos esquerdos são apagados para frente. Isto faz com que os estados finais que a Redução de Coordenação opcionalmente apaga, estejam constituídos (sob um nodo imediatamente dominante de cada vez) em só uma direção somente se o alvo apagado é periférico ao nodo imediatamente dominado; isto é: o substantivo não pode ser apagado se ele estiver imprensado entre dois outros constituintes do tipo "det" e "adj", exceto se um destes constituintes o acompanhar.

Estas restrições ajudam a decidir quando copiar e qual registrador copiar, dependendo estas decisões de qual construção coordenada foi derivada através do apagamento para frente ou para trás. Além disso, Blackwell /BLA 81/ faz com que diferentes conjuntos de registradores ATN sejam associados com constituintes diferentes.

É importante aqui observar que quando um "e" é encontrado de um estado Pop final, o sistema deve esperar para ter preenchido os buracos causados pelo apagamento. A restrição de direcionalidade prediz que os registradores relacionados ao primeiro constituinte devem ser copiados de novo e vice-versa; por outro lado, a Restrição da Periferia diz que, se os registradores correspondentes aos elementos de um lado do constituinte forem preenchidos, então os constituintes não podem sofrer apagamento, e devem não ser preenchidos por nenhum outro registrador se não existir um apontador já construído na rede para este local.

O grande problema da solução estabelecida por Blackwell /BLA 81/ é que ela implica na duplicação de quase todos os registradores em todos os níveis da rede, além de requerer a introdução de arcos não naturais do tipo Jump para impor restrições de direcionalidade no apagamento para trás. Na verdade, isto bloqueia o processamento de um constituinte e começa uma computação paralela para uma locução similar. De um modo mais geral, a abordagem de /BLA 81/ curva o formalismo ATN a restrições de programação procedural, sendo sob este aspecto uma aplicação não natural para um instrumento de escrita da gramática.

Um outro problema com a estratégia de Blackwell /BLA 81/ é que ela não pode ser estendida para atingir as conjunções reduzidas do tipo "Pedro ofendeu e bateu em Antonio". No entanto estas construções não parecem ser implementáveis num formalismo ATN, porque as redes ATN fornecem apenas um processo limitado de comunicação entre os níveis (ações SENDR e ações LIFTR) e utilizam apenas uma única saída para um certo nível de processamento (arco Pop).

O problema básico aqui é que o reconhecimento de fragmentos ligados de tipos diferentes requerem que o processamento atual seja bloqueado, e iniciado outro através de uma subrede diferente, num nível diferente. E não existe tal facilidade num formalismo ATN.

Quanto a isso, Woods /WOO 73/ sugere que, quando o interpretador encontra um "e", ele passe a operar de um modo diferente. A proposta é que ele suspenda o processamento corrente e tente atingir um arco de algum ponto precedente de computação que seja selecionado de um modo não determinístico, processando assim o que se segue ao "e". O que se segue ao "e" é processado então até um ponto na cadeia de entrada onde a computação deve ser bloqueada e a computação aí pode atingir o texto subsequente. Esta abordagem é especialmente adequada para a análise de conjunções reduzidas, onde as subcadeias do lado esquerdo e do lado direito dos dois constituintes conjugados foram fatoradas, deixando construtos reduzidos.

Esta abordagem envolve contudo muito esforço para se tornar uma abordagem geral, porque o tratamento de todas as estruturas coordenadas como casos especiais de conjunções reduzidas acarretaria uma explosão combinatória no processamento. Pode ser suposto aqui que Woods na implementação provavelmente foi forçado a algum sacrifício em prol da eficiência do processamento no ambiente em que o processador LUNAR opera, porque este processamento requer uma análise profunda para reconstruir o processo de apagamento.

Uma tentativa de solução, dentro do formalismo ATN, seria assumir que, uma vez que "e" seja detectado, seja possível estabelecer certas expectativas de um modo top-down, pois o que segue a "e" deve ser um constituinte categorialmente idêntico ao que está sendo processado. Avancando nesta idéia, olhando então para os arcos "wrd e", em cada nível de uma ATN, pode ser visto que todo arco "wrd e" retorna para o início de um constituinte categorialmente idêntico ao que foi analisado. A solução seria então postular um daemon que se tornasse ativo quando "e" fosse encontrado; este daemon suspenderia o processamento normal, inspecionaria o contexto corrente (os registradores locais que guardam os constituintes reconhecidos neste nível) e a história recente do processamento, e usaria esta informação para construir dinamicamente um novo arco ATN que procurasse reconhecer um constituinte categorialmente semelhante a um imediatamente terminado no processamento. Isto tudo envolveria construir um arco "cat" que olhasse para uma palavra pertencente a uma categoria lexical específica ou, na falta disto, que construísse e avaliásse um arco Push (cujo argumento seria deduzido da história do processamento).

Se isto falhasse, seria necessário construir um novo arco, desempenhar de novo um constituinte do nível corrente da computação, e assim sucessivamente. Ao todo, este mecanismo deveria ser tal que resolvesse os apagamentos para frente ("as casas com portas e + + com janelas"), para trás ("as grandes + + e as austeras casas") e para frente e para trás ("as grandes + + e + + austeras casas da Rua da Praia"). O problema com esta proposta é que o processamento das conjunções reduzidas envolveria a existência de um segundo daemon: ele esperaria pelo processamento iniciado pela avaliação do arco e iniciaria depois uma busca do constituinte esperado no ponto da primeira ativação do daemon /BOG 83/. É necessário, além disso, que a ativação deste segundo daemon construísse uma estrutura do tipo [e constituinte1 constituinte2] e que o processamento normal ATN fosse reimplantado. Porém, mesmo assim, o fenômeno do "gapping" deveria ser processado separadamente.

A solução aqui defendida é extremamente mais simples. Ela assume que somente constituintes categorialmente idênticos podem ser conjugados /RAD 81/, onde o termo categoria cobre as categorias principais e as categorias derivadas.

Uma primeira proposta seria as regras tipo DCG

$a(\dots) \rightarrow a(\dots), \dots, [e], a(\dots)$

$a(\dots) \rightarrow a(\dots), \dots, [ou], a(\dots)$

onde "a" é uma categoria sintática e "[e]" e "[ou]" denotam a conjunção de coordenação colocada entre o penúltimo e o último a.

Porém Ross /ROS 67/ apresentou razões sintáticas para que o coordenador seja colocado como um constituinte junto com o nó imediatamente seguinte, de tal forma que este último não deve ser visto como um irmão de todo o conjunto.

Para implementar esta proposta, uma estratégia é supor que os nomes dos coordenadores aparecem como traços de categorias e que estes traços sejam eliminados por meio de uma regra que expande esta categoria como coordenador e categoria. Isto é:

$a(3, e, \dots) :- a(3, \dots), a(3, e, seg, \dots)$ .

$a(3, ou, \dots) :- a(3, \dots), a(3, ou, seg, \dots)$ .

onde "seg" denota "seguinte", com as respectivas interpretações semânticas

$e'((a(3, \dots))', (a(3, e, seg, \dots))')$

$ou'((a(3, \dots))', (a(3, ou, seg, \dots))')$

onde  $e'(x' y')$  é só uma variação notacional do  $[x' \& y']$  do PTQ.

Além disso deve ser colocado que

$a(3, e, seg, \dots) :- [e], a(3, \dots)$ .

$a(3, ou, seg, \dots) :- [ou], a(3, \dots)$ .

sendo que a interpretação semântica para estas regras vale

$a'$ .

Levando em conta a expansão para a direita, devem ser acrescentadas as regras

$a(3, e, seg, \dots) :- [e], a(3, e, seg, \dots)$ .

$a(3, ou, seg, \dots) :- [ou], a(3, ou, seg, \dots)$ .

onde as interpretações semânticas respectivas valem

$(a(3, e, seg, \dots))'$

$(a(3, ou, seg, \dots))'$ .



Para as expressões sem coordenação, é necessário existir regras do tipo

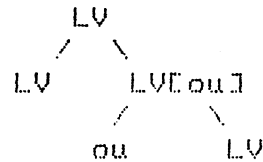
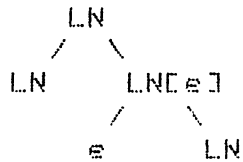
$a(3, \dots) :- \dots$

as quais são as regras comuns da gramática.

Deve ser observado que a existência do argumento "seg" impede a existência de uma locução do tipo [e a] como uma forma de expressar a categoria "a".

Em segundo lugar deve aqui ser ressaltado que esta proposta assume que "a" pode ser uma categoria principal (s, sí, n3, n2, n1, n0, etc) ou mesmo uma categoria derivada (s/n, v3/n).

A utilização dos dois esquemas gerais mostrados acima permite as árvores seguintes:



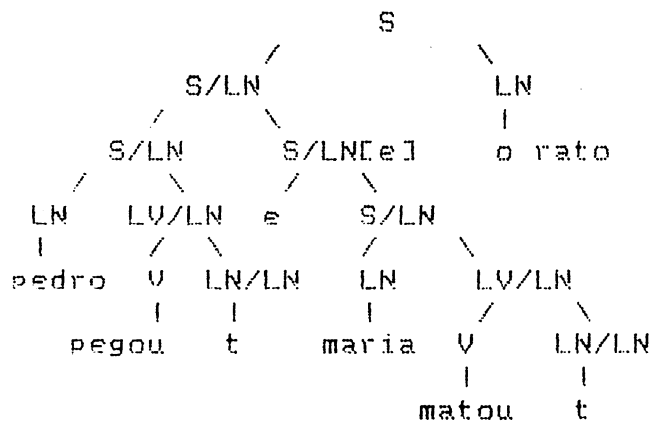
Um efeito desta proposta é que o esquema de coordenação só pode ser aplicado a elementos de uma mesma categoria sintática: assim não é possível coordenar LV e LV/LN e daí a Restrição da Estrutura Coordenada de Ross /ROS 67/ não é mais necessária. No entanto, esta proposta permite coordenar uma S/LN com uma S/LN, como em "Pedro leu um livro que Paulo viu e Maria copiou".

Supondo o seguinte formato para as regras que expressam dependência não ligada para a direita

$A \rightarrow A/B B$

$\lambda h [ (A/B)' ] (B')$

onde A varia sobre a categoria da sentença e B é uma categoria sentencial (S ou S1) ou locucional (LN, LV, LP), então a sentença "Pedro pegou e Maria matou o rato" possui a seguinte árvore



onde t denota um nodo vazio.

Um problema com a proposta acima é que ela não gera uma sentença como "? João disse e Pedro mostrou a Paulo que Maria era louca" (que é discutível), porque o esquema acima prediz que somente constituintes podem aparecer numa posição B à direita.

### 4.3 Tratamento da oração relativa

A extraposição é um fenômeno linguístico que ocorre basicamente nas linguagens naturais do tipo SVO (sujeito-verbo-objeto), como foi mostrado por Kuno /KUN 72/ e Funk /FUN 83/. O Português é uma língua deste tipo, e o fenômeno então deve merecer um tratamento pelo processador.

O tratamento da extraposição a esquerda em Prolog é feito em Pereira /PER 83/. Ele assume que uma sentença como "O homem que João conheceu é carioca" tem como estrutura (simplificada):

```
o homem que [João conheceu t] é carioca
onde "t" representa um traço.
```

Uma proposta inicial, numa gramática livre de contexto, para este tipo de estrutura envolveria regras como

```
n -> traço
traço -> []
```

e daí numa DCG as regras deveriam ser

```
sentença(Buraco0) :- ln(Buraco0, Buraco1), lv(Buraco1).
ln(Buraco, Buraco) :- det, n, relat.
ln(traço, nil) :- traço.
traço :- [].
lv(Buraco) :- verbo, ln(Buraco, nil).
lv(nil) :- verbo.
relat :- [que], sentença(traço).
relat :- [].
```

onde a variável "Buraco" tem como valor "traço" se a locução nominal extraposta ocorrer no lado direito da regra e "nil" se não ocorrer.

A partir daí, é possível conceber o movimento da locução nominal para a esquerda, deixando um traço, ou o traço aparece de um modo marcado e o movimento da locução ocorre para a direita. Assim na cláusula

```
ln(Buraco1, Buraco2) :- det, n, relat.
```

"Buraco1" terá o valor "traço" quando o traço ocorrer em algum lugar a direita do limite esquerdo da locução. Além disso, "Buraco2" terá como valor "nil" se a locução nominal contiver o traço, e terá como valor "traço" se o traço aparecer à direita do limite direito desta locução nominal. Estas regras estabelecem, em conjunto, então que a locução nominal expande-se num traço se um traço aparece a sua esquerda.

O não terminal "relat" não possui argumentos aqui porque uma locução nominal não poderia mover-se para fora de uma oração relativa /ROS 67/.

Pereira /PER 83/ argumenta contudo a favor das Gramáticas de Metamorfose, a qual permite um tipo de regra cujo lado esquerdo pode ser composto por um não terminal seguido por uma cadeia de símbolos terminais vazios, que não ocorrem no vocabulário de entrada. Esta regra seria algo como

```
marcador_de_relativa, [t] -> pronome_relativo
```

significando que o pronome relativo pode ser analisado como um "marcador\_de\_relativa" provido com o terminal "t" que é adicionado ao início da entrada, sendo que "t" resta após a aplicação da regra.

De um modo mais claro, para Pereira /PER 83/, uma oração relativa pode ser vista com um marcador "r1" seguido de uma sentença "s", onde

ln -> det, n, relat.

relat -> [].

relat -> r1, s.

O marcador "r1" pode ser expandido como um sujeito extraposto, e assim

r1 -> r2

ou pode ser expandido como um complemento de verbo extraposto

r1, [s\_núcleo, argumento\_movido] -> r2, arg, s\_núcleo.

A regra

r2, [arg] -> pronome\_relat.

define o marcador "r2", analisando um "pronome\_relat" como um marcador "r2" e um terminal vazio "arg" a ser colocado na frente da entrada remanescente.

Usando as regras

r1 -> r2.

r2, [arg] -> pronome\_relat.

a locução nominal extraposta "arg" é movida da posição complemento, jogada para fora à esquerda do sujeito e do verbo "s\_núcleo" da oração relativa, sendo depois pega através da regra complemento -> argumento\_movido, complementos.

Além disso, é necessário

sentença -> s-núcleo, complemento

e regras que movem a locução nominal extraposta para sobre o objeto direto.

Em suma, a solução proposta por Pereira /PER 83/, chamada por ele de Gramática de Extraposição (XG), envolve a descrição de um relacionamento não contíguo de partes da sentença a partir de regras do tipo

sentença -> ln, lv.

ln -> det, subst, relat.

ln -> traço

relat -> []

relat -> marcador\_relat, sentença

marcador\_relat ... traço -> pron\_relat

onde todas as regras, exceto a última, são livres de contexto. A última regra estabelece que um pronome relativo é analisado como um marcador, seguido por algum constituinte desconhecido (denotado por "..."), seguido por um traço.

Assim a sentença "O rato que o gato cacou morreu" é vista como

o rato marcador\_relat o gato cacou traço morreu

e daí como

o rato pronome\_relat o gato cacou morreu.

A implementação da proposta de Pereira /PER 83/ em Prolog envolve o acréscimo de mais dois argumentos aos predicados das cláusulas DCG: além dos dois argumentos da DCG que representam as posições na cadeia, existem mais dois argumentos que são usados para representar as posições na lista de extraposição, sendo que esta lista carrega os elementos a serem reposicionados. Assim cada elemento da lista de extraposição representa um símbolo a ser reposicionado como uma quádrupla

x (contexto, tipo, símbolo, xlista)  
 onde  
 "contexto" é "buraco" se o símbolo é precedido por "..." na regra onde ele se origina, e "não\_buraco" se o símbolo é precedido por ",";  
 "tipo" pode ser terminal ou não terminal, dependendo se "símbolo" é um terminal (marcado por "[" e "]"), ou um não terminal no lado esquerdo da regra de onde ele vêm;  
 "símbolo" é o símbolo propriamente dito;  
 "xlista" é o resto da lista de extraposição.  
 Assim a regra

s -> ln, lv  
 é traduzida como  
 s(S0, S, X0, X) :- ln(S0, S1, X0, X1), lv(S1, S, X1, X).

Por outro lado, um terminal "t" no lado direito da regra traduz-se pela chamado do predicado "terminal", cujo análogo é o predicado "conecta" na DCG. Assim  
 pronome\_relat -> [que]

fica como  
 pronome\_relat(S0, S, X0, X) :- terminal(que, S0, S, X0, X).

A tradução de uma regra com mais de um símbolo do lado esquerdo é mais complicada, tal que a regra  
 marcador\_relat ... traço -> pronome\_relat  
 é traduzida por

marcador\_relat(S0,S,X0, x(buraco, não\_terminal, traço, X)):-  
 pronome\_relat(S0, S, X0, X).

Para cada não terminal "nt" do lado esquerdo da regra X0, a tradução inclui a cláusula

nt(V1, ..., Vn, S, S, X0, X):-  
 virtual(nt, (V1, ..., Vn), X0, X).

onde "virtual(C, X0, X)" pode ser lido como "C é um constituinte entre X0 e X na lista de extraposição" e as variáveis do tipo "Vi" transferem os argumentos do símbolo na lista de extraposição para o predicado que traduz o símbolo. A definição de "virtual" é  
 virtual(NT, x(C, não\_terminal, NT, X), X).

e a de "terminal" é  
 terminal(T, S0, S, X, X):- buraco(X), conecta(S0, T, S).  
 terminal(T, S, S, x(C, terminal, T, X), X).

buraco(X, (buraco, T, S, X)).  
 buraco([]).

onde a cláusula para "virtual" extrai um não terminal NT da lista de extraposição e retorna o resto da lista X. A primeira cláusula para "terminal" consome o terminal T da cadeia de entrada S0 e retorna o resto S da cadeia, de modo que a corrente lista de extraposição X permite que um buraco apareça na derivação. O predicado "conecta" toma o próximo terminal da entrada e o predicado "buraco" testa se existe um buraco na lista de extraposição.

A segunda cláusula para "terminal" consome um terminal T da lista de extraposição, e não da cadeia de entrada. Para isso, o primeiro símbolo da lista corrente de extraposição deve ter o tipo "terminal", e o resto X da lista de extraposição deve retornar; a cadeia de entrada S é retornada sem alteração.

A primeira cláusula para "buraco" estabelece que um buraco é verdadeiro se o primeiro símbolo na corrente lista de extraposição tem "buraco" como seu traço contextual.

A segunda cláusula para "buraco" estabelece que um buraco é sempre verdadeiro se a lista corrente de extraposição é vazia, isto é: não existe extraposição e o uso dos terminais é como nas regras DCG.

O tratamento das orações relativas apresentado a seguir utiliza apenas regras livres de contexto. As discussões formais estão na seção seguinte.

Aqui é assumido que as orações relativas são dominadas por uma categoria sentencial R diferente de S, tal que as categorias sentenciais são distinguidas pelos seguintes traços

C - oração complemento  
R - oração relativa  
Q - oração interrogativa

Neste caso,

[-C, -R, -Q] denota S  
[+C, +R, -Q] denota R  
[-C, -R, +Q] denota interrogativa matriz  
[+C, +R, +Q] denota interrogativa encaixada

Assumindo que [+C, +R, -Q] somente pode ser expandido como [que S] (o "que" não é opcional), então Si pode ser usado como abreviação de [+C, +R, -Q]. De outro lado, sujeitos sentenciais topicalizados e orações extrapostas devem ser [+C].

A regra que introduz R é

$n(S, r, \dots) :- n(S, \dots), r(\dots).$

e a interpretação semântica é

$\lambda R [ (n(S, \dots))' ] (R')$

a qual assume que toda locução nominal possui um conjunto R livre de variável. A interpretação semântica acima abstrai sobre esta variável e aplica a função resultante sobre o conjunto denotado pela oração relativa. Logo a interpretação da oração relativa é quantificado sobre a interpretação da locução nominal.

As regras seguintes expandem R, sendo colocado junto a interpretação semântica:

$r(\dots) :- n(r, pro, \dots), s/n(\dots).$

$\lambda x [ \lambda h [ (s/n(\dots))' ] (n(pro, \dots))' \& R(x) ]$   
onde h varia sobre n(pro, ...)

$r(\dots) :- p(wh, pro, \dots), s/p(\dots).$

$\lambda x [ \lambda h [ (s/p(\dots))' ] (p(wh, pro, \dots))' \& R(x) ]$

onde h varia sobre p(wh, pro, ...) e além disso

$n(r, pro, \dots) :- [que].$

$\lambda p p(x)$

que vale para "O livro que Pedro pegou é de João" e

$p(wh, pro, \dots) :- [em], [quem].$

$\lambda p p(x)$

A interpretação semântica das regras dadas acima liga a variável  $h$  na tradução de  $S/LX$  e aplica a função resultante sobre  $LX'$ , o qual contém em si uma ocorrência livre da variável designada  $x$ . Esta variável  $x$  é depois ligada por abstração para formar uma expressão que denota um conjunto.

No caso, a denotação de uma locução nominal com uma oração relativa apositiva (não restritiva) como "Cabral, que descobriu o Brasil, era português", pode ser representada como

$\{x \mid (a \mid a \text{ descobriu o Brasil}) \text{ pertence (Cabral)} \ \& \ x \text{ pertence (Cabral)} \}$

isto significando que a locução nominal total tem a mesma denotação que "Cabral" se a denotação da oração relativa estiver incluída nela. De outro modo, a denotação será vazia ("Cabral, que descobriu a América, era português").

Deve ser observado que as relativas restritivas modificam substantivo ( $n(\emptyset, \dots)$ ) e que as apositivas modificam locução nominal ( $n(3, \dots)$ ). O problema é que as apositivas não podem modificar qualquer tipo de locução nominal:

?Cada homem, que conhece Carlos, trabalha.

\*Nenhum homem, que conhece Carlos, trabalha.

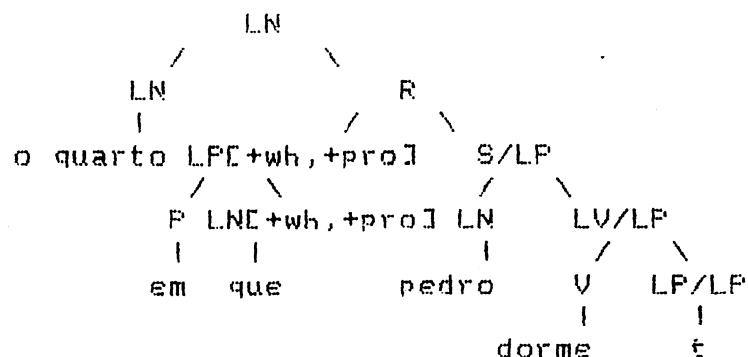
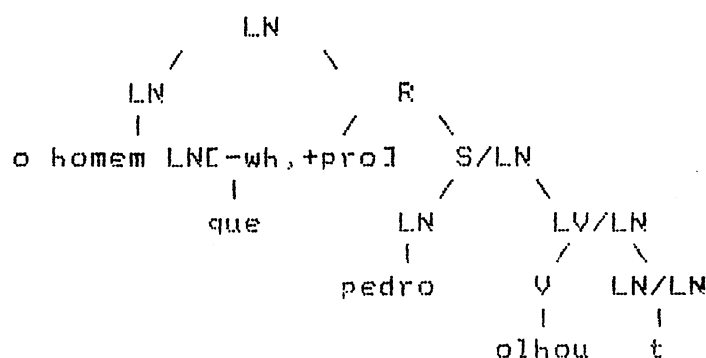
Um homem, que conhece Carlos, trabalha.

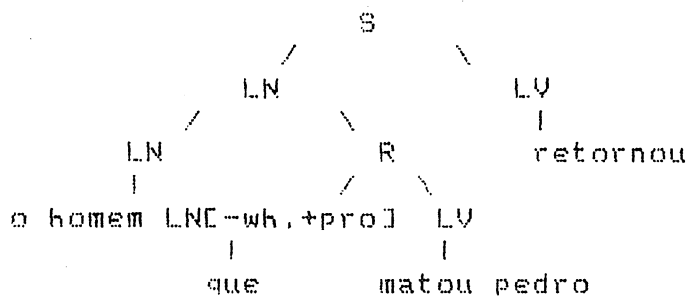
\*Ao menos um homem, que conhece Carlos, trabalha.

\*Todo homem, que conhece Carlos, trabalha.

Devido a estas dificuldades a diferenciação entre restritivas e apositivas não é feita aqui.

As regras mostradas geram as árvores de análise





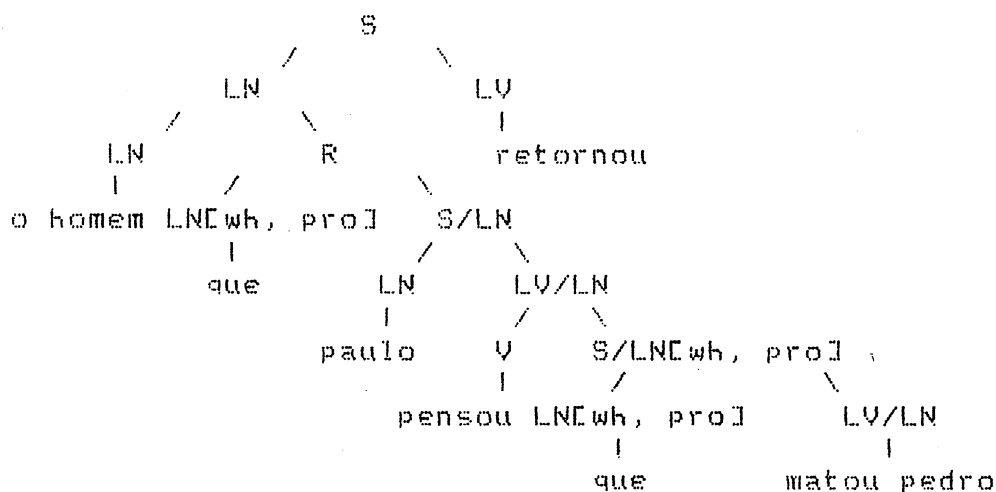
Acrescentando as regras

v/n(...) :- v(...), r/n(...).

v/n(...) :- v(...), n(...), r/n(...).

r(...) :- n(+wh, +pro, ...), v(...).

é possível gerar a árvore



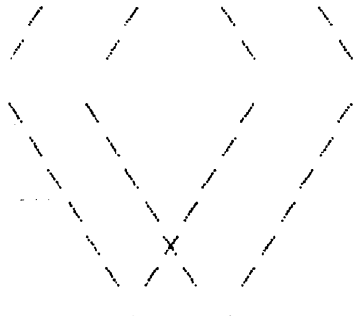
para a sentença "O homem que Paulo pensou que matou Pedro", devendo ser observada a estrutura gerada para a oração relativa mais internamente encaixada.

#### 4.4 Dependências nas árvores

Pereira /PER 83/ mostra que uma XG pode gerar a linguagem

$(a \text{ exp } n \ b \ \text{exp } n \ c \ \text{exp } n \ | \ n > 0)$

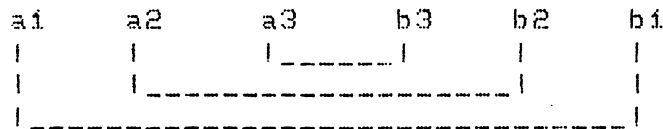
de modo que a classe das gramáticas XG gera linguagens dependentes de contexto que não podem ser geradas por gramáticas livres de contexto. Além disso, numa XG, na árvore de derivação, cada nodo pode ter mais do que um nodo pai, mas a classe das gramáticas XG obedece à restrição do cruzamento de ramos, não podendo ocorrer



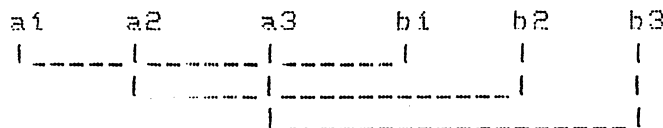
Em relação a isso, Joshi /JOS 85/ mostra que nas gramáticas livres de contexto as dependências também não podem cruzar: ou elas são disjuntas, como em



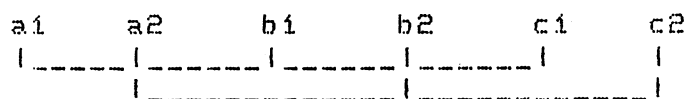
ou as dependências são encaixadas (esta é a diferença para com os conjuntos regulares), como em



No caso das TAGs, Joshi /JOS 85/ mostra que as dependências podem ser serialmente cruzadas, como em



mas não podem ocorrer entre mais do que dois elementos, como em





O problema geral aqui diz respeito à colocação de restrições na associação entre buracos e preenchedores de buraco /MOS 81/ /MOS 82/ /MOS 85/, que sejam linguisticamente relevantes. Aqui é argumentado que as restrições dentro do campo das gramáticas livres de contexto são suficientes.

No que diz respeito ao tratamento em processador das dependências entre preenchedores de buracos e buracos, deve ser lembrado que uma ATN possui uma memória especial (HOLD) na qual o preenchedor de buraco pode ser mantido até que seu buraco seja identificado na sentença. (No caso de extraposição à direita, o endereço do buraco deveria ser estocado em HOLD até que o preenchedor seja encontrado.) Isto corresponde, de certo modo, à regra transformacional que move, por exemplo, uma locução wh para a posição do complementizador: assim, existe uma rotina que pega a locução wh, identifica-a como na posição COMP na pilha HOLD, e depois uma rotina recupera esta locução de HOLD quando o processador encontra um buraco na cadeia de palavras correspondente a sua posição na "estrutura profunda". (Para as transformações que realizam apagamento, as rotinas de processamento deveriam fazer uma cópia do antecedente em HOLD e recuperar esta cópia quando o local do apagamento for identificado.) A saída do processador é então a "estrutura profunda" da sentença.

Já no sistema PARSIFAL de Marcus /MAR 80/ existe também o uso de memória temporária para relacionar preenchedores de buracos a seus respectivos buracos, mas os mecanismos são diferentes. Isto é assim porque PARSIFAL não representa um preenchedor de buraco, mas sim sua posição superficial na árvore, isto é, PARSIFAL cria simultaneamente um traço co-indexado ao preenchedor e insere este traço no buffer que é usado para estocar palavras e constituintes da cadeia de entrada. O traço permanece no buffer até que o processador atinja a posição do buraco, e é então processado como se ele aparecesse nesta posição. A saída deste processador é então uma representação da "estrutura superficial", aumentada com traços, da sentença.

Na abordagem aqui desenvolvida, não há uso de memória especial para a estocagem de preenchedores de buraco e de buracos, porque estes são estocados como partes da própria árvore de análise da sentença que o processador está concorrentemente computando e estocando. O problema é que um processador não pode associar um preenchedor de buraco com qualquer buraco que ele ache. Assim existem restrições sobre que preenchedores de buraco devem ser associados com quais buracos.

Deve ser recordado que as teorias transformacionais usam fundamentalmente cinco tipos de restrição, aqui apresentadas de modo informal (uma discussão formal aparece em /MOS 84b/ /MOS 84d/ /MOS 84e/ /MOS 84g/ /MOS 84j/ /MOS 84i/:

- a condição de comando, que requer que um preenchedor pode somente ser associado a um buraco que ele comande;
- as restrições de ilha, que proibem que o buraco apareça sob certos constituintes designados dentro do domínio de comando do preenchedor;
- certas restrições especificadas sobre posições de buracos, ligadas ao governo lexical;
- certas restrições de não-ambiguidade.

A seguir é mostrado que estas restrições são de difícil implementação numa ATN e, por outro lado, podem ser colocadas diretamente nas regras livres de contexto do processador no caso do modelo aqui proposto.

As restrições de comando e as restrições de ilha caracteristicamente limitam as posições de um buraco em relação à posição do preenchido deste buraco. A forma de implementação destas restrições num formalismo ATN envolveria fazer com que a recuperação de um preenchido na memória pudesse ser sensível a certas condições definíveis sobre a árvore sintagmática. Isto acarretaria um procedimento bastante complexo e o desejável seria que, quando um preenchido que estivesse na memória ficasse acessível ou não através do processamento, isto fosse feito de acordo com as restrições supracitadas: isto implicaria em que as regras de estrutura de frase deveriam ser usadas para controlar a acessibilidade de um preenchido.

Uma saída aqui é que o estoque na memória seja estruturado em níveis, onde um nível corresponde a cada nível da estrutura no marcador frasal; aí um preenchido poderia então ser inserido na memória em um nível que refletisse seu "peso" na árvore. Para capturar a condição de comando, deveria ser estipulado que todos os níveis da memória abaixo do nível correspondente ao "peso" de um certo buraco ficassem inacessíveis, enquanto o buraco esteja sendo processado.

Deveria ser colocado ainda que a memória HOLD deveria estar completamente vazia no término do processamento de uma oração relativa. Por outro lado, as restrições de ilha devem ser impostas de modo a fechar o acesso aos níveis altos do estoque no início do processamento de um constituinte "ilha", e o acesso deveria ser restaurado aos níveis somente quando o processamento da ilha esteja completo. Uma saída aqui seria enriquecer a estrutura de estoque, incluindo um conjunto de regras de tráfico para controlar o acesso a seus vários níveis.

Em oposição às dificuldades referidas acima, para um processador que utiliza a maquinaria exposta aqui, as condições de comando e as restrições das ilhas podem ser impostas de modo fácil. A diferença é que as restrições acima não se aplicam diretamente às associações preenchido-buraco nas sentenças, mas possuem seus efeitos na determinação de quais estruturas frasais a gramática pode conter e quais regras podem ser usadas pelo processador. Isto ocorre porque a coleção completa de regras de estrutura de frase da gramática consiste num conjunto de regras básicas junto com um conjunto de regras de ligação (introduzidas pela notação X/Y).

Deve ser lembrado aqui que uma notação de buraco do tipo X/Y não pode desaparecer durante a derivação: ela deve ser passada para um nível mais baixo do marcador frasal ou deve ser realizada como um traço Y/Y. Isto é equivalente no efeito à condição de que um preenchido na memória deve ser usado por um certo tempo do processamento.

A solução simples, apresentada neste trabalho, é que a condição de comando e as restrições de ilha podem ser impostas apenas pela limitação da classe das regras derivadas e da classe das regras de ligação. Assim por exemplo

\* S → S/LN S/(S/LN)

não deve aparecer na gramática.

A condição de regra de ligação para uma dependência não governada deve ser explícita assim nesta regra através da presença de um nodo irmão ao nodo com a "categoria barrada". Isto é, uma regra do tipo [S → LN S/LN] é permitida, mas não uma regra do tipo \* [S → S/LN] ou do tipo \* [S → LP S/LN]. Uma restrição, quanto a isso, discutida antes é a Condição Generalizada de Restrição à Esquerda /MOS 85/.

Em relação às dependências governadas pelo léxico, um processador ATN deveria tentar todas as posições potenciais de buracos e ver se a restrição é satisfatória, mas isto seria claramente ineficiente numa sentença como "Que voce fez para persuadir Marta a levar Pedro a cumprimentar voce na sala?", onde o preenchedor wh seria incorretamente assinalado ao buraco Equi e o erro só seria detectado no final da oração.

Na proposta apresentada aqui, os verbos da classe Equi podem ser especificados na gramática como ocorrendo com complemento LV1 mais do que com complemento S com buraco LN. Assim nenhum buraco Equi ocorre e nenhum preenchedor Equi em HOLD é acessado. A ligação entre o sujeito ausente do verbo subordinado e o sujeito do verbo matriz deve ser estabelecido contudo na regra semântica: esta é, no fundo, a análise não transformacional proposta por Brame /BRA 76/.

A análise tipo LV1 é muito natural na análise aqui desenvolvida, porque a idéia fundamental é que um preenchedor e seu buraco sejam ligados por um caminho marcado nos nodos: este caminho passa através dos ramos da árvore (via passagem da notação X/Y) e não existe um "caminho horizontal" sobre a árvore como faz a aplicação de uma transformação.

O fato que certos constituintes (como substantivos e determinantes) não podem ser extraídos da locução nominal, pode ser capturado pelo impedimento de nodos da forma X/N ou X/DET. Por outro lado, o fato de que uma locução preposicional não pode conter um preenchedor anterior a ela é capturado facilmente pelo impedimento de regras do tipo \* [LP → X LP/X]. A exclusão de categorias barradas, a exclusão de regras de estrutura de frase e a exclusão de traços de subcategorização representam a soma total de esquemas viáveis na proposta aqui apresentada, para restringir a distribuição de preenchedores e buracos.

De um modo geral, então, um processador com uma memória externa (caso da ATN) pode lidar com restrições estruturais sobre a distribuição dos buracos somente se esta memória for estruturada em níveis correspondentes aos níveis do marcador frasal. Em oposição, um processador, nos moldes do que aqui é proposto, usa os nodos do marcador frasal para estocagem dos buracos e então sua "memória" necessariamente cumpre o requisito da estrutura em níveis. Em segundo lugar, num processador com memória externa (ATN) as restrições sobre a distribuição dos

buracos requer um controle complexo sobre a acessibilidade do preenchedor, enquanto o processador aqui proposto não apresenta este problema. Em terceiro lugar, algumas restrições especiais sobre regras governadas pelo léxico complicam o mecanismo para controle ao estoque externo de preenchedores, enquanto que o processador aqui proposto utiliza esquemas como os traços de subcategorização para impor estas restrições. Deve ficar claro aqui que estes pontos seguem naturalmente da decisão de não permitir condições nas derivações que sejam externas à seleção de regras (livres de contexto) da gramática.

No que se refere aos aspectos formais, /MOS 85/ propõe uma restrição para as dependências encaixadas, a qual impede ambiguidades, formulada da seguinte forma:

se existem duas ou mais dependências preenchedor-buraco numa mesma sentença, seus escopos não podem interseccionar (elas devem ser disjuntas ou encaixadas).

Assim, para P1 e P2 referindo dois preenchedores diferentes e para B1 e B2 referindo dois buracos diferentes tal que P1 e B1 possuem dependência e P2 e B2 possuem dependência, não são gramaticais as construções:

P1 P2 B1 B2  
B1 B2 P1 P2  
P1 B2 B1 P2  
B1 P2 P1 B2

e, em oposição, as construções seguintes são gramaticais:

P1 P2 B2 B1  
P1 B2 P2 B1  
B1 B2 P2 P1  
B1 P2 B2 P1  
P1 B1 B2 P2  
P1 B1 P2 B2  
B1 P1 B2 P2  
B1 P1 P2 B2

Para um processador com memória externa para preenchedores e buracos (ATN), o acúmulo dos Ps e Bs será presumivelmente da esquerda para a direita (do mesmo modo como ele processa a cadeia de palavras) e com isto deve estar equipado para levar em conta as restrições concernentes à sequência destes elementos. No caso do processador proposto aqui, cada restrição apenas deve ser incorporada às regras da gramática especificamente às regras que transmitem informação sobre os Bs.

Analisando o caso mais de perto pode ser visto que num PDA simples, as posições relativas de dois preenchedores dependem da ordem em que eles entram na pilha, a qual por seu turno depende da ordem linear que eles estão na sentença (supondo um processador estritamente da direita para a esquerda). Por outro lado, a ordem em que os preenchedores são removidos da pilha depende das posições destes preenchedores na pilha, e assim da ordem linear que eles aparecem na sentença. Devido a isso, é possível assegurar que se dois preenchedores já estão na pilha quando um buraco é encontrado, o segundo preenchedor será associado com o buraco como requerido pela restrição de dependência encaixada.

Por outro lado, se um modelo ATN procurar incorporar a restrição de dependência encaixada, deve ser recordado que a memória HOLD foi suposta antes como comportando níveis para dar conta das restrições de comando e das restrições das ilhas. Não é óbvio então como a restrição de dependência cruzada pode ser incorporada a esta estrutura da memória HOLD (que é semelhante a um marcador frasal). O problema é que as posições dos preenchedores podem ser determinadas na memória HOLD com níveis através de suas alturas no marcador frasal, e não através de sua ordem linear. Para o caso de duas dependências preenchedor-buraco, não há problemas porque a condição de comando implica que a altura e a ordem linear sejam perfeitamente correlacionadas se os dois preenchedores não forem irmãos (o que é correto): este subcaso da restrição de dependência encaixada é então equivalente à condição de que o primeiro buraco seja associado com o mais baixo dos dois preenchedores que estão na pilha. Isto pode ser imposto como um tipo de uma condição de distância mínima de acesso em HOLD: uma busca para cima de buraco em HOLD para conseguir um preenchedor é tal que aceita o primeiro preenchedor encontrado. Assim o mecanismo de "barreira" descrito acima deve ser estendido, tal que se HOLD já contém um preenchedor quando um segundo preenchedor (baixo) é colocado nele, uma barreira deve ser estabelecida sobre o mais baixo preenchedor e deve permanecer aí até que a este preenchedor seja assinalado um buraco.

Esta solução sobre HOLD traz problemas porque a barreira que bloqueia o acesso ao mais alto preenchedor só deve ser usada se os dois preenchedores pertencerem à mesma categoria: isto pressupõe então a existência de algum mecanismo para comparar preenchedores em diferentes níveis do estoque.

No caso do processador aqui proposto, o mecanismo de ligação entre preenchedores e buracos está nas regras que utilizam as categorias derivadas. Estas regras podem ser de dois tipos:

$X \rightarrow Y \ X/Y$  (preenchedor primeiro)  
 $X \rightarrow X/Y \ Y$  (buraco primeiro).

O que deve ser feito aqui é a limitação das posições nas quais a notação para buraco para uma dependência alta seja passada por regras. Assim não é permitido escrever regras do tipo

$X/Z \rightarrow Y \ X/Z/Y$

envolvendo uma dependência para baixo do tipo preenchedor primeiro, nem regras do tipo

$X/Z \rightarrow X/Z/Y \ Y$

envolvendo uma dependência para baixo do tipo buraco primeiro.

Outra restrição, também natural, é que duas indicações de buracos da mesma categoria não podem mudar sua ordem linear enquanto eles passam através do marcador frasal. Isto pode ser expresso pela proibição de regras do tipo

$X/Y_1/Y_2 \rightarrow \dots \ Z/Y_2 \dots \ Z/Y_1 \dots$

onde "... " denota qualquer concatenação de categorias.

Estas restrições são aqui simplesmente colocadas para o escritor do processador, ficando colocadas a proibição (por razões linguísticas) de escrever regras deste tipo.

#### 4.5 Tratamento das perguntas

As sentenças interrogativas do Português podem ser classificadas em tres tipos:

- as perguntas globais (ou proposicionais) do tipo sim/não;
- as perguntas de instanciação (ou parciais) do tipo wh;
- as perguntas tag.

As interrogativas globais são geradas sintaticamente pela regra

$s(\text{interrog}, \dots) :- s(\text{decl}, \dots), \text{ponto\_de\_interrog}.$

tal que não ocorre nenhuma modificação sintática em relação à declarativa. Contudo, "O Antonio telefonou?" pode também ser dita como "Telefonou o Antonio?", de aceitação duvidosa. Se ela for tida como gramatical, devem ser acrescentadas as regras:

$s(\text{interrog}, \dots) :- s/n(\text{interrog}, \dots), n(3, \dots),$   
 $\text{ponto\_de\_interrog}.$

$s/n(\text{interrog}, \dots) :- v(3, \dots), n/n(\dots).$

As interrogativas de instanciação caracterizam-se pela presença de palavras interrogativas, as quais funcionam como quase-quantificadores. A regra sintática geral é

$s(\text{interrog}, \dots) :- a(\text{wh}, \dots), s/a(\dots), \text{ponto\_de\_interrog}.$

onde "a" pertence ao conjunto {LN, LP, LAdj, LAdv}, sendo que a interpretação semântica é

$\lambda \text{mbda } P [\lambda \text{mbda } h [ (s/a(\dots))' ] (p)] (a')$ .

Os morfemas quantificadores como "que" podem possuir um valor de especificador de locução nominal, como em "Que meia Pedro usa?"; as regras são:

$s(\text{interrog}, \dots) :- n(\text{wh}, \text{que}, \dots), s/n(3, n2, \dots),$   
 $\text{ponto\_de\_interrog}.$

$n(\text{wh}, \text{que}, \dots) :- [\text{que}], n(1, \dots).$

$s/n(3, n2, \dots) :- n(3, \dots), v/n(3, \dots).$

$v/n(3, \dots) :- v(3, \dots), n/n(\dots).$

onde aparece "n(1, ...)" devido a impossibilidade de artigos e determinantes após o "que", e o buraco de locução nominal ocorre dentro da locução verbal.

No caso de "quem" (referido só para [+humano]), ele substitui toda a locução nominal n(3, ...), e daí valem as regras:

$s(\text{interrog}, \dots) :- n(\text{wh}, \text{quem}, \dots), s/n(\dots),$   
 $\text{ponto\_de\_interrog}.$

$n(\text{wh}, \text{quem}, \dots) :- [\text{quem}].$

$s/n(\dots) :- n/n(\text{humano}, \dots), v(3, \dots).$

pegando por exemplo "Quem ama Pedro?" e

$s/n(\dots) :- n(3, \dots), v/n(\dots).$

$v/n(\dots) :- v(\dots), n/n(\dots).$

pegando "Quem Pedro ama?".

Por outro lado "qual" funciona com valor de especificador ("Quais os livros que Pedro comprou?"); assim valem as regras:

```
s(interrog, ...) :- n(wh, qual, ...), s/n(...),
                    ponto_de_interrog.
n(wh, qual, sing, ...) :- [qual], n(3, sing, ...).
n(wh, qual, pl, ...) :- [quais], n(3, pl, ...).

s/n(sing, ...) :- n/n(sing, ...), v(3, sing, ...).
s/n(pl, ...) :- n/n(pl, ...), v(3, pl, ...).
```

No que se refere a "quanto", ele funciona como especificador ("Quantas pessoas moram em Canoas?") ou como substituto de locução nominal. No caso de substituto de locução nominal, valem as regras:

```
s(interrog, ...) :- n(wh, quanto, ...), s/n(...),
                    ponto_de_interrog.
```

```
n(wh, quanto, sing, ...) :- [quanto].
s/n(sing, ...) :- n/n(sing, ...), v(3, sing, ...).
```

devendo ocorrer aqui repetição do mesmo esquema de regra para "quantos" e "quantas". Já no caso de especificador, valem as regras:

```
s(interrog, pl, ...) :- n(wh, quanto, pl, ...), s/n(pl, ...),
                        ponto_de_interrog.
n(wh, quanto, pl, masc, ...) :- [quantos],
                                n(3, pl, masc, ...).
n(wh, quanto, pl, fem, ...) :- [quantas], n(3, pl, fem, ...).
s/n(pl, ...) :- n/n(pl, ...), v(3, pl, ...).
```

Em relação a "onde" ("aonde"), valem:

```
s(interrog, ...) :- adv(wh, onde, ...), s/adv(lugar, ...).
adv(wh, onde, ...) :- [onde].
s/adv(lugar, ...) :- s(decl, ...), adv/adv(lugar, ...).
```

As interrogativas tag possuem, no Português, as formas:

```
Pedro casou, não é verdade?
Pedro casou, não é assim?
Pedro casou, não é?
Pedro casou, não?
```

de modo que a sequência que define o tag é constituído por:

```
não é verdade?
não é assim?
não é?
não?
```

Uma tag denota um pedido de confirmação do conteúdo proposicional da declaração precedente, ou uma estratégia manipulativa para fazer o respondedor responder como o perguntador pretende. Por estas funções, uma sequência tag não pode se ligar a outras interrogativas ("\*Onde Pedro vai, não é?"). Assim as regras de geração de tag são:

```
sent(tag, ...):- s(decl, ...), tag, ponto_de_interrog.  
tag:- [não, e', verdade].  
tag:- [não, é, assim].  
tag:- [não, é].  
tag:- [não].
```

e, no que diz respeito ao tratamento semântico, o tratamento é idêntico às interrogativas globais.

Em relação ao tratamento semântico, deve ser lembrado que o nome "erotetic logic" foi proposto por Prior e Prior /PRI 55/ para a lógica das perguntas. Esta proposta foi feita em analogia com a lógica das proposições e concebida como um sistema dedutivo que envolve a construção de um esquema inferencial, onde perguntas podem servir como premissas e, em alguns casos especiais, como conclusões. No caso das linguagens naturais, a tarefa real, para os autores envolve supor um perguntador e um respondedor de posse de uma linguagem em comum (linguagem natural) e implica no estabelecimento de um processo de tradução desta linguagem natural numa linguagem lógica. Esta última, do ponto de vista prático e teórico, deveria, para os autores, possuir uma maquinaria que permitisse perguntar e responder, implicando isso na utilização de um conjunto de conceitos úteis para a categorização, a avaliação e o relacionamento de perguntas e respostas.

Um traço conceitual importante nesta linguagem lógica é que o significado de uma pergunta endereçada ao sistema (no caso deste trabalho, sistema computacional) não deveria ser propriamente identificado com como o sistema processa a pergunta: a proposta é que o significado da pergunta deve ser identificado com a gama de respostas que a pergunta permite. Isto é, para o caso do problema em questão neste trabalho, para o sistema computacional e o usuário concordarem quanto ao significado da pergunta, deve haver acordo com o que conta como resposta da pergunta (levando isso em consideração como a pergunta foi formulada).

Considerando o aspecto da humanização da conversação com o sistema computacional, deve ser lembrado que numa conversação existem em geral tres tendências de reação a uma pergunta: resposta, ou rejeição da pergunta, ou indiferença à pergunta. Em certos casos, a obrigação de verdade solicitada pelo ato de fala do perguntador só fica validada após a resposta do respondedor, que envolve a aceitação da pergunta. Contudo existe uma hierarquia de preferência social para a reação do respondedor, pois após a pergunta em geral a aceitação da pergunta é o preferido na escala social, sendo que a posposição costuma ser escolhido quando a situação é conflituosa. Esta posposição em Português envolve respostas do tipo "Talvez.", "Não sei."; para os efeitos deste trabalho, uma resposta como "Não sei." deve ser dada quando o interpretador não consegue deduzir uma resposta à pergunta feita.



Um primeiro problema para o tratamento semântico de perguntas numa gramática nos moldes do PTQ é que as perguntas não são asserções, não sendo assim nem verdadeiras nem falsas. Um filósofo poderia contudo tentar parafrasear uma pergunta como uma sentença declarativa que indicasse a ignorância do falante, ou como uma declaração por este falante do seu desejo de conhecer, ou como uma meta-declaração que indicasse a declaração de qual deva ser o fluxo do discurso. Estas abordagens foram, de alguma forma, assumidas pelos linguistas e a primeira hipótese, por ser a mais fundamental, é discutida a seguir.

Esta discussão aqui é feita porque existem dois grandes tipos de sentenças interrogativas: as perguntas diretas ("Está chovendo?", "Qual o livro que Maria está lendo?") e as indiretas ("Responda se está chovendo."). As propostas tradicionais dos linguistas envolvem assimilar as perguntas diretas às perguntas indiretas, tal que cada pergunta direta é tratada como semanticamente equivalente a um certo tipo de sentença declarativa que possui uma correspondente pergunta indireta encaixada sob um verbo performativo.

Schiffer /SCH 72/ aí é mais radical e assume que a sentença "Que idade a Joana tem?" é a mera expressão de um ato de fala, mas não de uma proposição; porém Katz /KAT 77/ interpreta a pergunta acima como "A diz que B está para dizer para A a idade da Joana". Por outro lado, Sadock /SAD 74/ analisa a pergunta acima como um pedido de informação do tipo "Eu peço para voce (voce me diz a idade da Joana)", seguindo a tradição clássica entre os linguistas. Assim "Está chovendo?" é vista como expressando a mesma proposição que "Eu pergunto para voce (voce me diz) se está chovendo"; "Qual o livro que Maria está lendo?" é parafraseado como "Eu pergunto para voce (voce me diz) qual o livro que Maria está lendo". Isto reduz o problema da semântica das perguntas diretas ao problema de como as perguntas indiretas são interpretadas.

Existem dois modos clássicos de realizar esta redução: um modo é partir da sintaxe da pergunta indireta e derivar a pergunta direta por transformações que preservem o significado; o modo alternativo é gerar cada pergunta direta diretamente e construir regras de interpretação semântica que façam com que cada uma seja semanticamente equivalente à correspondente pergunta indireta.

Estas duas abordagens são aqui afastadas porque se perguntas diretas são semanticamente equivalentes a sentenças declarativas de certo tipo, então elas deveriam possuir um valor-verdade, e não é possível conciliar isso com a visão de que "Qual o livro que Maria está lendo?" não possui valor-verdade.

Uma proposta alternativa é devida a Ajdukiewicz /AJD 74/. Ela envolve a descrição de uma pergunta como consistindo de uma matriz sentencial (uma sentença com um ou mais componentes substituídos por variáveis) precedida por um operador interrogativo "para que x" (ou "para que x, y, z, ...", se a matriz possuir mais do que uma variável livre). Assim "Quem está lendo um livro?" é transformado em [Para que x, x está lendo um livro?] e simbolicamente  $(x?) [x \text{ está lendo um livro}]$ , sendo que nesta proposição o operador interrogativo possui um papel semelhante ao de um quantificador, de modo que ele liga uma

variável livre à sentença matriz. Além disso, para as variáveis ligadas pelo operador interrogativo, pode ocorrer substituição por locução que pertença a mesma categoria gramatical da variável; porém esta substituição não preserva a verdade, como é o caso da substituição de variáveis ligadas por um quantificador. No caso do operador interrogativo, há a preservação apenas da boa formação (gramaticalidade), mas não do valor-verdade. Isto ocorre porque os resultados da substituição são respostas possíveis à pergunta, mas necessariamente não são verdadeiras.

O conjunto das respostas possíveis, neste caso, depende essencialmente da lógica na qual a lógica das perguntas é fundada. Em particular, ela depende dos tipos de variáveis aceitas e do conjunto das locuções que uma variável representa (isto é: as locuções que podem substituir esta variável). Uma teoria dos tipos aqui é interessante porque ela oferece vários tipos de variáveis: a lógica que Ajdukiewicz teve em mente foi a teoria de Lesniewski das categorias semânticas, a qual é muito rica em relação a tipos. Nesta lógica, as categorias gramaticais não se superpõem (o que é padrão para qualquer tratamento lógico), e, de um modo muito mais importante, uma locução pode ser substituída por uma variável somente se esta variável é um funtor de um certo número de argumentos que pertencem a alguma categoria gramatical, e a locução é algum funtor do mesmo número de argumentos da mesma categoria. Ou ainda se ambas (a variável e a locução) são da mesma categoria primitiva como uma sentença ou um substantivo. Assim, para duas locuções, existe uma variável para cada uma delas e estas variáveis podem se substituir se e somente se, quando uma puder ser substituída, a outra puder também. Esta limitação parece contudo bastante rígida no caso das linguagens naturais, e ela deve ser diminuída.

A questão da formalização no "espírito das idéias" de Montague foi tratado pelo autor em /MOS 81/: a proposta básica do autor é que um ato de fala de pedido de informação (uma pergunta) possui um conteúdo proposicional incompleto, que pode ser preenchido pela resposta. Neste caso uma pergunta pode ser vista como uma proposição aberta, onde alguns elementos são variáveis lógicas: uma pergunta envolve a falta de um argumento na sua proposição, argumento este que aparecera na resposta. Uma situação delicada é apresentada por perguntas do tipo "sim/não": neste caso, a resposta do tipo "sim" (ou "não") serve para montar a proposição, pois sem ela a sentença da pergunta não pode ter um valor verdade.

Aceitando a idéia de conteúdo proposicional incompleto para as perguntas, Mosca /MOS 82/ discute o problema da relação semântica entre a pergunta e a resposta e coloca que, se a categoria das respostas é vista como nebulosa (fuzzy), pode ser assinalado à resposta um grau  $l$  de membro da conversação com resposta coerente, sendo  $l$  do tipo

$l(\text{pergunta}, \text{resposta})$ .

sendo que, desta forma, a resposta pode ser vista como  $w$ -admissível de primeira ordem se

$\max l(\text{pergunta}, \text{resposta}) \geq w$  (" $\geq$ " = maior ou igual)

e a resposta seria  $w$ -admissível de segunda ordem se

$$\max (\min (l(\text{pergunta-resposta } u)), \\ l(\text{pergunta, resposta } v)) \geq w,$$

tal que as admissibilidades de ordem superior à primeira descrevem as restrições internas do conjunto das categorias de resposta. Estas restrições de w-admissibilidade são de natureza local, sendo que Mosca /MOS 82/ concebe uma w-admissibilidade para toda a conversação. Uma proposta fraca seria caracterizar estas w-admissibilidades a partir de uma relação de implicação entre os tópicos dos enunciados como faz Mosca /MOS 82/; assim se p implica semanticamente q (sob o predicado R definido na elocução de p), então p e a negação de q não é admissível. A dificuldade prática contudo aqui é forte: a negação de q envolveria a construção de um conjunto de negações, as quais ou seriam todas iguais (por serem falsas) - o que não seria natural, ou existiriam tantas negações quantos enunciados falsos (o que tornaria o processo computacional extremamente demorado).

Num adendo a este trabalho (originalmente /MOS 81/), esta w-admissibilidade é vista como um problema de extensionalidade, no "espírito" das idéias do PTQ. Porém deve ser lembrado que Montague não modelizou perguntas, devido a dificuldades de estabelecer condições de verdade, contudo ele propôs numa nota de rodapé (número 3) do PTQ /MDN 73/ que as condições de verdade sejam substituídas pela utilização de "condições de preenchimento" que caracterizariam o conteúdo semântico de uma resposta correta. Uma leitura possível desta nota de rodapé é que uma pergunta possui como extensão o conjunto de suas respostas possíveis (ou verdadeiras).

Dentro dessa linha de leitura, existem algumas propostas já feitas. Assim Hintikka /HIN 77/ substitui perguntas indiretas por orações relativas, construindo perguntas encaixadas, tal que uma pergunta encaixada denota um tipo de função que toma como argumentos as intensões do verbo da pergunta encaixada; gera-se assim uma denotação de uma oração com verbo intransitivo. Esta proposta está baseada na "teoria dos jogos" e envolve a tradução de "Está chovendo?" (formulada por João) por uma sentença do tipo "João pergunta se está chovendo?" e a tradução formal e

$$(\text{está-chovendo})' = \text{lambda } w \text{ lambda } x [ \\ \text{[está-chovendo } \rightarrow w[\text{ext}(x, \text{int está-chovendo})] ] \& \\ \text{[ " está-chovendo" } \rightarrow w[\text{ext}(x, \text{int " está-chovendo" })] ]]$$

e, por outro lado, "Quem chegou?" tem como tradução final

$$\text{quem-chegou}' = \text{lambda } w \text{ lambda } x (\text{E}y)[\text{chegou}'(y) \rightarrow \\ \text{ext}(x, \text{int chegou}'(y))]$$

onde w é uma variável do tipo

$$\langle s, \langle \langle s, t \rangle, \langle \langle s, e \rangle, t \rangle \rangle \rangle$$

variando sobre possíveis intensões de verbos de perguntas encaixadas. Em relação a esta proposta, Karttunen /KAR 77/ mostra que ela não consegue atingir vários verbos que admitem perguntas encaixadas, e como tal ela é falsificada.

Hamblin /HAM 73/ por seu turno, propõe que "who" e "what" denotam um conjunto de indivíduos, tal que a pergunta coloca pragmaticamente uma escolha entre um conjunto de proposições: este conjunto seria expresso pelas possíveis respostas. A idéia, ao nível formal envolve a tradução de "Está chovendo?" por

$\lambda p [p = \text{int } \text{'está-chovendo'} \vee p = \text{int } \text{'est\u00e1-chovendo'}]$   
e de "Quem vem?" por

$\lambda p (\exists x) [p = \text{int } \text{'vem'}(x)]$ .

Uma reformulação possível aqui é fazer com que uma pergunta denote o conjunto de proposições expressas por suas respostas verdadeiras, e não como coloca Hamblin /HAM 73/ através de suas respostas possíveis. Um argumento aqui para sustentar esta proposta é lembrar que verbos como "dizer" com complemento do tipo "que" ("João diz para Maria que Pedro passou no exame") não acarretam que o que é dito é verdadeiro, como uma pergunta indireta leva a supor. Isto torna então mais aceitável a colocação de que perguntas devem denotar um conjunto de proposições verdadeiras.

Por outro lado, Karttunen /KAR 77/ interpreta perguntas diretas como perguntas indiretas, adicionando uma nova classe (Q) (definida como  $t/t$ ) para perguntas indiretas na Gramática de Montague. Ele propõe derivar cada pergunta indireta a partir de uma sentença declarativa, a partir de uma regra de pergunta prototípica: uma pergunta do tipo wh é derivada por quantificação da "locução nominal interrogativa" da proto-pergunta que possui um pronome subscrito como variável livre. Karttunen /KAR 77/ refere que as perguntas desta categoria  $t/t$  poderiam ser traduzidas em expressões de lógica intensional que denotassem conjuntos de proposições, isto é  $\langle \langle s, t \rangle, t \rangle$ . Esta definição sintática da categoria Q não significa que exista alguma regra sintática que combine perguntas com sentenças para fazer sentenças; em vez disso, as perguntas indiretas entram em construções maiores pela combinação com verbos encaixadores de perguntas (como "conhecer", "recordar", "dizer"). A locução resultante é então uma locução verbal intransitiva. E assim a categoria dos verbos encaixadores de perguntas é definida como VI/Q. O problema, contudo, envolve as perguntas diretas do tipo sim/não e a existência de pressuposições nas perguntas, como Karttunen mesmo reconhece

Bennett /BEN 79/ propõe que a extensão de uma pergunta seja o conjunto de respostas corretas. O problema que pode ser levantado é que desta forma as respostas deixam de ser proposições. Uma saída é que as respostas sejam vistas como proposições abertas, no sentido de serem funções de seqüências de indivíduos em proposições. Assim cada pergunta (direta ou indireta) pode possuir, como extensão, um conjunto de proposições abertas; além disso, sua intensão é uma propriedade das proposições abertas: uma função de pontos de referência para conjuntos de proposições abertas.

A solução apontada por /MOS 81/ leva em consideração o conceito de pressuposição de pergunta. O conceito usado de pressuposição de pergunta é simplesmente: uma pergunta Q pressupõe uma pressuposição A sse a verdade de A é uma condição lógica necessária para haver alguma verdade em Q.

Assim uma pergunta do tipo wh, como "Qual linguista Maria ama?", "Onde está Vera?", "Com quem Tereza saiu?", possui a forma geral

$s(\text{interrog}, \dots) :- a(\text{wh}, \dots), s/a(\dots), \text{ponto\_de\_interrog.}$   
e, em geral, a resposta a esta pergunta wh envolve a forma  $a(\dots)$ .

Pegando como exemplo uma pergunta wh do tipo "Qual é a capital do Brasil?", pode ser visto que ela assume a pressuposição

(Ex)  $[(s/n(\dots))' (\text{int } x')]$   
onde x é uma variável que aponta para  $n(\dots)$ .

Uma exceção para esta proposta é a existência de perguntas wh que possuam como verbo principal um verbo semanticamente fraco como "fazer", tal que a resposta deve ser a especificação de uma ação para ser adequada. Nesta abordagem, é necessário então desdobrar  $s/n(\dots)$  de modo que para a regra

$s(\text{interrog}, \text{fazer}, \dots) :- [o], [\text{que}], n(3, \dots),$   
 $v(\text{fazer}, \dots), \text{ponto\_de\_interrog.}$   
vale a interpretação semântica

(Ex)  $[x' (\text{int } (n(3, \dots)))']$ .

Afastando o caso dos verbos semanticamente fracos, como regra geral a pergunta wh, isto é com  $a(\text{wh}, \dots)$ , exige como resposta uma locução, que segundo a regra geral colocada no início desta secção é:

$\text{lambda } p \text{ [lambda } h \text{ [ } (s/a(\dots))' \text{ ] } (p) \text{ ] } (a')$ .

Isto pode ser feito porque as locuções interrogativas ("que homem", "quem", "quantos homens") são tratadas aqui como as locuções nominais ordinárias: as perguntas-wh são derivadas por mera quantificação de uma locução nominal (no "espírito" das idéias do PTQ). Assim a categoria sintática das locuções interrogativas  $(n(\text{wh}, \dots))$  é definida como  $t//IV$ , com definição similar às T-locuções do PTQ. Por outro lado, por razões semânticas, cada locução wh seria equivalente a uma locução nominal quantificada existencialmente. No estilo do PTQ, isto envolveria incluir a seguinte regra sintática, seguida da respectiva regra de interpretação semântica:

- se c é um CN, então  $[c \text{ ' } (x) \text{ e } P \text{ int } x']$  é um wh

-  $\text{lambda } P \text{ (Ex) } [c' (x) \text{ e } P \text{ int } x']$

sendo ser observado que esta tradução é idêntica a tradução de "um homem".

Esta opção é contudo problemática porque Hintikka /HIN 77/ argumenta que toda pergunta wh é ambígua entre uma leitura universal e uma leitura existencial. Para o quantificador interrogativo, o sentido existencial aparece por exemplo em "Alguém veio e João quer saber quem é". Por outro lado, deve ser lembrado que Hintikka não leva em conta que as perguntas-wh diretas muitas vezes não esperam uma resposta exaustiva; assim "Quem, por favor, veio na festa ontem?" pode ser satisfeita por uma lista de algumas pessoas (mas não todas) que vieram na festa. O problema talvez seja que a dicotomia existencial/universal não se aplique aqui de um modo totalmente adequado.

Meramente por opção, no que se segue, cada pergunta-wh é tratada universalmente, embora cada palavra-wh seja interpretada existencialmente. O tratamento universal da pergunta está implícito no uso da expressão [lambda p]. Porém deve ficar claro que a análise proposta aqui é tentativa, não devido ao modelo descritivo utilizado, mas sim devido a ignorância atual, no que se refere ao Português, no que diz respeito às restrições sintáticas sobre o escopo dos quantificadores.

A proposta aqui, nos moldes do PTQ, envolve adicionar a categoria Q, como faz Karttunen /KAR 77/. Porém ele define Q como t//t, e aqui é usada uma fórmula geral (não bem no estilo do PTQ) do tipo t//X, tal que

X = {sim, não}, nas perguntas sim/não;

X = a, no caso das perguntas wh;

X = t, nas perguntas com verbos semanticamente fracos;

X = t, nas perguntas alternativas.

resolvendo o problema do tratamento das perguntas sim/não na proposta de Karttunen /KAR 77/.

Uma primeira formulação para as perguntas alternativas é:

Sxx. Se p pertence à categoria t, então [p ou não p ?] é uma sentença da categoria Q, isto é

s(interrog, alter, ...):- s(decl, ...), [ou], s(decl, ...),  
ponto\_de\_interrog.

Txx. lambda p [ext p & p = int p].

e assim "Pedro dorme" é uma sentença (categoria t) que se traduz em [dorme'(p)]; logo "Pedro dorme ou Pedro não dorme?" é uma sentença da categoria Q, sendo traduzida como

lambda p [ext p & p = dorme'(p)]

onde a fórmula de tradução acima é uma expressão em lógica intensional que denota uma função de proposições em valores verdade (ou seja: um conjunto de proposições). Se Pedro dorme, então a denotação de "Pedro dorme ou Pedro não dorme?" é o conjunto unitário cujo único membro é a proposição que Pedro dorme. No caso em que Pedro não durma, ela denota o conjunto unitário "Pedro não dorme".

No caso de uma pergunta alternativa do tipo

p1 ou p2 ou ... ou pn

vale a regra

Sxx. Se [p1 ?] é um Q, e se [p2 ?] é um Q, ..., e se [pn ?] é um Q, então [p1 ou p2 ou ... pn ?] é um Q.

obs.: o tratamento da disjunção aqui segue as regras mostradas para a coordenação, em secção anterior.

Txx. lambda x [(p1?)'(x) v (p2?)'(x) v ... (pn?)'(x)].

Assim "Maria cozinha ou João dorme?" é traduzida como

lambda x[(maria-cozinha?)'(x) v (joão-dorme?)'(x)]

a qual é equivalente a fórmula não abreviada seguinte

lambda x [lambda x [ext x & x = int cozinha'(m)](x) v

lambda x [ext x & x = int dorme'(j)](x) ]

a qual é equivalente a

lambda x[ext x & [x = int cozinha'(m) v x = int dorme'(j)]]

sendo que esta tradução aceita um conjunto de proposições que pode conter zero, uma ou duas proposições.

Um tratamento para as perguntas sim/não é simplesmente:

Sxx. Se p é um t, então [p ?] é um Q, isto é:

s(interr, global, ...):- s(decl, ...),  
ponto\_de\_interrog.

Txx.  $\lambda x [p'(x) \vee \sim (E_y) p'(y) \ \& \ p = \text{int } \sim (E_y) p'(y)]$

sendo que a fórmula de tradução é equivalente a

$\lambda x [\text{ext } x \ \& \ [x = \text{int } p' \ \vee \ x = \text{int } \sim p']]$

a qual designa o conjunto unitário contendo ou a proposição p ou a proposição  $\sim p$ . Deve ser observado que este tratamento serve para as perguntas tag.

Para o caso das perguntas indiretas encaixadas como complemento de verbo ("Eu quero saber quem matou Pedro"), o tratamento é bastante simples:

Sxx. Se a é um VI/Q e b é um Q, então ab é um VI.

Txx. a'(int b').

Assim se "quer saber" é um VI/Q e "se João caminha" é um Q, então "quer saber se João caminha" é uma locução verbal intransitiva (VI) e se traduz como

quer'(int saber' (lambda p[ext p & [p = int caminha'(j) v  
p =  $\sim$  caminha'(j)]]) )

e logo "Antonio quer saber se João caminha" se traduz em

quer'(int a', int saber' (lambda p[ext p &  
[p = int caminha'(j) v p =  $\sim$  caminha'(j)]]) )

numa formulação simplificada.

Como comentário final, deve ser discutido aqui o tratamento dado a perguntas do tipo "Quem é João?", na qual o nome "João" é usado atributivamente (o falante não possui uma pessoa particular em mente). O problema está na distinção entre o uso atributivo e o uso referencial de uma descrição (conceitos discutidos em /KRI 77/). A resposta exigida normalmente para a pergunta acima não envolve o uso referencial de uma descrição, mas sim o uso atributivo: assim sendo esta resposta "normal" não é uma resposta total, mas só uma resposta parcial, pois uma resposta total (ou completa) envolve uma referência. Deve ser verificado que esta pergunta é tipicamente respondida com descrições do tipo "O homem que está na terceira cadeira". O problema é que a teoria aqui mostrada pode aceitar respostas do tipo "A pessoa que é João é João". Esta aparente consequência absurda da teoria tem explicação: a resposta do tipo "O homem que está na terceira cadeira" usa uma descrição definida referencialmente para referir-se a João, colocando então a face de João na mente do perguntador. Assim, embora parecendo uma resposta total e completa ela é somente uma resposta parcial.

## 5 TRATAMENTO LÓGICO DA LOCUÇÃO NOMINAL, DA LOCUÇÃO ADJETIVAL, DA LOCUÇÃO PREPOSICIONAL E DA LOCUÇÃO ADVERBIAL

### 5.1 Tratamento da locução nominal

A proposta de análise para as locuções nominais defendida aqui contém, no seu núcleo, as idéias expressas no PTQ de Montague. Ela utiliza e expande o que foi apresentado em /MOS 84f/ /MOS 85/.

A estrutura interna da locução nominal inclui obrigatoriamente um núcleo e opcionalmente outros dois tipos de constituintes: os especificadores e os complementos.

O núcleo da locução nominal é o elemento fundamental e determina a concordância dos especificadores e de alguns complementos (locução adjetival e epíteto). Podem funcionar como núcleo de locução nominal os substantivos (próprios ou comuns) e os pronomes. São exemplos dos primeiros "João fugiu" e "A mesa está quebrada". No caso de núcleo composto por pronome, podem ocorrer os pronomes pessoais, os quantificadores e os demonstrativos.

Os pronomes pessoais ("Ela saiu") ocupam toda a locução nominal, não admitindo especificadores nem complementos (exceto os epítetos: "Ele, o único barbeiro da cidade, havia morrido"); eles possuem como referência as pessoas do discurso ou aquelas de quem se fala e, além disso, são identificadores como os nomes próprios. Daí que os nomes próprios e os pronomes pessoais não precisam ser antecidos de artigos, visto que contêm em si sua própria determinação. Mesmo assim um nome próprio pode ser antecido por um artigo definido ("O João fugiu"), mas a função deste artigo não é neste caso a de operador de individualização, podendo ser suprimido na interpretação semântica.

Já os pronomes quantificadores ("alguém", "ninguém", "tudo") também não admitem especificadores nem complementos, exceto os epítetos e as orações relativas ("Ninguém chegou", "Traz tudo o que tiveres", "Alguém que tu conheces chegou"). O mesmo vale para os pronomes demonstrativos "aquilo" e "isso": "Aquilo é um livro", "Isso foi feito por Pedro". Os pronomes quantificadores e os demonstrativos constituintes do núcleo da locução nominal não se referem às pessoas do discurso, e possuem o traço [neutro]; os quantificadores possuem o traço [+humano] no caso de "alguém" e "ninguém" e o traço [-humano] no caso de "tudo".

Um outro tipo de núcleo de locução nominal são os nomes coordenados ("O papel e a caneta servem para escrever"), levando o verbo sempre para o plural. Isso é tratado a parte.

As regras sintáticas até aqui para os nomes próprios são:

```
n(3, nome_próprio, ...) :- nome_próprio (...).
n(3, nome_próprio, art_def, ...) :- art_def (...),
nome_próprio(...).
n(3, nome_próprio, epíteto( ... ) :- nome_próprio (...),
epíteto(...).
epíteto(...) :- vírgula, n(3, ...), vírgula.
```



sendo que as correspondentes interpretações semânticas são:

```
int (nome_próprio(...))'  
int (nome_próprio(...))'  
int (nome_próprio(epíteto, ...))'
```

onde na última interpretação o epíteto é tratado como uma oração relativa.

Para o caso dos pronomes quantificadores e demonstrativos núcleos de locução nominal, valem as regras sintáticas:

```
n(3, pron_pessoal, ...) :- pron_pessoal(...).  
n(3, pron_pessoal, epíteto, ...) :- pron_pessoal(...),  
epíteto.  
n(3, pron_demonstr, neutro, ...) :-  
pron_demonstr(neutro, ...).  
n(3, pron_demonstr, neutro, epíteto, ...) :-  
pron_demonstr(neutro, ...), epíteto(...).
```

sendo que as interpretações semânticas respectivas são:

```
int (pron_pessoal(...))'  
int (pron_pessoal(epíteto, ...))'  
int (pron_demonstr(neutro, ...))'  
int (pron_demonstr(neutro, epíteto, ...))'
```

Em relação aos especificadores de locução nominal, eles se encontram à esquerda do núcleo. Eles envolvem determinantes e quantificadores.

O termo determinante cobre uma classe não muito bem definida de elementos que precedem o substantivo, tendo como base o fato de que eles determinam o nome do ponto de vista semântico; em geral são considerados determinante os artigos, os demonstrativos, os possessivos e os indefinidos. Porém as propriedades semânticas desses elementos variam muito, o mesmo sucedendo com as propriedades sintáticas. Então, para os limites deste trabalho, o termo determinante denota somente artigos e deíticos (demonstrativos e possessivos), os quais se distinguem dos quantificadores em relação ao comportamento sintático e à interpretação semântica.

Os artigos definidos ("o gato") e os indefinidos ("uns gatos") não co-ocorrem com os demonstrativos ("o este professor", "estes o professor"), mas podem anteceder os possessivos ("os teus gatos"). Por outro lado, os deíticos podem conjugarem-se entre si na estrutura [DEMONST + POSSES], como em: "todos estes teus defeitos", "aqueles nossos livros". Os possessivos podem seguir o substantivo se este não é precedido de artigo ou ocorre com artigo indefinido ("noticias tuas", "alguns vizinhos teus"), mas isto não é obrigatório. As regras sintáticas são então:

```
n(3, ...) :- art_def(...), n(2, ...).  
n(3, ...) :- art_indef(...), n(2, ...).  
n(3, ...) :- demonstr(...), n(2, ...).  
n(2, ...) :- posses(...), n(1, ...).
```

tal que a interpretação semântica de cada regra

```
a(...) :- b(...), c(...).
```

é do tipo

```
(b(...))' (int (c(...)))').
```

Em relação à quantificação, deve ser observado que em Português a quantificação de substantivos e adjetivos é feita obrigatoriamente pelo número (singular, plural). Além desse tipo de quantificação, os substantivos, os adjetivos e os advérbios podem ser quantificados por um conjunto de especificadores denominado de "locução quantificada" /BRE 73/.

Incluem-se na classe dos quantificadores os indefinidos (com exceção de "próprio" e "mesmo") e os numerais. Destes, "outro(s)", "pouco(s)" e os numerais (exceto "um") podem ser precedidos de artigo definido ("o(s) outro(s) livro(s)", "o(s) pouco(s) caso(s)") ou deíticos da forma [DEMONST + POSSES]: ("estes outros livros", "aqueles teus poucos livros"). Por outro lado, "outro(s)", "pouco(s)", "quantos", "tanto(s)", "qualquer(qualsquer)", "certo(s)", numerais (exceto "um") são antecidos por artigos indefinidos ("um outro livro", "umas poucas casas", "uns quantos porcos", "umas tantas pessoas", "umas quaisquer pessoas", "uma certa pessoa", "umas duas pessoas"). Além disso, os quantificadores "todo(s)" e "ambos" são obrigatoriamente seguidos de artigos. As regras sintáticas então são:

```
n(3, quant, art_def, ...) :- art_def(...),
    n(2, quant, art_def, ...).
n(2, quant, art_def, ...) :- posses(...),
    n(1, quant, art_def, ...).
n(1, quant, art_def, ...) :- quant(art_def, ...),
    n(0, ...), ...

n(3, quant, art_indef, ...) :- art_indef(...),
    n(2, quant, art_indef, ...).
n(2, quant, art_indef, ...) :- posses(...),
    n(1, quant, art_indef, ...).
n(1, quant, art_indef, ...) :- quant(art_indef, ...),
    n(0, ...), ...
```

onde os pontinhos são os complementos de  $n(0, \dots)$ , e as interpretações semânticas para as regras do tipo  $[a \rightarrow b\ c]$  são do tipo  $[b' \text{ (int } c')]$ .

Em relação ao quantificador "todo(s)" . ele antecede os artigos definidos ("todos os livros", \*"todos uns livros"), os demonstrativos ("todos estes livros"), e as construções [ARTDEF + POSSES] e [DEMONST + POSSES] como em "todos os meus livros", "todos estes meus livros". Suas regras sintáticas são então:

```
n(3, todo, ...) :- todo(...), n(3, art_def, masc, sing, ...).
n(3, todo, ...) :- todo(...),
    n(3, demonstr, masc, sing, ...).

n(3, todos, ...) :- todos(...), n(3, art_def, masc, pl, ...).
n(3, todos, ...) :- todos(...),
    n(3, demonstr, masc, pl, ...).
```

o mesmo devendo ser repetido para "toda" e "todas", e a interpretação das regras do tipo  $[a \rightarrow b\ c]$  tem como padrão a expressão  $[a' \text{ (int } c')]$ .

No que diz respeito aos numerais, eles podem ser cardinais ("um", "dois"), ordinais ("primeiro", "segundo"), multiplicativos ("dobro") e fracionários ("metade"). Os ordinais (com exceção de "um") e os cardinais ocorrem após artigo ou demonstrativo ("os dois", "estes primeiros") e após as construções [ART + POSSES] e [DEMONST + POSSES], como em "os meus dois livros", "estes meus primeiros livros". Já os multiplicativos e os fracionários fazem parte das construções partitivas do tipo [ART + \_\_\_ + de + N3] ou [DEMONST + \_\_\_ + de + N3], como em "o dobro da raiz", "a metade da raiz". Outra construção com os numerais é a iterativa, como em "os primeiros tres capítulos", "os tres primeiros capítulos". As regras sintáticas são então:

```
quant(numeral, ...) :- ordinal(...).
quant(numeral, ...) :- cardinal(...).
quant(numeral, pl, ...) :- ordinal(pl, ...),
cardinal(pl, ...).
quant(numeral, pl, ...) :- cardinal(pl, ...),
ordinal(pl, ...).
```

e as interpretações semânticas para as regras do tipo [a -> b] são do tipo (int b') e para as regras do tipo [a -> b c] são do tipo (b' (int c')).

Uma forma mais complexa de locução quantificada é a forma partitiva ("algumas dessas pessoas", "duas das mulheres", "várias de suas objeções"), que existe para todos os quantificadores apresentados com exceção de "todo(s)" e "ambos" ("todas essas maçãs", "ambos os livros"). A exceção de "todo(s)" e "ambos" está ligado ao fato desses quantificadores especificarem a totalidade do conjunto referido, opondo-se a própria noção de estrutura partitiva. Esta estrutura partitiva nunca é precedida de artigo, porém o substantivo que segue à estrutura [QUANT + de] é sempre precedido de artigo definido ou indefinido ou de pronome demonstrativo ("algumas dessas maçãs", "duas das maçãs"). Por outro lado os quantificadores "pouco" e "tanto" inserem-se na estrutura [ART INDEF + QUANT + de]; quando seguidos de pronomes pessoais ou de deíticos, outros indefinidos podem inserir-se também nesta estrutura ("uns quantos desses bolos"). Em relação a "cada", sua estrutura partitiva é [cada + de], onde o "um" é obrigatório ("cada um dos alunos"). As regras sintáticas são:

```
n(3, partit, ...) :-
quant("todo(s)", "ambos", "pouco", "tanto", ...),
prep(de, ...), n(3, ...).
n(3, partit, pouco, ...) :- art_indef(masc, sing, ...),
quant(pouco, ...), prep(de, ...), n(3, ...).
n(3, partit, tanto, ...) :- art_indef(masc, sing, ...),
quant(tanto, ...), prep(de, ...), n(3, ...).
```

onde n(3, ...) pode ser substituído pelo pronome pessoal "eles" ("tres deles", "muitos deles", "alguns deles"); as respectivas interpretações semânticas são:

```
(quant("todo(s)", "ambos", "pouco", "tanto", ...))'
(int (n(2, pl, quant, ...)))'
(quant(pouco, ...))' (int (n(3, ...)))'
(quant(tanto, ...))' (int (n(3, ...)))'.
```

Uma proposta alternativa aqui seria colocar as regras sintáticas de  $n(3, \text{part}, \dots)$  na forma

$n(3, \text{part}, \dots) :- \dots, p(3, \text{de}, \dots)$ .

onde  $p(3, \text{de}, \dots)$  é analisada como  $[\text{de } n(3, \dots)]$ . Mas isto faria com que o núcleo de  $n(3, \text{part}, \dots)$  não seja  $n(2, \dots)$ , mas sim  $p(3, \text{de}, \dots)$ .

Jackendoff /JAC 69/ e Bresnan /BRE 73/ argumentam, por seu turno, que a forma partitiva teria como estrutura profunda  $[\langle n3 \rangle \text{ q3 } (\text{de}) \text{ n3}]$ , a qual geraria também todas as locuções nominais não partitivas. Assim as construções do tipo "um livro" deveriam ser geradas da seguinte forma:

```

um (de) det3 n3
  |
  | transformação
  |
um n3

```

ocorrendo aí uma transformação de apagamento. A crítica maior a ser feita é que isto obriga a criação de um mecanismo ad hoc para geração de sentenças extremamente simples.

Por outro lado, certos substantivos usados para quantificar (significando pesos ou medidas) podem ocupar o lugar dos quantificadores na estrutura partitiva. A seleção desses substantivos é feita em função das propriedades lexicais dos substantivos quantificados. Assim expressões como "um litro de leite", "cinco metros de fazenda" envolvem um numeral não precedido de artigo e um substantivo que significa medida ou peso; a quantificação aqui é determinada. Por outro lado, expressões como "um bocado de água", "a maior parte do livro" envolvem um artigo definido ou indefinido antes do quantificador; a quantificação aqui é vaga, referindo uma parte globalmente considerada, isto implicando que o quantificador fique no singular. As regras sintáticas são então:

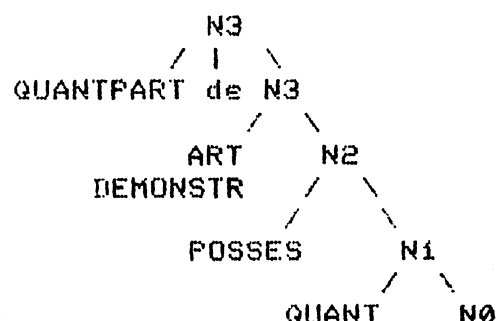
$n(3, \text{sing}, \dots) :- \text{numeral}(\text{peso}, \dots), \text{prep}(\text{de}, \dots), n(2, \text{peso}, \dots)$ .

$n(3, \text{sing}, \dots) :- \text{art\_def}(\dots), \text{quant}(\dots), \text{prep}(\text{de}, \dots), n(2, \dots)$ .

e as interpretações semânticas respectivas são:

$\text{int}(n(2, \text{peso}(\text{numeral}), \dots))'$   
 $(\text{quant}(\dots))' (\text{int}(n(2, \dots)))''$

O modelo geral defendido aqui assume então a seguinte estrutura geral para os especificadores de locução nominal:



sendo concebido que os nodos  $\text{quant}$ ,  $\text{art}$ ,  $\text{demonstr}$  e  $\text{posses}$  são opcionais, por serem especificadores.

Em relação a n1, deve ser lembrado que as construções com adjetivos podem anteceder ou serem colocadas após o substantivo, como em "casa bonita" e "bonita casa". Levando em consideração as locuções preposicionais dentro das locuções nominais ("o casamento do Mário"), pode ser proposto que elas ocorrem após o segundo adj3 ("a casa bonita do Pedro"), tal que valha a estrutura abaixo:

N1

/ / | \ \

QUANT Adj3 N0 Adj3 P3

## 5.2 tratamento da locução adjetival

A locução adjetival é constituída obrigatoriamente por um adjetivo (que funciona como núcleo) e opcionalmente por especificadores e complementos.

Os especificadores de uma locução adjetival quantificam o núcleo, constituindo o que é tradicionalmente conhecido como os graus do adjetivo. São exemplos as construções "muito alto" (tradicionalmente chamado de grau superlativo absoluto), "um pouco triste", "nada simpático", "meio triste". Outro tipo de quantificação utiliza a anteposição dos advérbios de intensidade ("imensamente", "bestialmente", "bastante", "imensamente magra"), os quais como regra geral não coocorrem com os quantificadores citados logo acima. Outra forma de fazer a quantificação é a repetição do adjetivo ("homem velho, velho, velho"). Quanto ao superlativo relativo ("a mais bonita das mulheres"), ele é um quantificador do adjetivo, possuindo a forma [ART + QUANT + ADJ + de], sendo tratado pelas regras das construções partitivas. A regra sintática é então simplesmente:

adj(3, ...) :- adv(quant, ...), adj(2, ...).

e a interpretação semântica é:

(adv(quant, ...))' (int (adj(2, ...)))'.

Os complementos só aparecem nos adjetivos com função predicativa; eles podem ser locuções preposicionais e sentenças, como em "cheios de neve", "magoados com o tio", "complicado para tu entenderes", "mais alta que a Maria", "tão alta como a Maria". As regras sintáticas são então:

adj(2, ...) :- adj(1, ...), p(3, ...)

adj(2, ...) :- adj(1, ...), sí(...).

e as interpretações semânticas respectivas são

(adj(1, ...))' (int (p(3, ...)))'

(adj(1, ...))' (int (p(3, ...)))'.

Por outro lado vale

adj(1, ...) :- adj(0, ...).

e a interpretação semântica respectiva é

int (adj(0, ...))'.

### 5.3 Tratamento da locução preposicional

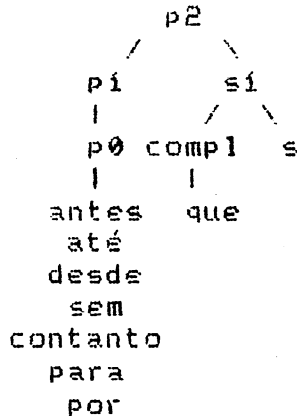
A categoria nuclear da locução preposicional é a preposição, simples ou composta (locução prepositiva); os especificadores inexistem, mas os complementos podem ser locução nominal ("de Maria") ou oração relativa ("de que tu falaste", "para viver com o Pedro"). As regras sintáticas são:

$$p(3, \dots) :- p(2, \dots), n(3, \dots).$$
$$p(3, \dots) :- p(2, \dots), sí(\dots).$$

e as interpretações semânticas respectivas são

$$(p(2, \dots))' (int (n(3, \dots)))'$$
$$(p(2, \dots))' (int (sí(\dots)))'.$$

No caso da segunda regra, ela vale para expressões do tipo



Considerando expressões como "por sobre a mesa", podem ser propostas as regras sintáticas

$$p(2, \dots) :- p(1, \dots) p(2, \dots).$$
$$p(2, \dots) :- p(1, \dots).$$
$$p(1, \dots) :- p(0, \dots).$$

com as respectivas interpretações semânticas

$$(p(1, \dots))' (int (p(2, \dots)))'$$
$$(p(1, \dots))'$$
$$(p(0, \dots))'$$

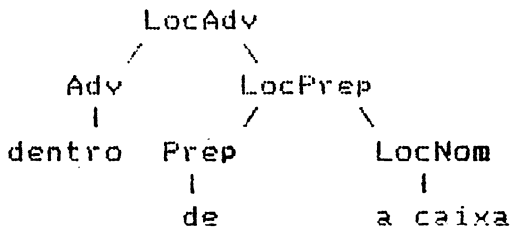
porém existem dois problemas: estas construções não são habituais no Português não literário e a ordem das preposições é extremamente restrita (\*"sobre por a mesa"). Para os efeitos de ordem prática, elas podem então ser retiradas da gramática.

Um comentário deve ser feito sobre a análise tradicional de expressões como "a cavalo" em sentenças como "Pedro chegou a cavalo" como locuções adverbiais /CUN 79/. Esta regra de nomenclatura procura obedecer ao princípio de que o nome da locução deve ser dado em termos de sua função na construção maior, e não (como na sintaxe X-barra) através da rotulação categorial. Além disso Cunha /CUN 79/ classifica "a cavalo" em expressões como "um homem a cavalo" como uma locução adjetiva; por outro lado, Lemle /LEM 84/ classifica a expressão "todos os dias" nas sentenças "Todos os dias chove", "Essa chuva todos os dias estraga as férias", "Todos os dias Pedro escova os dentes" respectivamente como locução nominal, locução adjetival e locução

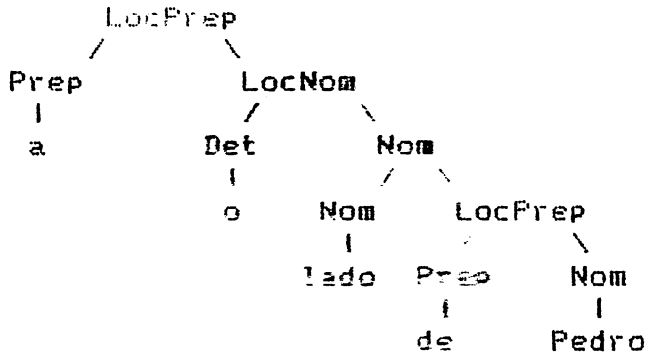
adverbial; porém a rotulação categorial fornece como resultado apenas locução nominal.

A proposta desenvolvida aqui segue a sintaxe X-barra. Isto implica que todas as preposições simples são núcleos de locuções preposicionais de algum nível (p3, p2, p1, p0); deve ser lembrado todavia que elas relacionam em geral um antecedente com um conseqüente. São exemplos de preposições simples: a, ante, após, até, com, contra, de, desde, em, entre, para, por, sem.

Alguma diferença ocorre entretanto para as locuções prepositivas da gramática tradicional, as quais são: abaixo de, acerca de, acima de, adiante de, a fim de, antes de, ao lado de, ao redor de, apesar de, atrás de, através de, de acordo com, debaixo de, de frente de, dentro de, depois de, diante de, em baixo de, em cima de, em frente de, em lugar de, em redor de, em vez de, junto a, junto de, para baixo de, para cima de, para com, perto de, por baixo de, por causa de, por cima de, por detrás de, por entre, por trás de. A proposta aqui desenvolvida assume que estas locuções prepositivas são classificadas segundo o tipo de seu núcleo: assim "dentro de", "abaixo de", "depois de" são locuções adverbiais



e, por outro lado, as locuções prepositivas "ao lado de", "em lugar de", "ao redor de", "em lugar de", "de acordo com", "em cima de", "com respeito a" são locuções preposicionais



e, nos casos em que ocorre contração entre uma preposição (de, a) e uma advérbio (baixo, cima, frente), existe a construção de uma locução adverbial (abaixo de, acima de, defrente de).

Em relação ao significado das preposições simples, elas relacionam um antecedente com um conseqüente, estabelecendo uma relação de movimento ("Pedro vai para Paris") ou uma situação ("Pedro chorava de dor"). Tanto a situação como o movimento podem ser considerados em referência ao espaço, ao tempo e à noção: "Pedro saiu de casa", "Paulo trabalha de 8 às 8 horas", "João tem um livro de Pedro". Nisto, a preposição "a" estabelece uma direção para um certo limite do movimento ("Pedro foi para

Paris"); "antes" refere anterioridade e "após" refere posterioridade; "até" indica aproximação a um limite, "com" refere associação, "contra" indica direção contrária, "de" refere afastamento de um ponto. Algumas destas preposições tem sua tradução semântica na discussão sobre verbos.

Outro caso é a seleção obrigatória de uma preposição por um verbo (por exemplo, "concordar com"), o que é discutido dentro da seção sobre locuções verbais.

#### 5.4. TRATAMENTO DA LOCUÇÃO ADVERBIAL

O advérbio é a categoria nuclear da locução adverbial, devendo ser observado que em "Pedro viu uma aranha muito grande que fugiu rapidamente", "muito" e "rapidamente" integram-se respectivamente na locução adjetival complemento da locução nominal e na locução nominal. Isto contudo foi discutido em outro local.

Os advérbios classificam-se em advérbios sentenciais (certamente, realmente, provavelmente, talvez), de quantificação ou de grau (assaz, bastante, mais, menos, pouco, tão), de lugar (aí, além, ali, aqui, dentro, fora, junto, lá, longe, onde, perto), de modo (assim, depressa, devagar, melhor, pior), de negação (não, nunca, jamais), de tempo (agora, amanhã, antontem, antes, breve, hoje, já, logo, nunca, ontem, sempre, tarde).

No caso dos advérbios sentenciais, como em "Francamente, não sei a resposta" e "Necessariamente, Pedro sabe a resposta", os advérbios "francamente" e "necessariamente" são dominados pela sentença e daí sua mobilidade ("Pedro, necessariamente, sabe a resposta", "Pedro sabe necessariamente a resposta", "Pedro sabe a resposta necessariamente").

Os especificadores no interior da locução adverbial são advérbios como função de quantificadores ("muito rapidamente"), porém os advérbios de lugar e de tempo não são quantificáveis. Os advérbios de grau que modificam um outro advérbio, colocam-se antes destes ("muito grande").

Em relação aos advérbios que modificam um verbo, se forem de modo, tempo ou lugar, eles podem colocar-se antes ou depois do verbo: "Pedro chorava tristemente", "Pedro tristemente chorava", "Tristemente Pedro chorava"; "Pedro acordou cedo", "Pedro cedo acordou", "Cedo Pedro acordou"; "Pedro chegou aqui", "Pedro aqui chegou", "Aqui Pedro chegou".

Por outro lado, os advérbios de negação que modificam um verbo, antecedem este verbo ("Pedro não come massa"); mas os advérbios tipo "nunca" e "jamais" podem ter variação de sua posição na sentença ("Jamais/nunca Pedro sai", "Pedro jamais/nunca sai", \*"Pedro sai nunca/jamais"), mas há uma preferência para a posição anterior ao verbo. Em relação à interpretação desta negação, o comum é ser aceito uma negação do predicado (negação interna) e não uma negação de toda a sentença.



As regras que envolvem sentença são então:

s(...) :- adv(3, -negação, -grau, ...), n(3, ...), v(3, ...).  
s(...) :- n(3, ...), adv(3, -negação, -grau, ...), v(3, ...).  
s(...) :- n(3, ...), v(3, ...), adv(3, -negação, -grau, ...).  
  
s(...) :- adv(3, -negação, -grau, ...), v(3, intr, ...).  
s(...) :- v(3, intr, ...), adv(3, -negação, -grau, ...).

onde -x denota uma simplificação para efeitos de apresentação, pois a regra que possui este traço apenas é um esquema para representar várias outras regras onde o traço x não deve aparecer. As interpretações semânticas para o primeiro grupo de regras é

(adv(sent, ...))' ( (v(3, ...))' (int (n(3, ...))' )  
(v(3, ...))' (adv(3, -sent, ...))' ) (int (n(3, ...))' )

e em relação ao segundo grupo de regras é

(adv(...))' ( (v(3, ...))' )

As outras regras mais específicas para locução adverbial são:

adv(3, ...) :- adv(1, negação, ...), adv(2, ...).  
adv(3, ...) :- adv(2, ...).  
adv(2, -tempo, ...) :- adv(1, grau, ...),  
adv(1, -tempo, ...).  
adv(2, ...) :- adv(1, ...).  
adv(1, ...) :- adv(0, ...).

sendo que as respectivas interpretações semânticas são

(adv(1, negação, ...))' (adv(2, ...))'

- onde 'não' vale [~] e 'nunca' e 'jamais' possuem interpretação que leva em conta o tempo

(adv(2, ...))'  
(adv(1, grau, ...))' (adv(1, -tempo, ...))'  
(adv(1, ...))'  
(adv(0, ...))'.

## 6 TRATAMENTO LÓGICO DA LOCUÇÃO VERBAL

### 6.1 Tipos de locução verbal

Os tipos de locução verbal são referidos em função dos tipos de complemento requeridos pelo verbo núcleo da locução verbal.

Assim os verbos "absolver", "afastar", "chamar", "cobrar", "encarregar", "herdar", "remover", "tingir" sofrem a seguinte análise sintática:

$v(i, n3\_de\_n3, \dots) :- v(0, n3\_de\_n3, \dots), n3(\dots),$   
 $p(3, de, \dots).$

e a interpretação semântica é:

$((v(0, n3\_de\_n3, \dots))' (int(p(3, de, \dots)))' )$   
 $(int(n3(\dots)))' ).$

Por outro lado os verbos "acusar", "cair", "derivar", "descender", "descer", "desconfiar", "desistir", "discordar", "duvidar", "falar", "escapar", "gostar", "impedir", "morrer", "suspeitar", "tratar", "voltar" sofrem a seguinte análise sintática:

$v(i, de\_n3, \dots) :- v(0, de\_n3, \dots), p(3, de, \dots).$

e a interpretação semântica é:

$(v(0, de\_n3, \dots))' (int(p(3, de, \dots)))' ).$

Os verbos "agradar", "agradecer", "ajudar", "feder", "obedecer", "perdoar", "perguntar", "pertencer", "recorrer", "renunciar" sofrem a seguinte análise sintática:

$v(i, a\_n3, \dots) :- v(0, a\_n3, \dots), p(3, a, \dots).$

e a interpretação semântica é:

$(v(0, a\_n3, \dots))' (int(p(3, a, \dots)))' ).$

Os verbos "ir", "vir" sofrem a análise sintática:

$v(i, de\_n3\_para\_n3) :- v(0, de\_n3\_para\_n3), p(3, de, \dots),$   
 $p(3, para, \dots).$

e a interpretação semântica:

$((v(0, de\_n3\_para\_n3))' (int(p(3, para, \dots)))' )$   
 $(int(p(3, de, \dots)))' ).$

Já os verbos "alimentar", "ameaçar", "comparar", "disputar", "ferrar", "juntar", "tapar", "unir" sofrem a análise sintática:

$v(i, n3\_com\_n3, \dots) :- v(0, n3\_com\_n3, \dots), n(3, \dots),$   
 $p(3, com, \dots).$

e a interpretação semântica é:

$((v(0, n3\_com\_n3, \dots))' (int(p(3, com, \dots)))' )$   
 $(int(n(3, \dots)))' ).$

Os verbos "substituir", "trocar" sofrem a análise sintática:

$v(i, n3\_por\_n3, \dots) :- v(0, n3\_por\_n3, \dots), n(3, \dots),$   
 $p(3, por, \dots).$

e a interpretação semântica é:

$((v(0, n3\_por\_n3, \dots))' (int(p(3, por, \dots)))' )$   
 $(int(n(3, \dots)))' ).$

Os verbos "ansiar", "clamar", "orar", "transitar" sofrem a análise sintática:

$v(i, por\_n3, \dots) :- v(0, por\_n3, \dots), p(3, por, \dots).$

e a interpretação semântica é:

$(v(0, n3\_por\_n3, \dots))' (int(p(3, por, \dots)))'$ .

Os verbos "apelar", "recorrer", "mentir" sofrem a análise sintática:

$v(i, para\_n3, \dots) :- v(0, para\_n3, \dots), p(3, para, \dots)$ .

e a interpretação semântica é:

$(v(0, para\_n3, \dots))' (int(p(3, para, \dots)))'$ .

Os verbos "assinalar", "contribuir", "falar", "romper", "sonhar" sofrem a análise sintática:

$v(i, com\_n3, \dots) :- v(0, com\_n3, \dots), p(3, com, \dots)$

e a interpretação semântica é:

$(v(0, com\_n3, \dots))' (int(p(3, com, \dots)))'$ .

Os verbos "cair", "crer", "intervir", "pensar", "persistir", "recair", "recidir" sofrem a análise sintática:

$v(i, em\_n3, \dots) :- v(0, em\_n3, \dots), p(3, em, \dots)$ .

e a interpretação semântica é:

$(v(0, em\_n3, \dots))' (int(p(3, em, \dots)))'$ .

Os verbos "opinar", "falar" sofrem a análise sintática:

$v(i, sobre\_n3, \dots) :- v(0, sobre\_n3, \dots), p(3, sobre, \dots)$ .

e a interpretação semântica é:

$(v(0, sobre\_n3, \dots))' (int(p(3, sobre, \dots)))'$ .

Os verbos "inculcar", "transformar" sofrem a análise sintática:

$v(i, n3\_em\_n3, \dots) :- v(0, n3\_em\_n3, \dots), n(3, \dots), p(3, em, \dots)$ .

e a interpretação semântica é:

$((v(0, n3\_em\_n3, \dots))' (int(p(3, em, \dots)))' ) (int(n(3, \dots)))'$ .

Os verbos "confessar", "convidar", "dar", "dever", "ensinar", "solicitar" sofrem a análise sintática:

$v(i, n3\_a\_n3, \dots) :- v(0, n3\_a\_n3, \dots), n(3, \dots), p(3, a, \dots)$ .

e a interpretação semântica é:

$((v(0, n3\_a\_n3, \dots))' (int(p(3, a, \dots)))' ) (int(n(3, \dots)))'$ .

Os verbos "amar", "chamar", "aumentar", "eleger", "fechar", "namorar", "querer", "saber", "usar" sofrem a análise sintática:

$v(i, n3, \dots) :- v(0, n3, \dots), n(3, \dots)$ .

e a interpretação semântica é:

$(v(0, n3, \dots))' (int(n(3, \dots)))'$ .

Por outro lado, os verbos "chover", "trovejar" sofrem a análise sintática:

$s(0, imp\_intr, \dots) :- v(3, imp\_intr, \dots), adv(3, tempo, \dots)$ .

$s(0, imp\_intr, \dots) :- v(3, imp\_intr, \dots)$ .

$v(3, imp\_intr, \dots) :- v(2, imp\_intr, \dots)$ .

$v(2, imp\_intr, \dots) :- v(1, imp\_intr, \dots)$ .

$v(1, imp\_intr, \dots) :- v(0, imp\_intr, \dots)$ .

sendo que a primeira regra corresponde a interpretação semântica:

$(adv(3, tempo, \dots))' (v(3, imp\_intr, \dots))'$

e às outras corresponde uma interpretação semântica da forma:

$(v(i, imp\_intr, \dots))'$

onde  $i = (1, 2, 3)$ .

Os verbos "morrer", "cantar", "correr", "almoçar" sofrem a análise sintática:

$v(3, \text{intr}, \dots) :- v(2, \text{intr}, \dots).$   
 $v(2, \text{intr}, \dots) :- v(1, \text{intr}, \dots).$   
 $v(1, \text{intr}, \dots) :- v(0, \text{intr}, \dots).$

esquecendo-se aqui as formas com advérbios; a interpretação semântica destas formas é

$(v(i, \text{intr}, \dots))'$

onde  $i = \{1, 2, 3\}$ .

Para os verbos que trabalham com locuções adnominais adjetivas ("ser", "parecer"), a análise sintática é:

$v(i, \text{loc\_adn\_adj}, \dots) :- v(0, \text{loc\_adn\_adj}, \dots),$   
 $\text{adj}(3, \dots).$

sendo que a interpretação semântica é:

$(v(0, \text{loc\_adn\_adj}, \dots))' (\text{int}(\text{adj}(3, \dots)))'$ .

Em relação à ausência de complementos para qualquer um destes tipos de verbos, é assumido aqui a aplicação da Convenção do Argumento Opcional, assim formulada:

Se  $b' = \dots(\text{int } a')\dots$ , onde  $b$  imediatamente domina um constituinte opcional  $a$  e  $\text{int } a'$  é do tipo  $\langle s, \langle \langle e, t \rangle, t \rangle \rangle$ , então quando  $a$  é omitido vale  $b' = \dots(\text{lambda int } P [(Ex) P(x)])\dots$

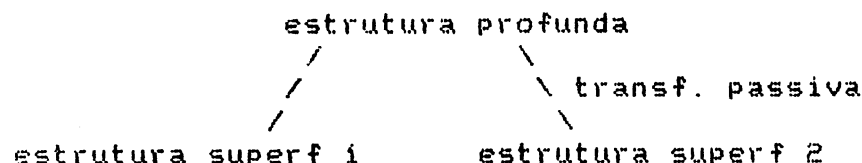
Isto estabelece então que se um complemento não for encontrado numa sentença, ele é suposto existir.

## 6.2 Análise da passiva

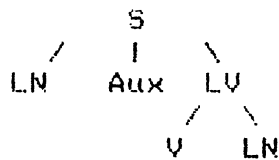
### 6.2.1 Crítica à abordagem transformacionalista

Dentro dos marcos das gramáticas transformacionalistas, é geralmente aceito que uma gramática de estrutura de frase livre do contexto não pode atingir a relação existente entre sentenças ativas e suas correspondentes passivas; a idéia é que a derivação das passivas envolve necessariamente uma transformação referida como Passiva /CHO 57/. Esta posição está baseada na assunção de que a relação ativa-passiva seja de natureza estrutural e que deve ser expressa por uma transformação. No que se segue, tanto essa assunção quanto essa solução no campo das gramáticas transformacionais são rebatidas.

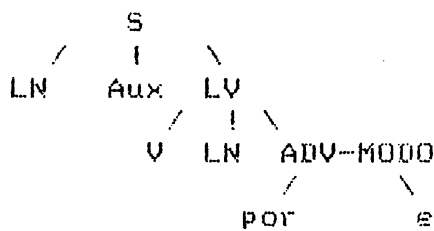
Na teoria transformacionalista antiga /CHO 57/, a relação ativa-passiva foi vista a partir de uma transformação opcional que era aplicada a uma estrutura profunda comum tanto para a sentença ativa quanto para a passiva.



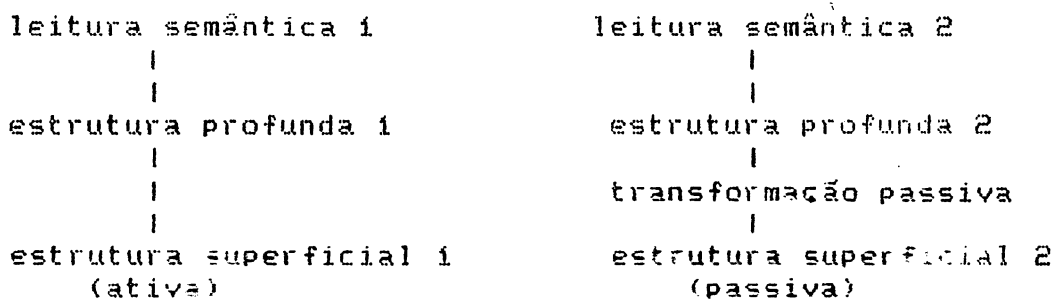
Esta situação muda com a publicação de Katz e Postal /KAT 64/ e na Teoria Standard de Chomsky /CHO 65/: em ambos os trabalhos é defendido que as sentenças ativas e passiva devem ser derivadas de diferentes estruturas profundas, sendo a transformação passiva uma regra de aplicação obrigatória. A estrutura profunda para a forma ativa, segundo o modelo do "Aspects" é



e a estrutura profunda para a forma passiva é



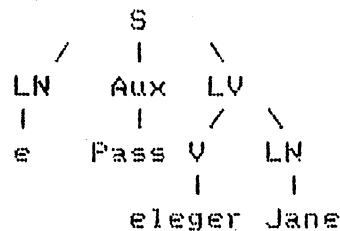
onde 'e' refere um elemento vazio. Sob esta análise, a transformação passiva não relaciona a derivação de uma sentença ativa a sua forma passiva, mas o modelo abaixo é o aceito:



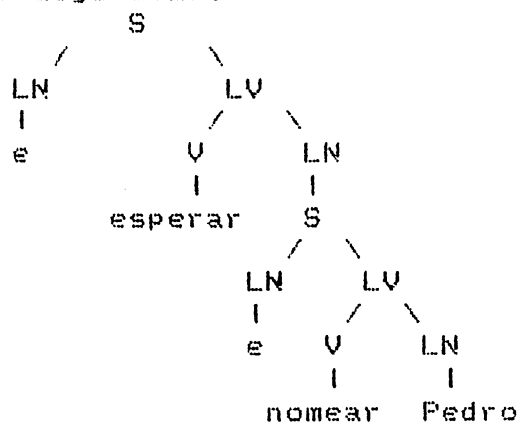
Nesta proposta, as derivações para ativas e passivas são distintas: nenhum nível de representação é comum. O mesmo ponto é defendido em Katz e Postal /KAT 64/ e em Lakoff /LAK 71/. Além disso, como as representações sintáticas profundas da ativa e da passiva não são idênticas, a relação ativa-passiva é estabelecida pela colocação de uma equivalência entre estruturas, em algum nível da derivação: Katz e Postal /KAT 64/ supõem uma equivalência ao nível de estrutura profunda e Chomsky /CHO. 65/ em termos de leitura semântica. Em oposição a isso, a Teoria Standard Extendida supõe que as funções semânticas se aplicam ao nível das estruturas superficiais: daí a estrutura superficial da passiva pode ser gerada pela base /EMO 76/ e a regra de transformação passiva pode ser vista como desnecessária.

Um problema para a Teoria Standard e a Standard Extendida é o tratamento das passivas truncadas (isto é: sem a locução preposicional por-LN). As passivas truncadas podem ser geradas transformacionalmente ou por uma regra de elipse (que apaga a locução por-LN), ou mesmo por uma regra que obrigatoriamente antepõe o objeto da estrutura profunda na posição do sujeito da estrutura profunda, quando este último for lexicalmente vazio. A análise via elipse derivaria uma passiva truncada como "João foi mordido" a partir de uma regra opcional que apagaria uma locução como "por alguém"; porém este apagamento não poderia violar a Condição de Recuperação de Apagamentos, porque o elemento apagado é um representante pronominal da categoria substantivo /CHO 65/. O problema é que as locuções que devem ser apagadas podem não ser representações pronominais: "A Alemanha foi atacada" seria derivada de "A Alemanha foi atacada por seus inimigos", e isto violaria a Condição de Recuperabilidade de Apagamento. Além disso "Jane foi eleita presidente do clube" não é sinônimo de "?Jane foi eleita presidente do clube por alguém".

Uma outra solução para as passivas truncadas (nos moldes da Teoria Standard Extendida Revisada) envolve estabelecer a estrutura profunda como



onde aparece um nodo vazio na posição sujeito. Como o nodo vazio é considerado como lexicalmente vazio, ele não é sujeito à interpretação semântica /EMO 76/. Daí que o sujeito vazio obrigaria a pré-posição do objeto na posição do sujeito e a passivização do verbo. Nesta análise, os problemas referidos acima não valem; porém mesmo na ausência de um sujeito na estrutura profunda na passiva truncada, é necessário uma transformação para realizar apagamentos. Por exemplo, a regra de apagamento de complemento sujeito seria necessária para distinguir as duas derivações "Pedro espera ser nomeado" e "É esperado que Pedro seja nomeado". Se as duas sentenças forem derivadas a partir de nodos vazios, a estrutura profunda para ambas seria algo como:



e a última sentença deveria então ser derivada por uma pré-  
 posição da LN "Pedro" duas vezes. A derivação de "Pedro espera  
 ser nomeado" é mais problemática, porque ela deve envolver uma  
 pré-posição não motivada que não resulta na passivização do verbo  
 matriz. Logo o uso dos nodos vazios não aparece como motivado  
 empiricamente.

Outras críticas ao tratamento transformacional das  
 passivas poderiam ser colocadas. Assim, por exemplo, não existe  
 razão para esperar que cada locução nominal objeto não possa  
 funcionar como sujeito em sentenças passivas, nos moldes do  
 transformacionalismo. Porém existem os exemplos:

João viu-se no espelho  
 \*Se foi visto por João no espelho  
 Maria e Pedro amam um ao outro  
 \*Um ao outro são amados por Maria e Pedro  
 1943 encontrou Pound na Itália  
 \*Pound foi encontrado por 1943 na Itália  
 Maria ficou minha amiga  
 Minha amiga foi ficada por Maria

exemplos estes tirados de /FER 78/ e /BAC 80/.

### 6.2.2 Tratamento em regras livres do contexto

Uma primeira solução para o problema da passiva, dentro  
 do campo das gramáticas livres do contexto, seria a utilização da  
 maquinaria das meta-regras, discutida antes. A utilização de uma  
 meta-regra está de acordo com a posição de /KEE 80/, pela qual  
 uma passiva é uma operação locucional, não sendo uma operação  
 lexical nem uma operação sentencial. A idéia é que uma LV passiva  
 difere sintaticamente de uma LV ativa correspondente porque o  
 objeto direto LN não está presente na LV ativa e porque pode  
 existir uma construção tipo [por-LN] na passiva; o resto  
 permanece como estava na ativa. Esta visão permite então  
 introduzir as locuções verbais passivas através da seguinte meta-  
 regra:

$$\langle v(i, \text{trans}) \rightarrow v(0, \text{trans}) n(2, \dots), v'(\text{int } n') \rangle$$

$$=)$$

$$\langle v(i, \text{pass}) \rightarrow v(0, \text{pass}) p(2, \text{por}_y), \text{lambda } x \setminus v'(\text{int } \text{lambda } y \setminus (y'(\text{int } \text{lambda } P \setminus P(x))))(y) \rangle$$

isto é: para cada regra de locução verbal ativa que expande a  
 locução verbal como um verbo transitivo seguido por uma locução  
 nominal, existe uma regra de locução verbal passiva que expande a  
 locução verbal como um verbo seguido por uma locução  
 preposicional com "por". A manipulação semântica substitui uma  
 variável tipo intensão de locução nominal na tradução da locução  
 verbal, aplica a função resultante ao agente int y', e abstrai na  
 sentença aberta resultante para produzir uma função que o  
 significado do tipo de locução verbal requer.

Em relação às passiva truncadas, poderia ser assumido  
 que na ausência de uma locução preposicional com "por", seria  
 aplicada uma convenção chamada aqui de Convenção do Argumento  
 Opcional, estabelecida acima e aqui repetida:

Se  $b' = \dots(\text{int } a')\dots$ , onde  $b$  imediatamente domina um constituinte opcional  $a$  e  $\text{int } a'$  é do tipo  $\langle s, \langle \langle e, t \rangle, t \rangle$ , então quando "a" é omitido vale  $b' = \dots(\text{lambda int } P [(Ex) P(x)])\dots$ .

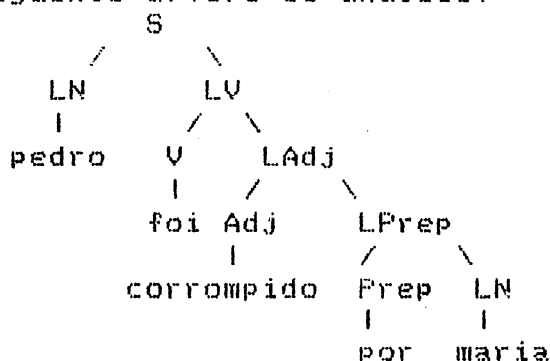
As locuções verbais passivas podem então ser introduzidas pela seguinte regra livre do contexto:

Sx1.  $v(i, \text{aux}) \rightarrow v(\emptyset, \text{ser}) v(i, \text{pass})$

Tx1.  $\text{lambda } P [(v(\emptyset, \text{ser}))' (\text{int } (v(i, \text{pass}))' (P))] ]$

ficando a meta-regra acima responsável pela identidade de interpretação da forma ativa e de sua correspondente forma passiva.

Uma segunda proposta, mais simples, pode ser feita, assumindo que os participípios passados sejam tratados como adjetivos. Esta proposta de cunho lexicalista para as passivas aparece em /WAS 77/ /VIE 78/ /PIM 81/ /LEM 84/. Os argumentos para isso são os seguintes: a) os participípios passados podem receber o sufixo -íssimo de superlativo ("riscos calculadíssimos"); b) os participípios passados concordam em gênero e número com o substantivo. Esta visão elimina a necessidade da meta-regra porque a uma construção sentencial transitiva corresponde uma construção passiva com adjetivo, na qual o adjetivo predica o nome que é o objeto direto do verbo da construção ativa. Assim "Pedro foi corrompido por Maria" tem a seguinte árvore de análise:



Em defesa desta proposta é possível estabelecer os seguintes paralelismos de estruturas:

[<LN> Det N Adj]    [<S> LN ser [<LAdj> Adj LP]]    [<LN> Det N LAdj]

A porta fechada                    O rapaz foi chamado por Pedro                    O rapaz chamado por Pedro

A saída inteligível                    O linguista foi entusiasta sobre o lexicalismo                    O linguista entusiasta sobre o lexicalismo

onde a primeira fila mostra construções com participípios passados e a segunda mostra construções com adjetivos.



Pode ser contra-argumentado que os predicados passivos não são adjetivos porque eles não ocorrem em algumas construções onde ocorrem adjetivos, como [LN parece Adj]:

Luiz parece inteligente

?Luiz parece ajudado

mas isto não parece ser algo essencial para a recusa desta segunda proposta.

Esta saída tem contudo consequências sobre o tipo de análise proposto aqui para as construções verbais com auxiliares. Uma primeira consequência desta proposta é que a análise de [ser + \_do] realizado no tratamento das construções verbais com auxiliares deve ser alterada, pois, na proposta da análise da passiva como adjetivo, o auxiliar "ser" deve ser visto como um copulativo que pega um adjetivo. Além disso, uma construção como "Pedro está ansioso por liberdade" é analisado tal que "ansioso por liberdade" é uma locução adjetiva semelhante a "corrompido por Maria" no exemplo dado acima. Desta forma, o agente da passiva não recebe aqui nenhuma sintaxe particular: ele é tratado como complemento de adjetivo. A correspondência entre a forma transitiva ativa e a locução adjetiva referida fica capturada pela seguinte regra de correspondência lexical entre o verbo no particípio passado e o adjetivo

forma fonológica		<-->	forma fonológica + _do
verbo			adjetivo
a ----- b			b -----
z relaciona a com			b é afetado pela
b			relação z

Em relação aos particípios passados após "ter", ele aí é invariável ("Maria tinha sofrido de tuberculose"). Outros exemplos de particípios passados invariantes são "Maria ri forçado", onde o particípio passado funciona como um advérbio. Uma análise para "Maria tem estudado este problema" é então:



Esta segunda forma de tratamento da passiva possui defesa na história da língua portuguesa, pois, segundo Câmara Junior /CAM 79/, os particípios passados ligados a "ter" entraram no Português a partir de construções onde estes particípios exerciam a função de predicativos de objeto direto, sendo sua função a de um adjetivo. Após um período de variação livre da língua é que a concordância do particípio com o objeto direto se esvaiu; nesta mesma época, a construção se estendeu também para os verbos intransitivos. Em suma: de uma função adjetival, o particípio passado passou a ter uma função adverbial.

Uma terceira solução seguiria a idéia de "transformação governada pelo léxico" /BRE 78/ /WAS 77/ /DOW 78/, mas formalizando isto a partir de uma regra lexical numa Gramática de Montague. Dentro desta proposta, uma sentença como "O livro foi rasgado" (passiva truncada) teria como árvore de análise:

```

      um livro foi rasgado, t, 4
      /      \
um livro, T, 2      foi rasgado, IV, 20
                    /      \
                  _do, LP      rasgar, TV
  
```

e, por outro lado, a sentença "Um livro foi rasgado por Pedro" teria como árvore de análise:

```

      um livro foi rasgado por Pedro, t, 4
      /      \
um livro, T, 2      foi rasgado por Pedro, IV, 20
                    /      \
                  por Pedro, LP, 21      rasgar, TV
                    /      \
                  por, LP/T      Pedro, T
  
```

sendo uma característica importante destas análises o papel atribuído ao formador do particípio passado (\_do) e a sua contraparte transitiva "por": o formador \_do combina-se com um verbo transitivo e gera uma locução verbal passiva; a preposição "por" combina-se com uma locução nominal gerando uma expressão, a qual é combinada com um verbo transitivo para dar uma locução verbal passiva.

O efeito semântico de "por" é o de reverter a interpretação do sujeito e do objeto do verbo transitivo. Isto pode ser feito no "espírito do PTQ" pelo assinalamento de uma tradução para "por" que leve em conta que "por" é um mapeamento de uma relação diádica R (a denotação de um verbo transitivo) em outra relação R1, tal que para os indivíduos x e y

$\langle x, y \rangle$  pertence a R sse  $\langle y, x \rangle$  pertence a R1.

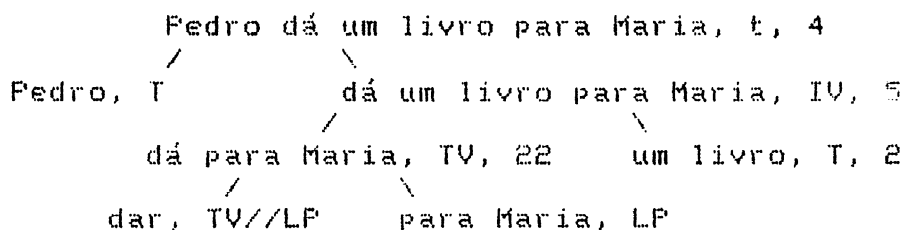
Por outro lado, o efeito semântico de \_do (que forma as passivas truncadas) é similar na reinterpretação do sujeito passivo como objeto do verbo ativo; porém no local do sujeito ativo deve ser colocado um quantificador existencial. Em termos de referência, o significado de \_do mapeia uma relação R no conjunto K tal que para um indivíduo x

x pertence a K sse existe algum y tal que  $\langle y, x \rangle$  pertence a R.

O problema com esta proposta é que, no PTQ, Montague adotou um esquema semântico que trata os verbos transitivos como funtores da categoria TV (ou IV/T) que se combinam com um argumento do tipo locução nominal gerando locuções da categoria IV. Isto então prediz que um verbo transitivo com objeto passa a funcionar sintaticamente, nos moldes do PTQ, como um verbo intransitivo sozinho; daí que a relação "sujeito\_de" fica igual para os verbos transitivos com objeto e para os intransitivos: uma locução nominal é um sujeito quando se combina com um IV (lexical ou complexo - TV/T + T) para formar uma sentença.

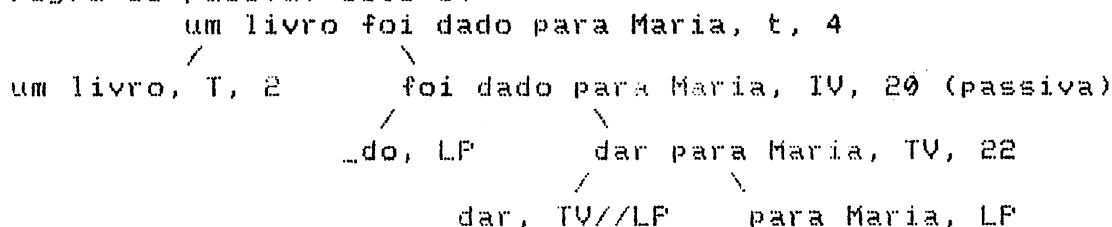
A solução deste problema envolve assumir que, no caso dos verbos com tres argumentos, existe uma categoria funtor TV//T que se combina com uma locução nominal formando um TV complexo. É possível assim definir a relação "objeto direto de", comum aos verbos com dois e aos verbos com tres argumentos, da seguinte forma: uma locução nominal é um objeto direto quando ela se combinar com um TV (lexical ou complexo) e gerar um IV. Por outro lado, uma locução preposicional é um objeto indireto quando ela se combinar com um TV//LP e gerar um TV.

Com esta maquinaria é possível então propor a seguinte árvore de análise (simplificada):

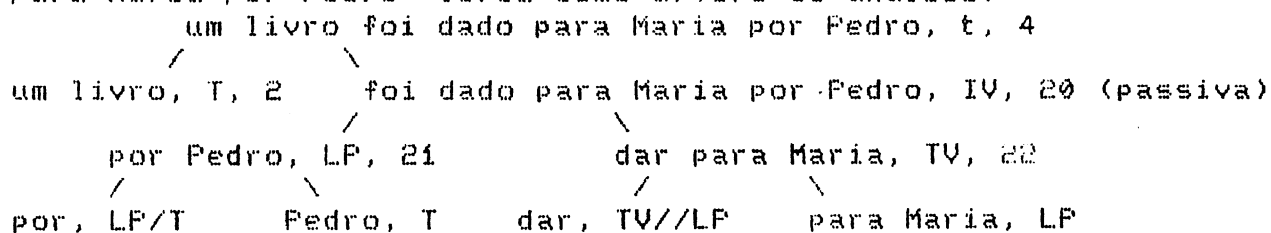


Porém esta derivação requer que a regra do FTQ que combina verbos transitivos com objetos seja complicada, pois a regra deveria especificar que, na combinação de uma expressão TV com uma locução preposicional, esta locução preposicional deve ser colocada imediatamente antes da primeira expressão básica na expressão TV. Além disto ser uma regra dependente de contexto, há complicações importantes para expressões do tipo [persuadiu Pedro a viver].

A saída envolve estabelecer que a análise de [dar para Maria] como TV é correta, porque este verbo pode ser passivizado ("Um livro foi dado para Maria"). Assim, sob a regra passiva que muda a categoria verbal, a locução TV deve ser a entrada para uma regra de pasiva. Isto é:



Por outro lado, uma sentença como "Um livro foi dado para Maria por Pedro" teria como árvore de análise:



Se isto for feito, é necessário acrescentar novas categorias ao FTQ, tais como TV//LP, IV/TV (ou PP - formadores de participio passado) e PP/T (formadores de participio passado transitivos). É exemplo de TV//LP o verbo "dar"; é exemplo de IV/TV a forma \_do; é exemplo de PP/T o verbo "por".

As novas regras acrescentadas ao PFG seriam as listadas abaixo:

S5 (revisado do PTQ): Se a é um TV e b é um T, então F5(a, b) é um IV, onde (i) se b não é uma variável, então F5(a, b) vem de a por inserção de b após o primeiro verbo básico em a, ou (ii) se b = elen, então F5(a, b) é o resultado da inserção de elen após o primeiro verbo básico em a.

T5: a' (int b')

Exemplos:

F5(ama, Maria) = ama Maria,

F5(dá para Maria, um livro) = dá um livro para Maria

F5(persuade a viver, ele3) = persuade-o a viver

Regra da Passiva:

S20 : se a é um PP e b é um TV, então F20(a, b) é um IV e (i) se a = \_do, então F20(a, b) = foi b1, onde b1 vem de b pela substituição do primeiro verbo de b pelo seu participio passado, ou (ii) se a ≠ \_do, então F20(a, b) = foi b1 a, onde b1 é como em (i).

T20: a' (int b')

Exemplos:

F20(por Pedro, ferir) = foi ferido por Pedro

F20(\_do, dar para Maria) = foi dado para Maria

Regra da formação da locução com "por":

S21: Se a é um PP/T e b é um T, então F21(a, b) é um PP, onde F21(a, b) = a on se b = elen; de outro modo vale ab.

T21. a' (int b')

Exemplos:

F21(por, Pedro) = por Pedro

F21(por, ele5) = por o5 (= pelo)

Regra da formação do objeto dativo:

S22: Se a é um TV//LP e b é um T, então F22(a, b) é um TV, onde F22(a, b) = a para on se a = elen; de outra forma F22(a, b) vale a para b

T22: a' (int b')

Exemplo:

F22(dá, Maria) = dá para Maria

As traduções para as novas constantes lógicas são:

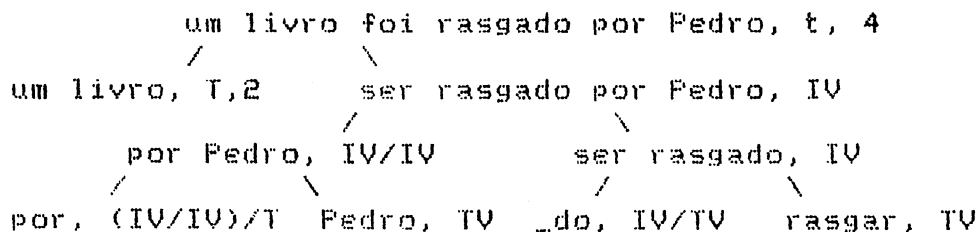
por => lambda P lambda R lambda x P .

(int lambda y [ext R (y, int lambda P [P (x)])]) ]]

onde R é uma variável do tipo <s, f(TV)>.

\_do => lambda R lambda x (Ey) [ext R(y, int lambda P [P(x)])]

Poderia ser pensado que seria mais natural produzir as passivas completas a partir das passivas truncadas pela adição de uma locução com "por" (como um modificador opcional), usando IV/TV para a forma \_do, como abaixo:



Mas esta análise não é semanticamente viável porque a árvore acima não pode ser logicamente equivalente a "Pedro rasgou um livro"; a razão é que, criado um particípio passado truncado "rasgado", o feito semântico é reduzir uma relação a um conjunto de indivíduos: só a partir deste conjunto não é possível recuperar o conjunto de pares que faz a relação original.

Um problema sério com esta terceira proposta é a nova formulação de S5, a qual estabelece um regra dependente do contexto, na medida em utiliza a noção de "janela à direita" do primeiro verbo básico /BAC 80/. Uma maneira de resolver isto é através de uma meta-regra que substitua S5.

No caso das passiva truncadas, é possível assumir que a denotação de um locução verbal é uma relação n-ária (como faz Dowty /DOW 82/), tal que o efeito semântico das regras que alteram estas relações é a de realizar operações algébricas simples sobre o significado do verbo. Assim uma relação n-ária seria reduzida a uma relação n-1-ária, sendo rearranjados os respectivos lugares da relação. Desta forma a passiva sem agente ficaria como um exemplo de uma regra que muda uma relação pela eliminação do argumento sujeito e sua formulação seria:

Sxx: Se a é um TV, então Fxx(a) é um IV, onde Fxx(a) = ser a+\_do

Txx:  $\lambda P [a'(P) (\text{int } \lambda P (Ex) P(x)) ]$

Nessa visão, uma locução verbal transitiva seria uniformemente combinada com uma locução nominal objeto direto não simplesmente por concatenação das duas locuções, mas por inserção da locução nominal objeto após a primeira palavra da locução verbal transitiva - a qual é uma operação dependente do contexto. No entanto, este efeito pode ser reconstruído através de uma gramática livre do contexto que utilize meta-regras; para isso a meta-regra deve especificar que para uma regra que produz uma locução verbal transitiva, existe outra regra que produz uma IV que é exatamente como a anterior exceto que possui uma locução nominal adicional imediatamente após o verbo. Assim a categoria verbo transitivo fica como uma categoria fantasma cuja única função sintática é a de induzir IV com locução nominal objeto por meio desta meta-regra.

Uma quarta proposta de solução para a análise da passiva é estabelecer uma classificação dos verbos transitivos do Português, segundo os argumentos que eles pegam. Existiriam assim as categorias LVI (locução verbal intransitiva), LVI/LN (locução verbal transitiva), LVI/X (com X diferente de LN), (LVI/LN)/X (locuções verbais transitivas complexas), etc. Alguns exemplos seriam: LVI/Predicado\_nominal (ser), LVI/Predicado\_adjetival (ser), (LVT/LP[para])/LN (dar), (LVI/LP[para])/LV (prometer).

A estas categorias se acrescentaria a categoria sintática LVP (locução verbal passiva), a qual pode ser combinada com "ser" para formar uma LVI; mas uma LVP pode ocorrer sozinha também. Semanticamente uma LVP é um predicado que expressa alguma categoria t/e, denotando conceitos individuais. As regras da gramática seriam:

Regra da LVP sem agente:

Sxx: Se a é uma LVT, então  $\_do(a)$  é uma LVP, onde  $\_do(a)$  é a forma particípio passado de a.

Txx:  $(\_do(a))' = \text{lambda } x (E y) [y' (\text{int } \text{lambda } P P(X))] (y)$

Regra da LVP com agente:

Sxx: Se a é uma LN e b é uma LVT, então a expressão  $[\_do(b)$  por a] é uma LVP.

Txx:  $(\_do(b) \text{ por } a)' = \text{lambda } x a' (\text{int } \text{lambda } y (y' (\text{int } \text{lambda } P P(x)))) (y)$ .

A tradução de "João foi lesado" seria:

$\text{lambda } j (E y) [E \text{lesar}' (y, j)]$

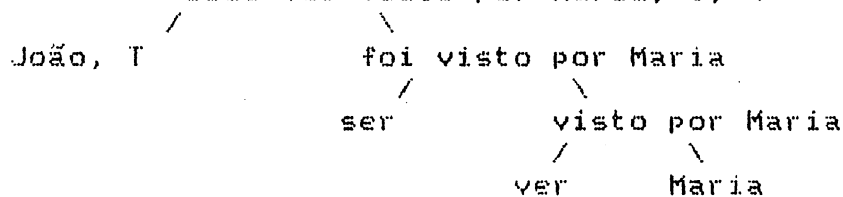
isto é: João é tal que alguém lesou ele, e a de "João foi visto por Maria" seria:

$\text{lambda } j [E \text{ver}' (m, j)]$

isto é: João é tal que Maria viu ele.

Para derivar sentenças com LVP, é necessário assumir que o verbo "ser" seja da categoria LVI/LVP, tal que "João foi visto por Maria" teria como árvore de análise:

João foi visto por Maria, t, 4



Como comentário final, deve ser observado que esta quarta proposta não é antagônica da primeira, encaixando-se mesmo nela.

### 6.3 Tratamento das construções com auxiliares

#### 6.3.1 Tratamento tradicional dos auxiliares

As análises tradicionais das construções com verbos auxiliares podem ser divididas em duas: a hipótese da estrutura de frase e a hipótese do verbo principal.

A hipótese da estrutura de frase existe na tradição transformacional e aparece originalmente em /CHO 57/ na forma das seguintes regras para o Inglês:

S --> LN AUX LV

AUX --> Tempo (Modal) (have+en) (be+ing)

onde os elementos entre parênteses são opcionais. A primeira regra coloca a categoria AUX (denotando "auxiliar") ao mesmo nível que as categorias LN (locução nominal) e LV (locução verbal).

Já a segunda regra estabelece que a categoria AUX envolve uma coleção de elementos. Estas regras persistem ainda no campo transformacional, apesar das pequenas modificações acrescentadas por /EMO 76/ e /CUL 76/, entre outros.

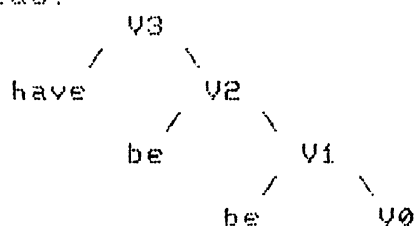
A proposta transformacionalista parece originar-se da tradição estruturalista e Sampson /SAM 79/ observa que esta análise não é compatível com a teoria formal das transformações, porque a postulada Transformação sobre Auxiliar (conhecida na literatura como "Affix-hopping") ignora várias Condições de Transformação. Por outro lado, é difícil dizer que um elemento lexical ou gramatical é um AUX no sentido de que "morrer" é um verbo, porque não existem itens no léxico que pertencem à categoria AUX; além disso AUX não é uma categoria locucional (não lexical), porque AUX não possui um núcleo. E mais: nenhuma transformação afeta AUX /CHO 85/. A única função de AUX aparece na teoria de /CHO 80/ conhecida como Teoria Standard Extendida Revisada: AUX proíbe certos itens serem comandados por outros.

A segunda hipótese ocorre no campo da Semântica Gerativa e originalmente aparece em /ROS 69a/. Nela não existe a categoria AUX e todos os verbos auxiliares são tidos como verbos principais que tomam complemento sentencial vazio, necessitando tudo isso a aplicação de algumas transformações.

Akmajian, Steele e Wasow /AKM 79/ propõem uma análise intermediária, aceitando a categoria AUX e postulando a existência do traço aux; suas regras possuem a forma:

S --> LN AUX V3  
 AUX --> Tempo Do  
 AUX --> Modal Do  
 Vn --> ( V(aux) ) Vn-1

onde os parenteses em torno de V(aux) indicam que este elemento é opcional. É importante aqui a utilização do sistema X-barrado de /JAC 77/; eles procuram mostrar que o perfeito "have" requeriria complemento do tipo V2, que o progressivo "be" requeriria complemento do tipo V1, e que o passivo "be" requeriria complemento do tipo verbo principal. A análise é então:



O tratamento das construções com auxiliares apresentado aqui é diferente do esboçado acima. A motivação geral do autor é a de apresentar um tratamento das construções com verbos auxiliares utilizando idéias que se enquadrem no "espírito" do trabalho de Montague referido como PTQ, não utilizando transformações, mas sim regras livres do contexto. O que aqui é apresentado é uma expansão do que está em /MOS 86/. Esta proposta não postula a existência da categoria AUX, mas utiliza o traço aux, entre outros, referido acima. Ela é mostrada a seguir.

### 6.3.2 Tratamento lógico das construções com auxiliares

Em primeiro lugar este trabalho não distingue entre tempo composto e conjunção perifrástica. Isto se justifica porque não existe entre os gramáticos tradicionais da língua portuguesa uma definição rigorosa de tempo composto nem de conjunção perifrástica. Assim, por exemplo, Dias /DIA 59/, Melo /MEL 68/, Reis /REI 71/, Ribeiro /RIB 85/ e a Nomenclatura Gramatical Brasileira (apud Juca Filho /JUC 58/) assumem que são tempos compostos apenas as sequências verbais formadas por "ter" e "haver" mais participípio. Porém Bechara /BEC 66/, Goes /GOE 17/, Ribeiro /RIB 26/ incluem aí as construções tipo [ser + participípio]. /PER 09/ inclui [estar + participípio] e exclui as construções com "ser". Juca Filho /JUC 58/ inclui "ter", "haver", "ser" e "estar" como formando tempos compostos. Said Ali /SAI 57/ refere que as construções tipo "tinha feito", "tenho feito", etc não são tempos compostos, mas sim conjunções perifrásticas; o mesmo Said Ali /SAI 63/ declara que as denominações tipo pretérito perfeito composto, pretérito mais que perfeito composto, etc são inexpressivas. Bueno /BUE 68/ não distingue entre tempo composto e conjunção perifrástica. Lima /LIM 64/ estabelece que as construções com "querer", "ter", "estar", "ficar", "ir" mais gerúndio, ou infinitivo ou participípio são todas tempos compostos. Câmara Junior /CAM 64/ não distingue entre tempo composto e conjunção perifrástica, chamando todas estas construções de conjunções perifrásticas.

O formalismo aqui apresentado, nos moldes de /MOS 85/, procura capturar os padrões sintáticos das construções com auxiliares através de regras livres de contexto, acrescentadas de símbolos complexos, e de meta-regras. A questão da interpretação semântica das árvores sintagmáticas é apresentada à luz de um formalismo originário no PTQ de Montague /MON 73/. É assumido aqui também a Convenção de Traços para o Núcleo de Locução /GAZ 82/ /MOS 85/.

Este formalismo parte inicialmente da análise dos padrões das regularidades sintáticas das construções verbais com auxiliares. Estes padrões são apresentados a seguir e sua formulação segue a forma

a ( b ) + c

onde 'a' denota a característica de um conjunto de verbos, 'b' mostra exemplos de 'a' e 'c' denota o tipo de verbo que segue 'a':

verbos modais (poder, dever, querer, tencionar, desejar, haver de, ter de, ter que, estar para, saber) + infinitivo

verbos ingressivos (começar, entrar, continuar, chegar, por-se, deitar, largar, lançar, meter-se, romper, disparar) + a (preposição) + infinitivo

verbos sensitivos ou declarativos (declarar, julgar, prometer, jurar, admirar, estranhar, sentir, lastimar, temer, recear, evitar, determinar, preferir, desejar, merecer, consentir, conceder, proibir, impedir, aconselhar) + infinitivo



verbos de movimento (andar, ir, vir, ficar, permanecer, continuar) + gerúndio  
ir + infinitivo

auxiliares causativos (fazer, mandar, deixar) + infinitivo

verbos acurativos (começar a, costumar, cessar de, continuar a, acabar de, tornar a) + infinitivo  
ter/haver + participípio

ficar/sair/deixar + participípio (com sentido resultativo)

estar + gerúndio  
estar + participípio (raro)

ser + participípio (voz passiva)

Em relação à ordem interna da construção, a passiva vai para o final. [estar + \_ndo] só pode ocorrer depois de [ter + \_do] ou [modal + \_r] e antes de [ter + \_do]. São exemplos: "deve ter estado comprando", "tem estado comprando", "\*tem devido comprar", "\*está devendo comprar", "\*está tendo comprando", "\*tem devido estar comprando". Isto implica então na construção:

[modal + \_r] [ter + \_do] [estar + \_ndo] [ser + \_do] [verbo]

onde as quatro primeiras formas são opcionais. Nesta fórmula, a desinência colocada para o primeiro elemento é na construção final acrescentada ao segundo elemento; a desinência colocada para o segundo elemento é acrescentada ao terceiro elemento, e assim sucessivamente.

Neste padrão, o slot correspondente a modal imprime o caráter de irrealidade (futuro, habilidade, possibilidade, probabilidade, necessidade, obrigação); o slot correspondente ao verbo "ter" imprime o caráter de anterioridade; já o slot para o verbo "estar" aponta para um caráter de estatividade. Esta análise segue aproximadamente as idéias de Givón /GIV 82/ e de Bickerton /BIC 75/ para as línguas creoles. Este sistema no seu conjunto permite conhecer a ordem temporal de ocorrência dos eventos passados, distinguir entre entrada sensorial e imaginação (fato versus irrealis), dizer se um evento ocorreu sem interrupção (contínuo) ou com interrupção (iteração).

As regras propostas a seguir seguem o "espírito do PTQ", utilizando a metodologia apresentada em /MOS 85/, envolvendo regras sintáticas livres do contexto com símbolos complexos e interpretação em lógica intensional. Como está em /MOS 86/, o tratamento das construções com auxiliares utiliza os seguintes traços para a constituição dos símbolos complexos:

part - participípio  
ger - gerúndio  
inf - infinitivo  
fin - finitivo  
pas - passiva

os quais são mutuamente exclusivos; outros traços são:

aux - auxiliar  
 asp - aspectual  
 cop - copulativo

Assumindo ainda o sistema X-barra apresentado antes e em /MOS 84f/ e a Convenção de Traços para o Núcleo da Locução também apresentada antes e em /MOS 85/, aqui é proposto um sistema de regras sintáticas do tipo:

$v(2, a, \dots) \rightarrow v(0, a, \dots), v(1, b, \dots)$

onde  $v(0, a, \dots)$  é colocado como núcleo da construção e logo recebe todos os traços de  $v(2, a, \dots)$ .

Em termos do PTQ, isso pode ser assim parafraseado:

Sxx. Se  $x$  é um TV designado por possuir os traços 0 (na notação X-barra) e  $a$ , e se  $y$  é um TV designado por possuir os traços 1 e  $b$ , então  $xy$  é um TV designado por possuir os traços 2 e  $a$ .

A interpretação semântica desta árvore de análise

é:

Txx.  $\lambda P [ (v(0, a, \dots))' (int (v(1, b, \dots))' (P)) ]$ .

onde  $P$  denota a locução nominal sujeito.

Assumindo que o traço aux faz parte do argumento 'a', para simplificar é estabelecida a seguir uma tabela onde constam 'n' (número da regra), 'a' (argumento de  $v(2, \dots)$  e de  $v(0, \dots)$ ), 'b' (argumento de  $v(1, \dots)$ ) e exemplos de verbos que assumem estas combinações, levando em conta os padrões sintáticos mostrados acima:

n	a	b	exemplos
2	fin	inf	modais, declarativos, sensitivos, causativos
3	fin prep	inf	preposições [a], [de]
4	asp	part	ter, haver
5	asp	ger	estar, verbos de movimento
6	cop	part pass	ser
7	asp	part pass	estar

As regras correspondentes possuem a forma:

S2. Se  $x$  é um TV designado por possuir os traços aux, fin e modal (ou declarativo ou sensitivo ou causativo), e se  $y$  é um TV designado por possuir o traço inf, então  $xy$  é um TV.

S3. Se  $x$  é um TV designado por possuir os traços aux, fin e prep, e se  $y$  é um TV designado por possuir o traço inf, então  $xy$  é um TV.

S4. Se  $x$  é um TV designado por possuir os traços aux, asp, verbo "ter" ou "haver", e se  $y$  é um TV designado por possuir os traços part, então  $xy$  é um TV.

95. Se x é um TV designado por possuir os traços aux, asp, verbo "estar" ou de movimento, e se y é um TV designado por possuir o traço ger, então xy é um TV.

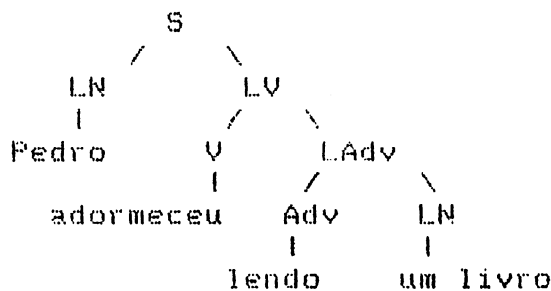
96. Se x é um TV designado por possuir os traços aux, cop, verbo "ser", e se y é um TV designado por possuir os traços part e pass, então xy é um TV.

97. Se x é um TV designado por possuir os traços aux, asp, e se y é um TV designado por possuir os traços part, pass, verbo "estar", então xy é um TV.

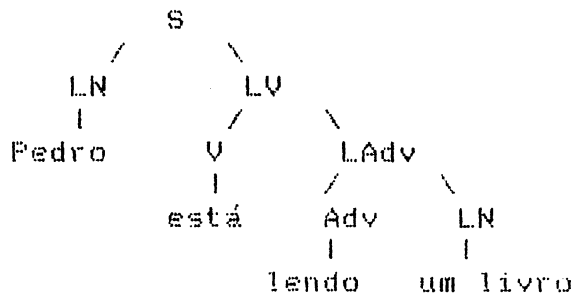
É preciso estabelecer também regras na gramática que relacionem o traço pass com a forma +\_do, o traço ger com a forma +\_ndo, o traço inf com a forma +\_r, o traço part com a forma +\_do. Além disso o escritor da gramática deve cuidar para que ela contenha restrições do tipo: se cop então aux, se asp então aux, se pass então ~aux.

Uma discussão maior sobre o caso n=6 ocorre na secção correspondente ao tratamento das passivas.

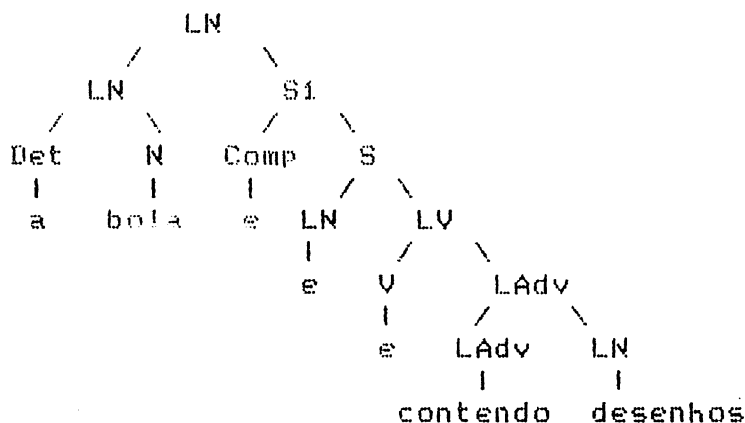
Em relação ao caso n=5, deve ser lembrado que as sentenças "Conversando a gente se entende" e "Pedro está assoviando" são analisadas nas gramáticas tradicionais do Português, colocando-se o verbo no gerúndio como uma oração subordinada adverbial. Isto ocorre porque os gramáticos tradicionais aceitam que o verbo no gerúndio modifica o verbo; esta postura é contemplada pela proposta elaborada acima. Porém deve ser lembrado que a forma gerúndio do Português é originária do caso ablativo do gerúndio latino /LEM 84/; ao lado disso, deve ser também recordado que muitos advérbios do Português vieram de ablativos latinos: todos os advérbios em \_mente, as expressões "de certo", "de novo", "falar alto", "comprar barato", etc. Uma proposta aqui seria admitir que os gerúndios são advérbios, e assim "Pedro está comendo..." não seria uma forma composta onde "estar" seja o auxiliar. Ao lado disso, existem construções como "Pedro adormeceu lendo o livro" e "Maria fala dormindo", as quais são analisadas pelos gramáticos tradicionais pela colocação de que o primeiro verbo não é um auxiliar para o gerúndio, pois o gerúndio é considerado por eles como um advérbio. Esta argumentação coloca os gerúndios como advérbios, tal que a sentença "Pedro adormeceu lendo um livro" teria a seguinte árvore:



e uma sentença como "Pedro está lendo um livro" teria como árvore o seguinte:



Já nas sentenças do tipo "A bola contendo desenhos deve ficar aqui", o gerúndio assume o papel de um adjetivo, o qual contudo não concorda com o substantivo. Uma solução seria analisar esta sentença como "A bola que está contendo desenhos deve ficar aqui", tal que valha:



onde "e" denota um nodo vazio.

Em relação à negação nas construções com auxiliares, considerando as expressões "não pode estar comendo massa", "pode não estar comendo massa", "pode estar não comendo massa", \*"pode estar comendo não massa", é possível estabelecer duas regras básicas:

$v(2, \dots) :- v(0, \dots), \text{neg}(\dots), v(1, \dots)$

$v(2, \dots) :- \text{neg}(\dots), v(1, \dots)$

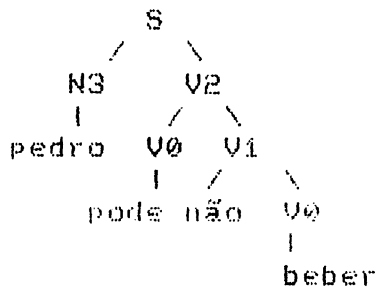
onde a primeira regra se aplica a construções tipo "pode não beber" e a segunda a "não pode beber" (bem como a outras construções sem modais). A interpretação da primeira regra acima é a seguinte:

$(v(0, \dots))' (" \text{int} (v(1, \dots))'$

e a da segunda é:

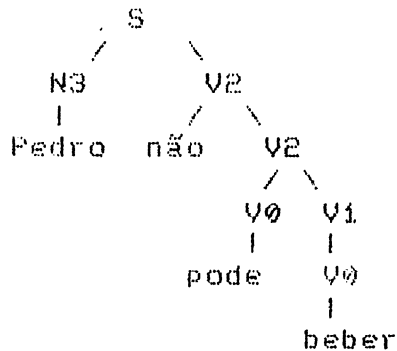
$" (v(1, \dots))'$

Uma aplicação para isso seria na análise de "Pedro pode não beber" e de "Pedro não pode beber". A primeira teria como árvore de análise simplificada:



sendo como interpretação semântica:

$[\lambda P \text{ [poder' (int beber'(P)) ]] (' pedro')}$   
 e a segunda teria como árvore de análise simplificada:



e sua interpretação semântica é:

$[\lambda P \text{ [poder' (int beber'(P)) ]] (' pedro')}$ .

No que se refere à topicalização da locução verbal, é possível adotar uma regra proposta em /MOS 86/ e semelhante à regra da topicalização da locução nominal /MOS 85/. Assim, se o topicalizado foi uma locução 'a' interna à locução verbal mais complexa, a regra é a seguinte:

$v(2, \dots) :- a(\dots), v/a(2, \dots)$ .

onde  $v/a(2, \dots)$  denota uma locução verbal com "buraco" do tipo 'a' e a interpretação semântica é:

$\lambda h(a, \dots) [ (v/a(\dots))' ] (a')$

sendo que isto se aplica a expressões tipo "indo, ele estava", "comprando, ele deve estar".

Esta análise faz uso da noção de dependência não ligada e utiliza a idéia de categoria derivada. A importância aqui é que o fenômeno da topicalização de locução verbal pode ser atingido pela assunção de que  $a = v(1, \dots)$

Por outro lado, o apagamento de uma locução verbal é tratado pela regra sintática:

$v(1, \dots) :- []$ .

e a interpretação semântica é

$r$

onde  $r$  é uma variável contextualmente ligada cujo domínio são as denotações de  $v(0, \dots)$ . Neste caso, as restrições para o Português no que diz respeito a apagamento de locução verbal são ainda pouco estudadas /KAT 81/. Porém uma aproximação inicial seria através da seguinte meta-regra:

$v(i, +aux, \dots) :- v(0, \dots), v(i, \dots)$

$==>$

$v(i, +aux, \dots) :- v(0, \dots), v(i, nulo, \dots)$

tal que ela toma uma regra de  $v(i, +aux, \dots)$  que se expande como  $v(0, \dots)$  seguido de  $v(i, \dots)$ , e daí adiciona o traço 'nulo' ao  $v(i, \dots)$  complemento. Isto assegura que  $v(i, \dots)$  gera um string vazio.

### 7.1 Introdução

Do ponto de vista semântico, um enunciado é um objeto abstrato no qual é atribuída uma dada propriedade a um elemento ou é estabelecida uma relação entre elementos; ou seja: um enunciado é uma predicacão. A explicitação do valor das variáveis (via operação linguística de determinação) tende a converter esta predicacão numa proposição.

As características do estado de coisas descrito e o seu modo interno de ser exprimem-se linguisticamente através da seleção de um predicado pertencente a uma certa classe e da forma aspectual exibida por este predicado; além disso, a localização do estado de coisas descrito, relativamente ao intervalo da fala, apresenta sua expressão linguística na categoria tempo. Por outro lado, a indicação do universo de referência relativamente ao qual esta predicacão é válida e a indicação do modo como o falante encara a ocorrência do estado de coisas descrito são expressas através da categoria linguística modalidade. Afóra as características ligadas ao predicador propriamente dito, existe a seleção dos argumentos deste predicador, tal que estes argumentos convertem-se em expressões com um certo valor referencial.

De todos os subsistemas gramaticais, provavelmente o mais complexo seja o conjunto tempo-aspecto-modalidade, referido acima. De um lado, estas três categorias são universais para todas as linguagens naturais conhecidas; de outro lado, elas constituem um grande instrumento de codificação da coerência das sentenças do discurso. Ao nível da semântica lexical, elas estão intimamente envolvidas com a estrutura-significação dos verbos ("predicadores"); ao nível da semântica mais de natureza proposicional, o conjunto tempo-aspecto-modalidade codifica várias facetas do estado de coisas descrita; ao nível do discurso, estas três categorias desempenham um papel no sequenciamento das proposições que compõem o discurso, nas relações entre figura e fundo no discurso e na indicação de seu tempo/verdade/certeza para o falante.

A reunião destas três categorias em torno do verbo não é accidental, na medida em que, das categorias lexicais, o verbo é a mais intimamente associada com a codificação de estados, eventos e ações na proposição. Numa visão histórica do desenvolvimento de uma linguagem natural, deve ser realçado que as categorias de tempo, aspecto e modalidade tendem a se unir em torno do verbo a partir de marcadores. Um ponto inicial deste processo de marcação é um verbo em si mesmo, na medida em que este verbo passa a servir como um auxiliar de outro. Quando este marcador torna-se mais especializado, na forma de um morfema gramatical, ele perde a sintaxe, a morfologia e a significação do verbo original e torna-se uma marca fonologicamente menor: torna-se um sufixo ou um prefixo do verbo principal /GIV 71/. Um exemplo típico em Português é a formação histórica do futuro do presente, a partir da clitização do verbo "haver" como sufixo do verbo principal.

De um modo simples, as categorias de tempo, aspecto e modalidade representam três diferentes abordagens para a experiência do tempo. O tempo verbal envolve basicamente, mas não exclusivamente, a experiência ou conceito de um tempo como pontos numa sequência. O aspecto envolve basicamente a noção de limites no transcorrer do tempo, em torno de um ponto de referência, além de outras noções a serem aqui defendidas. A modalidade, por seu turno, abrange a noção de realidade, no sentido de ter existência factual em algum/nenhum tempo real, ou ter existência potencial em algum tempo a vir.

A abordagem a ser seguida aqui envolve tratar todas estas características (aspecto, modo, tempo verbal) ligadas ao elemento predicador, bem como tratar o problema da interação entre dois ou mais predicadores (construções com auxiliares, orações subordinativas), dentro do modelo da Gramática de Montague, realizando algumas extensões.

## 7.2 A noção de aspecto

A categoria aspecto foi pobremente estudada nas duas décadas iniciais da tradição gerativo-transformacional, sendo uma exceção o trabalho de Anderson /AND 73/. Além disso, uma investigação aprofundada na Linguística moderna sobre a noção de aspecto ainda não foi suficientemente feita, existindo contudo a sempre referida revisão de /CDM 76/, a monografia de Friedrich /FRI 74/ e o simpósio editado por Hooper /HOO 82/.

A colocação da noção de aspecto no campo da Linguística ocidental muito deve ao estudo das línguas eslavicas, as quais possuem a distinção entre "Aspekt" e "Aktionsart"; assim, por exemplo no russo, é possível distinguir entre "cital" (lia, pretérito imperfeito), "procital" (li, pretérito perfeito) e "pocital" (lia muito, pretérito perfeito atenuado).

Friedrich /FRI 74/ propõe que o aspecto seja visto como uma categoria linguística básica, distinguindo um aspecto durativo ou continuativo, um aspecto pontual ou perfeitoivo ou completivo e um aspecto estativo. Hooper /HOO 79/ /HOO 79a/, por seu turno, assume a mesma divisão, colocando que o aspecto imperfeitivo está ligado à provisão de informação sobre acontecimentos concorrentes, enquanto que o aspecto perfeitoivo está ligado à provisão de informação sobre eventos discretos e para relacionamento de eventos a tempos de referência da narrativa ou da fala experiencial. Esta distinção, mostra Hooper, é mais facilmente vista no discurso narrativo, quando as cláusulas perfeitivas ligam-se aos eventos principais da história, cada evento sendo discreto e apresentado numa ordem sequencial; em contraste, as cláusulas imperfeitivas assinalam eventos durativos, habituais e em realização. Numa frase: o imperfeitivo é o fundo, o perfeitoivo é a figura.



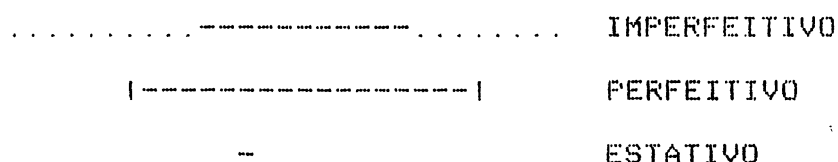
Por seu turno, Li, Thompson e Thompson /LIT 82/ mostram que o aspecto imperfeito ou durativo fornece informação sobre o que está ocorrendo, sobre eventos concorrentes, sobre o que chama a atenção na cena de fundo, sendo que os eventos não são discretos e a ação é incompleta; já o aspecto perfeito envolve apenas eventos discretos e o aspecto estativo envolve o relacionamento de um estado ao tempo de referência da fala.

Givon /GIV 82/ propõe que o espaço semântico referente ao aspecto dos verbos seja visto como o produto da interação de dois traços básicos:

- a limitação: o aspecto durativo constrói um evento como não tendo limites, em oposição ao aspecto pontual;
- o ponto de referência: a localização com respeito a qual a posição dos limites do evento podem ser assinalados.

Com estes traços, Givon propõe a existência de um aspecto durativo (falta de limites) de um lado, e a perfeiuidade (completude) e a imperfeituuidade (incompletude) de outro lado envolvendo apenas o limite terminal do evento em relação ao ponto de referência. Além disso, Givon mostra que a completude de um evento em geral precede o ponto de referência, de modo que existe algum intervalo entre a completação do evento e o ponto de referência.

Um modelo gráfico inicial para o aspecto seria:



Contudo um uso também típico de perfeito no Português é a descrição de uma "sequência de eventos", como em "Pedro comeu duas maçãs", o que foge um pouco ao modelo gráfico acima; outro uso típico está ligado à descrição de "eventos ligados temporalmente", como em "Pedro reinou por 3 anos". De qualquer forma uma característica básica é a finalização do evento descrito, devendo ser observado aí o uso do pretérito perfeito. Em contraste, o imperfeito (manifestado em geral pelo pretérito imperfeito) provê informação sobre um evento habitual ("Pedro costumava ler o jornal"), continuativo ou durativo ("João seguia chorando") ou mesmo iterativo ("Maria tossia durante todo o concerto"), mas em todos os casos com limites incertos.

A origem da flexão de aspecto (perfeito/imperfeito) em Português pode ser encontrada nas formas verbais latinas chamadas de "perfectum" (perfeito, feito cabalmente, concluso) e "infectum" (imperfeito, não feito cabalmente, inconcluso). Esta oposição era expressa morficamente, na medida em que o "perfeito" possuía a marca formal, através de uma marca acrescentada ao radical do verbo, antes mesmo dos constituintes flexionais de tempo, modo e pessoa. Assim existia o seguinte quadro:

	pretérito	presente	futuro
infectum	amabas	amas	amabis
perfectum	amaueras	amauisti	amaueris

Já, contudo, no latim clássico a noção de aspecto concluso entrou em conflito com a noção de presente, e o resultado foi a interpretação de uma forma como "amauisti" como designativa de tempo pretérito /CAH 77/: a razão essencial para isso foi uma mudança na significação do aspecto perfeito, pois que êle inicialmente possuía um carácter "permansivo" (indicava que o evento concluso continuava em seus efeitos). Assim, enquanto "amas" (imperfeito) ficava sem membro opositivo, "amauisti" passava a se opor a "amabas", como pretérito perfeito e imperfeito respectivamente. Uma forma perfeita de pretérito como "amaueras", cedendo seu lugar a "amauisti", recuou no pretérito, indo indicar um pretérito concluso antes de out., ou seja, um pretérito mais que perfeito. O resultado foi a assimetria seguinte:

	amabas	amas	amabis
amaueras	amauisti		amaueris

Assim o sistema latino de indicativo, ponto de partida do sistema português, ficou com um conjunto de quatro grupos de formas em que o jogo das oposições está profundamente alterado. O presente imperfeito perde o membro opositivo, mas continua associado a um carácter imperfeito. O pretérito mais que perfeito conserva o significado aspectual, essencialmente ligado à sua posição de um pretérito anterior a outro, e logo necessariamente concluso, mas fica assimetricamente isolado na zona pretérita em que se colocou. A oposição entre aspecto perfeito e imperfeito mantém-se intacta na primeira zona pretérita, onde se distinguem uma forma como "amabas" (amavas) e outra como "ama(ui)sti" (amaste).

Já em relação ao futuro, êle tem no Português uma coloração de modalidade, como é típico da maioria das linguagens naturais. A característica do sistema que evoluiu para o Português, no que se refere aos elementos aspectuais, é que a combinação do infinitivo imperfeito com o presente do verbo "habere" (haver), em grande parte do mundo romano europeu, estabeleceu-se como uma locução volitiva, focalizando a partir do presente a vontade que uma certa ocorrência se desse. Concomitantemente se estabeleceu uma locução paralela do ponto de vista do passado. Para o futuro do presente ocorreu a forma "amare habeo" (que evoluiu para "amarei") e para o futuro do pretérito a forma "amare habebam" (que evoluiu para "amaria"). Estas formas são formalmente de aspecto não marcado, mas elas possuem um forte aspecto imperfeito.

No que diz respeito ao modo subjuntivo, o Latim compreendia dois tempos imperfeitos e dois tempos perfeitos. Porém a gramática latina considerava o presente perfeito do subjuntivo como um pretérito perfeito e o pretérito perfeito do subjuntivo como um pretérito mais que perfeito. A diferença funcional básica em relação ao subjuntivo, entre o Latim e o Português, é que neste último as formas subjuntivas são próprias das orações subordinadas; elas continuam a exprimir uma ocorrência duvidosa, desejável ou hipotética (característica de um tipo de modalidade), mas no âmbito de uma comunicação dependente de outra. Numa sentença isolada, estão condicionadas à anteposição de um advérbio como "talvez".

No que se refere a associação entre aspecto perfeitoivo e pretérito, ela tornou-se gradualmente mais forte no latim vulgar; o resultado foi a fixação, como pretérito único, do chamado pretérito mais que perfeito ("amauisses") e o abandono gradual do pretérito imperfeito ("amarem"), criando-se no subjuntivo a oposição entre presente imperfeitoivo e pretérito perfeitoivo do tipo "ames"/"ama(ui)sses" no plano temporal. É esta oposição que vigora em Português, onde o presente ("ames") alterna com o pretérito ("amasses"), sendo que, neste sistema bipartido, o presente abarca naturalmente o futuro. Porém foi conservado o reflexo do presente perfeito latino ("amaueris") em certas construções de carácter futuro e é assim denominado nas gramáticas portuguesas. Este futuro alterna-se nesses padrões com o pretérito do subjuntivo, numa oposição temporal de futuro e pretérito. A estruturação geral então do subjuntivo em Português ficou então a seguinte:

presente (ames) ----- pretérito (amasses)  
 |  
 |  
 futuro (amares)

Além destas questões relativas à distinção perfeitoivo/imperfeitoivo no que concerne aos tempos verbais, em primeiro lugar, deve ser observado que a língua portuguesa é muito mais rica em distinções aspectuais, haja visto a existência dos assim denominados tempos verbais compostos (para os modos indicativo e subjuntivo) e das conjunções perifrásticas. O problema do aspecto em relação a estas construções não será tratado aqui, mas na secção sobre construções com auxiliares.

Em segundo lugar, e de um modo muito mais importante, o que foi referido na discussão acima foi o relacionamento da noção de aspecto com certos afixos flexionais e temporais (ou, se se quizer, com esquemas morfo-sintáticos). Esta preocupação (com os marcadores de aspecto) relaciona-se com o estudo dos diferentes modos de mostrar a constituição interna do estado de coisas descrito. Porém, em todas as línguas naturais as diferenças semânticas inerentes aos significados dos verbos determinam que estes verbos possuam diferentes interpretações quando combinados com os marcadores de aspecto. Além disso, é sabido na Linguística /COM 76/ que certos tipos de verbos só podem aceitar certos marcadores de aspecto e certos advérbios de tempo. Com isto então é fácil justificar uma noção mais ampla de aspecto que envolva uma classificação dos verbos, nos moldes das categorias aristotélicas.

Seguindo a linha geral apresentada neste trabalho, esta tipologia dos predicadores é mostrada dentro do campo da Gramática de Montague e a partir da proposta de um cálculo lógico (aspectual) sobre os significados dos verbos, de modo a permitir montar um sistema dedutivo. Isto é o que é apresentado a seguir.

## 7.3 Um cálculo lógico sobre os predicadores

### 7.3.1 A decomposição do significado de um verbo

A análise da decomposição do significado das palavras na Linguística moderna começa com Hjelmslev /HJE 53/ e com Jakobson /JAK 71/; o exemplo padrão de Hjelmslev, baseado somente em considerações paradigmáticas, foi algo como:

mulher	homem	menino
vaca	boi	bezerro
égua	cavalo	potro
galinha	galo	pinto

onde por exemplo as palavras da primeira coluna contrastam com as da terceira sempre do mesmo modo, esta relação sistemática pode ser descrita pelo assinalamento do traço semântico "feminino" a todas as palavras da primeira coluna, o traço "adulto" para as palavras da primeira e segunda colunas, o traço "não adulto" para as palavras da terceira coluna, e traços como "humano", "bovino", "equino", etc às várias filas.

Uma versão mais recente /KAT 72/, dentro desta abordagem estruturalista, assume que este sistema de traços é adequado para representar todos os contrastes semânticos evidenciados numa linguagem natural. Esta proposta de análise componencial do significado assume por exemplo a seguinte análise para "cadeira":

(objeto) (físico) (não-vivo) (artefato) (portável) (algo com pés) (algo com assento) (algo com encosto) (assento para um)

conforme /KAT 72:40/.

O que é feito aqui é simplesmente definir as denotações dos predicados extensionais em termos da intersecção das denotações de outros supostamente mais básicos. Como Creswell /CRE 75/ mostra, a análise de Katz para "cadeira" é, do ponto de vista referencial, similar a dizer que "x é uma cadeira" é analisado como:

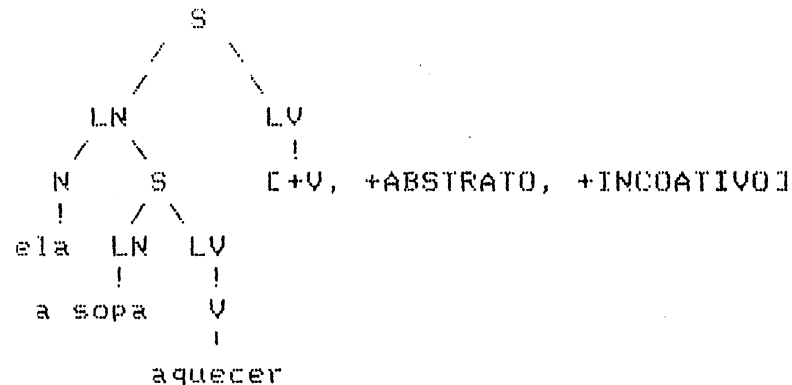
objeto'(x) & físico'(x) & ..... & assento-para-um'(x),  
onde objeto', físico', ..., assento-para-um' são predicados extensionais de primeira ordem de um linguagem artificial. Se for adicionado negação e conjunção a esta linguagem, então ficará criada uma "linguagem de marcação", nos moldes das Gramáticas Gerativas.

Esta análise permite estabelecer implicações sobre palavras "não lógicas" da linguagem natural. Assim por exemplo "Cada solteiro é um homem não casado" pode ser traduzido como:

$(\forall x)[\text{solteiro}(x) \rightarrow [\sim \text{casado}(x) \ \& \ \text{homem}(x)]]$   
e se "solteiro" for decomposto em ( $\sim$ casado), (adulto) e (masculino), e "homem" for decomposto em (adulto) e (masculino), a fórmula lógica resultante é:

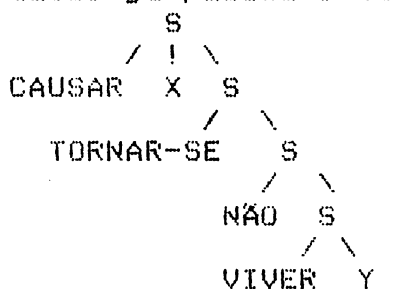
$(\forall x)[[\sim \text{casado}(x) \ \& \ \text{adulto}(x) \ \& \ \text{masculino}(x)] \rightarrow [\sim \text{casado}(x) \ \& \ \text{adulto}(x) \ \& \ \text{masculino}(x)]]$ .

O desenvolvimento deste tipo de análise para verbos começou com Lakoff /LAK 65/. A idéia básica é que estruturas tipo "A sopa tornou-se quente", "A sopa foi aquecida" e "A sopa quente" possuam um significado similar e logo devam vir de uma mesma estrutura profunda. Quanto a isso, Lakoff pressupõe que a última expressão possua um verbo abstrato na sua estrutura profunda com o traço [+incoativo]. A estrutura profunda seria algo como:



e, pela utilização de transformações obrigatórias, ocorreria a substituição do verbo abstrato por verbos reais.

A partir daí, McCawley /McC 68/ /McC 68a/ /McC 71/, Ross /ROS 69/ e Lakoff, começaram a aprofundar mais o nível da estrutura profunda, atribuindo a ela propriedades formais de representação semântica. Assim por exemplo uma estrutura como [x matou y] passou a ter como estrutura profunda algo como:



e sobre esta estrutura agiriam transformações, gerando num dos estágios finais a estrutura:

[[CAUSAR [TORNAR-SE ENÃO VIVER] ] [ X Y ]

Um problema que foi discutido é se a lexicalização das "palavras com várias partes" deveria ocorrer ou diretamente na estrutura profunda acima (como sugere McCawley) ou em estágios tais que, a cada novo estágio de elevação de predicado dentro da árvore, uma transformação substitui o novo predicado complexo (derivado pelo estágio) por uma palavra da linguagem natural; neste último caso, as transformações de lexicalização seriam cíclicas. A primeira abordagem leva a muitas dificuldades para acomodar palavras morfologicamente relacionadas pertencentes a categorias sintáticas diferentes; para fazer isso, ela postulou regras de redundância lexical para dar conta desta regularidades morfológicas /JAC 75/.

A proposta a ser defendida aqui não envolve a pura decomposição do significados dos verbos do Português em uma conjunção de traços, ou numa conjunção de predicados aplicados a uma mesma variável. Ela também não é igual à proposta da Semântica Gerativa exposta acima, pois ela não envolve noções como estrutura profunda e transformações. A idéia é denotar o significado de qualquer verbo do Português através de fórmulas lógicas, compostas fundamentalmente por predicados elementares (definidos numa meta-teoria epistemológica) e variáveis, tendo como campo de trabalho as idéias apresentadas no FTQ de Montague. A finalidade desta formalização é a realização de "cálculos lógicos sobre verbos", de modo a permitir a realização de inferências sobre o significado de e entre sentenças do Português. Com a finalidade de realizar isso, a seguir é exposta uma classificação de verbos do Português e sua respectiva formalização segundo as linhas gerais propostas.

### 7.3.2 Uma tipologia dos predicadores

Ryle /RYL 49/ em "The concept of mind" cunhou os termos 'achievement' para os verbos resultativos e 'activity' para os verbos não resultativos (atividades). Exemplos de resultativos seriam "convencer", "curar", "provar"; as atividades, por outro lado, duram um longo período de tempo, como por exemplo "ouvir uma música". Ryle também distingue uma sub-classe dos resultativos que não possui um desempenho ou tarefa associada com a ação; ele chega a sugerir um conjunto de advérbios para distinguir esta sub-classe dentre os resultativos ordinários; estes resultativos puros são chamados por Ryle de 'performatives'.

Kenny /KEN 63/ em "Action, emotion, and will" acrescentou mais critérios gramaticais e lógicos para esta classificação: assim se x é um verbo de desempenho (resultativo típico), então "w está x-ndo" implica em "w não foi x-do". Porém se x é um verbo de atividade, então "w está (agora) x-ndo" implica em "w foi x-do".

Vendler /VEN 67/, por seu turno, procurou distinguir melhor estas categorias usando restrições sobre advérbios de tempo, tempos verbais e implicações lógicas. Ele distingue assim entre estados, atividades, 'accomplishments' (resultativos com uma tarefa associada à ação, isto é: os 'achievements' puros de Ryle) e 'achievements' (resultativo sem uma tarefa associada à ação). Uma mostra desta classificação é a seguinte:

estados	atividades	accomplishments	achievements
conhecer	correr	pintar um quadro	reconhecer
acreditar	caminhar	fazer uma cadeira	perder
desejar	nadar	falar um discurso	morrer

Além disso, Vendler considera que estados e atividades pertençam a um mesmo polo, e que resultativos com e sem tarefa associada pertençam a um outro polo.

A distinção, trazida do campo da Filosofia, entre estados e atividades aparece na Linguística moderna já na tese de Lakoff /LAK 65/ sob os conceitos de estativo e não estativo. Já Lakoff já apresenta uma série de testes linguísticos para a distinção entre as quatro categorias de verbos; assim:

- só não estativos possuem construções tipo [estar + gerúndio]:

\*Pedro está conhecendo

Pedro está correndo

Pedro está construindo uma casa

- só não estativos são complementos de "persuadir":

\*Pedro persuadiu Paulo a conhecer a resposta

Pedro persuadiu Paulo a correr

Pedro persuadiu Paulo a construir uma casa

- só não estativos possuem imperativos:

\*Conheça a resposta!

Corra!

Construa uma casa!

- quando uma atividade ocorre no tempo presente simples, ela tem uma interpretação habitual como frequentativa, o que nem sempre é verdade para estativos:

Pedro corre

Pedro conhece

- etc.

Matheus et alii /MAT 83/ propõem, para o Português, uma tipologia mais simples que a de /VEN 67/, envolvendo:

- estados, nos quais nenhuma entidade envolvida sofre qualquer alteração durante o intervalo de tempo em que tal estado de coisas ocorre;

- eventos, nos quais existe a alteração de um estado p para um estado q;

- processos, nos quais ocorre uma mudança de estado de uma entidade a partir de um fazer realizado por outra entidade.

A proposta de Matheus et alii /MAT 83/ está baseada num estudo realizado por Bennett e Partee /BEN 78/. O autor deste trabalho considera, contudo, esta classificação como muito simplificadora e propõe em contraposição uma outra que segue a linha do estudo clássico de Vendler /VEN 67/, referido antes. Assim são distinguidos aqui quatro grandes tipos de predicados: os estativos, as atividades, os resultativos sem tarefa associada à ação e os resultativos com tarefa associada à ação.

São exemplos de estativos os predicados compostos por [ser + adjetivo] do tipo "ser alto", "ser verdade", os predicados do tipo "existir", "viver", "amar", "conhecer", "provar", os predicados relacionados à percepção física do tipo "ver", "ouvir" (os quais são também resultativos sem objeto associado à ação), os predicados cognitivos do tipo "compreender", "conhecer", "acreditar" (os quais são também resultativos sem objeto associado durante a execução da ação), os predicados associados a "movimento psíquico" como "desmaiar", os predicados com objeto não extensional do tipo "desejar", "necessitar", os predicados locativos do tipo "estar em", "estar em torno de", os predicados de rio ou estrada como "velejar", "flutuar", "rolar", predicados do tipo "sentar", etc.

São exemplos de atividades os predicados com adjetivos do tipo "estar brabo", "ser rude", "ser polido", os predicados com nomes predicativos do tipo "ser um herói", "ser um bastardo", os predicados intransitivos do tipo "vibrar", "rotar", "correr", os predicados cosmológicos do tipo "chover", os predicados do tipo "falar", "andar", "dancar", "sentar", "escrever", os predicados com apagamento de objeto do tipo "fumar", "comer", "beber", os predicados transitivos de movimento do tipo "dirigir um carro", "empurrar LN", etc.

São exemplos de resultativos sem tarefa associada à ação os predicados com mudança de estado físico do tipo "explodir", "nascer", "aquecer", os predicados com complemento "aspectual" do tipo "começar a", "cessar de", "parar de", os predicados possessivos do tipo "adquirir um livro", "receber", "perder", os predicados cognitivos "ouvir", "cheirar", "reconhecer", "detectar", "esquecer" (estes cognitivos podem ser também estativos), etc.

São exemplos de resultativos com tarefa associado à ação os predicados locativos do tipo "voar para", "colocar em", os predicados de criação do tipo "desenhar", "fazer uma casa", os predicados de destruição do tipo "destruir", "apagar", os predicados de mudança de estado do tipo "matar", "petrificar LN", os predicados de criação de objeto "fotografar um LN", "pintar um unicórnio", "compor uma sonata", "ouvir uma sinfonia", etc.

### 7.3.3 A lógica dos predicadores

Um exemplo para início de discussão seria a tradução de sentenças contendo um verbo estativo e uma locução adverbial de tempo. Assim a sentença "João ama Maria por tres anos" poderia ter como tradução em lógica uma fórmula como:

(Et :t pertence a 3 anos) EM (t, João ama Maria)

na qual o predicado EM(a, b) denota que o que é denotado por b ocorre no tempo a. O problema para traduzir esta sentença nos moldes de uma proposta como a do PTQ, é que no PTQ não existem constantes nem variáveis, nem expressões que denotem diretamente o tempo. Exemplos de linguagens interpretadas temporalmente com expressões denotando tempo são a lógica das séries B de Prior e o cálculo R de Rescher e Urquhart.

Outro caso problemático para o sistema de índices do PTQ é representado pelos eventos, os quais não são verdadeiros nem falsos por um período de tempo, nem ocorrem num certo ponto do tempo: pois eventos simplesmente ocorrem no tempo.

Uma primeira proposta bastante simples para a solução destes casos, mantendo o "espírito das idéias de Montague", seria a utilização de uma lógica que trabalhasse com mudanças de estado, como a de Von Wright /VON 63/. Nela um evento é uma mudança do estado p para o estado q, onde  $p = \sim q$ .

Estabelecendo as seguintes leituras para o operador diádico T (lido como "e depois") como:

$\sim p$  T p : o estado  $\sim p$  é destruído  
 p T  $\sim p$  : o estado p é destruído  
 p T p : o estado p permanece



$\sim p \text{ T } \sim p$  : o estado  $\sim p$  permanece  
é possível apresentar definições semânticas para alguns verbos  
como:

Tornar-se (p) =  $\sim p \text{ T } p$   
Acabar (p) =  $p \text{ T } \sim p$   
Permanecer (p) =  $p \text{ T } p$

Além disso, se o tempo for considerado como discreto, se  
x é uma fórmula e se t é um tempo, então:

TORNAR-SE x é verdade em t sse x é verdade em t e falso em  
t-1

ACABAR x é verdade em t sse x é falso em t e verdade em t-1

PERMANECER x é verdade em t sse x é verdade em t-1.

Deve ser observado que aqui está sendo assumido que uma  
sentença com verbo de mudança de estado de coisas implica  
necessariamente na existência deste estado de coisas antes; o  
mesmo vale para sentenças que negam um estado de coisas como  
"João parou de bater em sua esposa" e "A porta não abriu às oito  
horas". Isto está de acordo com as pressuposições feitas pelo  
ouvinte destas sentenças numa conversação comum.

Com estes predicados definidos como acima, é possível  
então propor como tradução geral para todos os predicados  
resultativos sem tarefa associada (achievements) a seguinte  
fórmula:

TORNAR-SE : oração encaixada

lembrando aí que muitos destes predicados são compostos por  
verbos incoativos derivados de adjetivos e por verbos com  
complemento aspectual

Porém existem verbos incoativos que podem ocorrer com  
advérbios durativos como em "A sopa aqueceu em dez minutos" e em  
"João envelheceu quarenta anos durante a experiência"; estes  
verbos são ditos "predicados vagos" /LEW 70/. O problema em  
relação a eles é a dificuldade do estabelecimento de uma extensão  
definida para eles. Uma solução seria a utilização de um grau  
numérico de verdade entre verdadeiro e falso /KAM 75/, aplicável  
a exemplos como "O arroz está mais quente que a sopa" e "João é  
mais alto que Paulo"

Uma proposta inicial para os verbos resultativos com  
tarefa associada à ação (accomplishments) seria

[X CAUSAR Y]

onde X e Y são sentenças de algum tipo, levando em conta que  
estes predicados estão associados a alguma idéia de causação. É  
comum aqui que X seja uma sentença tipo TORNAR-SE ou do tipo que  
envolva o controle sobre alguma atividade e Y seja uma sentença  
tipo TORNAR-SE. Esta proposta permitiria a seguinte análise para  
"João matou Pedro"

[João fez algo] CAUSAR [TORNAR-SE "[Pedro ser vivo]]

e a seguinte análise para "João pintou um quadro"

[João pintou] CAUSAR [TORNAR-SE [um quadro existir]]

as quais estariam de acordo com as formalizações de McCawley  
/McC 71/ e Givon /GIV 72/.

Em relação ao predicado CAUSAR, é possível distinguir dois tipos de causação nas linguagens naturais: a causação direta ou manipulativa e a causação indireta ou diretiva /SHI 76/. A manipulativa necessariamente envolve uma manipulação física do objeto afetado pelo agente; a diretiva não. Assim "João derrubou uma criança" é manipulativa e "João fez com que a criança caísse" é diretiva. Como observação deve ser notado que a causação diretiva tende a ocorrer principalmente em construções perifrásticas. Esta subclassificação do predicado CAUSAR não é aqui todavia explorada.

Uma aplicação refinada deste tipo de proposta seria sua utilização para a análise das crenças envolvidas com a enunciação de sentenças como "João abriu a porta". Ela envolve o seguinte conjunto de crenças:

- a porta não estava aberta antes de João abrir (já comentado antes);
- a porta tornou-se aberta logo após João agir;
- a porta não tornar-se-ia aberta nesta ocasião se João não agisse e se todo o resto tivesse permanecido o mesmo.

As primeiras duas condições determinam que o evento de abertura da porta realmente ocorre; isto poderia ser mostrado pela fórmula [X CAUSAR [TORNAR-SE Y]], a qual implica a verdade da fórmula [TORNAR-SE Y]. A última condição é o elemento contrafactual na causação.

Outro predicado a ser analisado é FAZER. Em primeiro lugar o significado de FAZER nem sempre denota uma ação, devido à existência de exemplos como "João está quieto" e "João está ignorando Maria", o que torna discutível sua utilização para a distinção entre estativos e atividades. Uma proposta inicial aqui seria procurar estabelecer uma subcategorização entre os predicados estativos: uma subcategoria seria formada pelos predicados estativos que tornam-se atividades quando combinados com FAZER e a outra pelos que não se tornam atividades; isto mostraria a diferença entre "procurar a resposta" e "ser polido" que são atividades, e "estar no Rio" e "conhecer a resposta" que não são.

Uma outra discussão envolvendo FAZER é que seu significado às vezes está ligado com alguma coisa relacionada à causação volicional, de modo que isto pode ser útil numa proposta de subcategorização dos verbos que denotam atividades. Um refinamento disto seria a utilização da noção "estar sob o controle imediato do agente". Assim FAZER (a, b), onde a é um termo individual e b é uma sentença, ficaria interpretado como:

FAZER (a,b) (-) b & estar-sob-controle-imediato-do-agente (b) ficando em aberto a formalização da segunda parte da definição.

Sob esta discussão deve ficar clara a separação quer dos estativos e quer das atividades numa forma simples e numa forma na qual alguma idéia de causação está envolvida.

Considerando o acima exposto, as evidências para FAZER ser considerado predicado semântico são bem menores do que as evidências para CAUSAR e TORNAR-SE, pois os dois últimos podem ser vistos como operadores semânticos que envolvem mundos possíveis e FAZER apenas mapeia um predicado extensional em outro predicado extensional.

Mesmo aceitando estas considerações, é possível estabelecer a seguinte definição:

se X e Y são fórmulas, a é um termo e t é um termo denotando um tempo, então TORNAR-SE (X), [X CAUSAR Y], FAZER (a, X) e EM (t, X) são fórmulas

onde FAZER não modifica predicado. Por outro lado, as conceituações semânticas dos verbos não envolvem EM, porque as condições de verdade dos verbos nunca dependem de estados ou mudanças de estado que ocorram num tempo específico, independente do tempo de uso do verbo, mas sim possuem sua referência temporal determinada indexalmente. Como comentário deve ser notado que as locuções adverbiais podem fixar a referência de tempo eterno (como "durante o verão") e de um modo indexado (usando por exemplo "amanhã"), e assim requerem o operador EM.

Com esta maquinaria é possível propor definições para estativos, atividades, resultativos com tarefa e resultativos sem tarefa associada à ação, de modo a abrir a possibilidade de um cálculo de predicados que opere sobre a semântica dos verbos do Português.

As definições que se seguem devem ser encaradas como tentativas. A idéia é a utilização dos predicados estativos como predicados primitivos; assim, no abaixo exposto, são utilizados p e q para a denotação destes predicados estativos elementares e a1, a2, ..., an, b1, b2, ..., bn para a denotação dos argumentos destes predicados estativos elementares. A natureza e o número destes argumentos são explicitados segundo o predicado específico em questão.

Considerando a lista de predicados estativos dada acima é possível propor como expressão de tradução a fórmula geral:

p (a1, a2, ..., an)

a qual vale por exemplo para "João conhece a resposta".

Em relação as atividades, a tradução envolve o predicado FAZER. Podem ser distinguidas as atividades simples, como em "João está caminhando", traduzidas por

FAZER (a1, p (a1, ..., an))

e as atividades que envolvem causação e agentividade, como em "Ele está vendendo seu carro no banco", as quais teriam como fórmula de tradução

[FAZER (a1, p (a1, ..., a2))] CAUSAR q (b1, ..., bn)..

No que diz respeito aos resultativos sem tarefa associada durante a ação, uma tradução para os predicados simples é

TORNAR-SE (p (a1, ..., an))

a qual é aplicável a sentenças do tipo "João descobriu a solução". No caso de predicadores que estabelecem incoação de atividade, como em "João começa a andar", uma tradução é

TORNAR-SE [FAZER (a1, p(a1, ..., an))]

mas no caso de incoação de resultativo com tarefa associada à ação, como em "João começou a construir a casa", a tradução é simplesmente:

TORNAR-SE X.

No que se refere aos resultativos com tarefa associada à ação, para o caso dos predicados não agentivos ("A abertura da porta fez com que a vela apagassem"), uma tradução é

[TORNAR-SE X] CAUSAR [TORNAR-SE Y]

onde X e Y são sentenças estativas. Para o caso de predicados agentivos não intencionais ("João quebrou o vidro"), a tradução é simplesmente:

[FAZER (a<sub>1</sub>, p(a<sub>1</sub>, ..., a<sub>n</sub>))] CAUSAR [TORNAR-SE(b<sub>1</sub>, q(b<sub>1</sub>, ..., b<sub>n</sub>))]

e no caso de agentivo com agente secundário (João forçou Pedro a falar"), a tradução é

[FAZER (a<sub>1</sub>, p(a<sub>1</sub>, ..., a<sub>n</sub>))] CAUSAR [FAZER (b<sub>1</sub>, q(b<sub>1</sub>, ..., b<sub>n</sub>))]

e para o caso de agentivo intencional ("João assaltou Pedro") a tradução é:

[FAZER (a<sub>1</sub>, p(a<sub>1</sub>, ..., a<sub>n</sub>))] CAUSAR X

onde X é uma sentença não estativa.

Nesta proposta de classificação, as atividades e os resultativos com e sem tarefas associadas à ação ficam definidos em função dos estativos primitivos e de operadores (predicados) de natureza semântica. A intensão destes estativos primitivos pode ser vista como uma função do domínio das entidades De em (0, 1), segundo M, w, g, t (referidos e discutidos no capítulo sobre a Gramática de Montague); desta forma, o significado destes predicados estativos aponta para propriedades físicas de algum tipo (espaço, tamanho, cor, peso, etc). Outros estativos contudo não possuem uma intensão tão direta, como "conhece", "estar bêbado", "ser inteligente", etc, os dois últimos sendo o que Carlson /CAR 77/ chama de "predicados de estágios de indivíduos".

Assumindo o exposto imediatamente acima, é possível propor alguns postulados para cada uma destas categorias de predicados, usando aqui algumas idéias que estão em Taylor /TAY 77/:

- se p é um predicado estativo, p(x) é verdade no intervalo I e se p(x) é verdade em todos os momentos de I;
- se p é um predicado de atividade ou resultativo, então p(x) só é verdade num intervalo maior do que um momento;
- se p é um predicado resultativo e se p(x) é verdade no intervalo I, então p(x) é falso em todo subintervalo de I;
- se p é um verbo de atividade e se p(x) é verdade no intervalo I, então p(x) é verdade para todos os subintervalos de I maiores do que um momento.

A primeira regra vale tanto para os estativos momentâneos ("estar na mesa") como para os estativos de estágio com objeto ("conhecer", "ser inteligente"), e também é aplicável aos estativos de intervalo ("sentar"). A segunda regra tem como consequência que, se o tempo de uma elocução for considerado como um momento, então é impossível dizer uma sentença com um verbo não estativo denotando um instante; isto parece concordar em parte com a frequente interpretação frequentativa dada para sentenças no presente simples destes verbos, mas a regra não tem

contudo um caráter absoluto. Já a terceira regra segue naturalmente da análise destes verbos com TORNAR-SE sob a condição do mínimo intervalo.

#### 7.3.4 O papel do cálculo dos predicadores na gramática

Dentro do campo da Gramática de Montague, a proposta de cálculo aspectual sobre os predicadores envolve estabelecer relações entre resultativos, atividades e estativos, através da noção de decomposição lexical. Desta forma, a definição de uma linguagem natural (como o Português) passa a envolver dois componentes:

- a definição montaguiana da linguagem L, com sua sintaxe e sua interpretação, e
- a definição formal de um componente lexical P.

Este componente lexical fica, segundo a proposta aqui apresentada, definido formalmente como independente de L, porém ele possui certas partes em comum com L, pois:

- o conjunto dos nomes das categorias sintáticas de P deve ser o mesmo conjunto dos nomes das categorias básicas de L;
- o conjunto das expressões básicas de cada categoria em P deve ser o mesmo conjunto das expressões básicas de cada categoria em L;
- a interpretação de cada expressão básica de L deve ser a mesma interpretação de cada expressão básica de P.

Usando as definições apresentadas para estativos, atividades e resultativos em função de predicados primitivos, tais como FAZER, TORNAR-SE, CAUSAR, EM e outros predicados estativos primitivos, é possível contruir um sistema dedutivo que pode ser aqui apresentado na forma quer de um cálculo lambda, quer na forma de uma lógica de predicados. A seguir são apresentados fórmulas que permitem passar de um tipo de notação para o outro.

Assim no caso dos verbos estativos, sendo "a" um membro designado de IV (locuções verbais intransitivas), vale a idéia de que necessariamente este estado existe num certo tempo; ou mais formalmente:

$$(Vx) \text{ nec } [ a(x) \leftrightarrow (Et) EM(t, a(x)) ]$$

e para o caso de "a" ser um membro de TV (Locuções verbais transitivas) vale:

$$(Vx) (V P) [ a(x, P) \leftrightarrow (Et) EM(t, a(x, P)) ]$$

Por outro lado, para os verbos incoativos de mudança simples (traduzidos com um único TORNAR-SE) vale:

$$(E P) (Vx) \text{ nec } [ a(x) \leftrightarrow \text{TORNAR-SE } P(x) ]$$

onde "a" é um membro designado de IV; no caso de "a" ser um membro de TV vale

$$(E P1) (E Q) (Vx) (VP) \text{ nec } [ a(x, P) \leftrightarrow P(\text{int } y [P1(x) \text{ CAUSAR TORNAR-SE } Q(y) ]) ]$$

Para os verbos de atividade, pode ser colocada a idéia de que algo é verdadeiro num certo intervalo e falso logo após este intervalo; de um modo mais formal, onde "a" é um IV, vale:

$$(E Q) (Vx) \text{ nec } [ a(x) \rightarrow [Q(x) \text{ AND } \sim Q(x)] ]$$

onde  $[K \text{ AND } L]$  é lido como "K é verdadeiro no intervalo  $t$  e L é verdadeiro no intervalo  $t_1$ , onde  $t_1$  é disjuncto e posterior a  $t$ "; por outro lado, para "a" pertencente a TV vale

$(E Q) (Vx) (V P) \text{ nec } [a(x,P) \rightarrow [Q(x) \text{ AND } \sim Q(x)]]$

Para os resultativos com tarefa associada e complexos (com um ou mais operadores TORNAR-SE), o discutido antes girava em torno da idéia de que alguma coisa causava que certa coisa tornasse-se em outra (X CAUSAR [TORNAR-SE Y]); traduzindo em cálculo lambda, para "a" sendo um TV vale

$(E Q_1) (E Q_2) \dots (E Q_n) (Vx) \text{ nec } [a(x, \text{int } P \text{ P}(y) \langle \rightarrow \rangle [P(x) \text{ CAUSAR TORNAR-SE } [Q_1(y) \text{ AND } Q_2(y) \text{ AND } \dots \text{ AND } Q_n(y)]]]$ .

Por outro lado, para a maioria dos verbos que possuem uma tradução complexa vale:

$(E R) (V P) (Vx) \text{ nec } [a(x, P) \langle \rightarrow \rangle P (\text{int } y \text{ ext } R (x,y))]$ ,

onde a é um membro designado de TV.

Pegando um exemplo mais concreto para decomposição na tradução, o verbo intransitivo "morrer" (resultativo sem tarefa associada) pode ser traduzido como:

morrer  $\Rightarrow$  lambda x [TORNAR-SE "vivo" (x)]

e, se o cálculo aspectual defendido acima for mais profundamente instituído, a forma "vivo" pode ser substituída pelo predicado estativo (constante, na lógica) VIVO, o qual pertence a mesma categoria por exemplo de TORNAR-SE. Assim vale:

morrer  $\Rightarrow$  lambda x [TORNAR-SE "VIVO (x)].

No caso da sentença "João morreu", a tradução segundo o FTQ (desconsiderando-se o tempo verbal e o aspecto) seria:

morrer' (j)

e segundo a proposta do cálculo aspectual sobre predicadores seria:

lambda P [ P(j) ] (int x [TORNAR-SE "VIVO (x) ] )

que é idêntica a:

TORNAR-SE "VIVO (j).

Deve ser verificado que este tipo de representação permite responder logicamente a perguntas do tipo "No que João se tornou?" e "João está vivo?", levando a uma ampliação da "inteligência" do sistema.

Já um verbo tipo "matar" teria como tradução:

matar  $\Rightarrow$  lambda P lambda x P ( int y [ (E P1) [ P1(x) CAUSAR TORNAR-SE "VIVO (y) ] ] )

e uma construção como [matar Carlos] teria como tradução:

lambda P lambda x P ( int y [ (E P1) [ P1(x) CAUSAR TORNAR-SE "VIVO(y) ] ] ) (int P P(c))

a qual, por conversão lambda (substituição de P) e cancelamento entre int e ext, é equivalente a:

lambda x [ lambda P1 P1(c) ( int y [ (E P1) [ P1(x) CAUSAR TORNAR-SE "VIVO(y) ] ] ) ]

a qual, por conversão lambda sobre P1 e cancelamento entre int e ext, é equivalente a:

lambda x [ lambda y [ (E P1) [ P1(x) CAUSAR TORNAR-SE "VIVO(y) ] ] ] (c)

a qual, por conversão lambda sobre y, é equivalente a:

lambda x [ (E P1) [ P1(x) CAUSAR TORNAR-SE "VIVO(c) ] ].

Assim "João mata Carlos", não considerando o tempo verbal, tem como tradução:

$\lambda$  P P(j) (int x [(E P) [P(x) CAUSAR TORNAR-SE "VIVO(c)"]])

que é equivalente a:

(E P) [P(j) CAUSAR TORNAR-SE "VIVO(c)"]

que diz que João fez algo (êle tem uma propriedade P) e isto causa que Carlos torna-se não vivo. Esta decomposição permitiria responder perguntas do tipo "O que João fez?", "O que João causou?", "No que Carlos se tornou?", "Carlos está vivo?", "Quem causou a morte de Carlos?".

Em relação aos outros verbos resultativos com tarefa associada, a decomposição é do mesmo tipo, diferindo só no predicado estativo que ocorre na posição mais interna e no tipo de predicado de atividade que substitui a variável de propriedade P.

A idéia da decomposição pode ser aplicada também a outros causativos. Assim a sentença "João abriu a porta" pode ser pensada como traduzida informalmente por

(João V) CAUSAR (TORNAR-SE (porta aberta))

devendo ser observado aí que TORNAR-SE é um operador monádico de sentença e CAUSAR é um conetivo diádico de sentenças.

Semanticamente, TORNAR-SE é um tipo especial de operador temporal que diz que uma fórmula é agora verdadeira, mas que ela era falsa no momento anterior. Por outro lado, CAUSAR pode ser interpretado como segue:

[X CAUSAR Y] é verdade em algum mundo w sse X e Y são verdadeiros em w e também num mundo possível muito similar a w, exceto que, quando "X é verdadeiro, "Y é também verdadeiro.

Isto colocado como adição ao PTQ teria a forma:

- se X e Y são sentenças bem formadas, então [X CAUSAR Y] e [TORNAR-SE X] são sentenças bem formadas;

- existe uma função de seleção f que assinala a cada fórmula bem formada X e a cada w pertencente a W um membro f(X, w) de W;  
obs.: informalmente, f(x, w) é w<sub>i</sub> mais do que w, com a possível exceção de que X seja verdade /STA 70/;

- se X é uma fórmula bem formada, [TORNAR-SE X]<sup>i</sup> é 1 sse X<sup>i</sup> é 1 e X<sup>+</sup> é  $\emptyset$  para algum i<sub>1</sub> < i e para todo i<sub>1</sub> < i<sub>2</sub> < i;

obs.: X<sup>i</sup> é a interpretação de X em relação a M, w, i, g, e X<sup>+</sup> é a interpretação de X em relação a M, w, i<sub>1</sub>, g;

- se X e Y são fórmulas bem formadas, então [X CAUSAR Y]<sup>i</sup> é 1 sse [X & Y]<sup>i</sup> é 1 e [~Y]<sup>i\*</sup> é 1, onde a\* é a interpretação de a em relação a M, f(~X, w), i, g.

Com isto (cálculo aspectual de predicados), passam a existir então dois métodos, dentro do "espírito do PTQ", para o tratamento de verbos como "matar" e "abrir":

- à tradução complexa do verbo pode ser assinalada uma regra de tradução (nos moldes do que Montague fez com a palavra "necessariamente"), ou

- o verbo pode receber uma tradução padrão numa constante (como matar', abrir', etc) e existiriam "postulados de significado" que seriam adicionados de modo a estabelecer relações entre estas constantes e outras fórmulas (nos moldes do

que Montague fez com a tradução de "procurar").

Além disso existe a possibilidade de que o significado mais complexo seja produzido já no processamento de uma sentença declarativa; outra possibilidade é não usar esta tradução nesta fórmula complexa durante o processamento da sentença declarativa e deixar este caminhar só no processamento de perguntas que não possuem solução no processamento das representações simples. Desconsiderando aqui o segundo problema e voltando aos dois métodos de tradução (via regra "necessariamente" e via postulados de significado), segundo o primeiro método, o verbo transitivo "abrir" teria como tradução:

$$\text{abrir} \Rightarrow \lambda P \lambda x P(\text{int } y [(E M) [M(\text{ext } x) \text{ CAUSAR} \\ \text{ETORNAR-SE [aberto'}(y)]]]])$$

na qual M é uma variável do tipo  $\langle s, \langle e, t \rangle \rangle$  e aberto' é do tipo  $\langle \langle s, e \rangle, t \rangle$  e é entendido como representando a tradução do adjetivo predicativo "aberto".

Da mesma forma, "matar" poderia receber a seguinte tradução:

$$\text{matar} \Rightarrow \lambda P \lambda x P(\text{int } y [(E M) [M(\text{ext } x) \text{ CAUSAR} \\ \text{ETORNAR-SE [vivo'}(y)]]]])$$

onde vivo' representa a tradução de "vivo" e é do tipo  $\langle \langle s, e \rangle, t \rangle$ .

Segundo estas traduções, a tradução de "João abriu a porta" seria:

$$(E y)[(V x) [porta'(x) \langle - \rangle [x = y]] \& (E M)[M(j) \text{ CAUSAR} \\ \text{ETORNAR-SE [aberta'}(y)]]]]$$

e a tradução de "João matou Pedro" seria:

$$(E M)[M(j) \text{ CAUSAR ETORNAR-SE [vivo'}(\text{int } p)]]],$$

devendo ser observado aí que a quantificação existencial sobre a variável M de propriedades individuais é empregada porque os significados dos verbos em questão não especificam o tipo de atividade que produz o resultado: uma porta pode ser aberta por um encontro, por uma pancada, por pressionar um controle remoto, etc. O mesmo se aplica para "matar", mas isto não é sempre verdade para todos verbos resultativos com tarefa associada. Assim, por exemplo, as atividades que geram um quadro pintado são especificadas em duas sentenças resultativas com tarefa associada tais como "João desenhou um quadro" e "João pintou um quadro".

Em relação ao exposto, poderia ser objetado que nem toda propriedade de indivíduos é uma propriedade onde está envolvido algum evento, porque somente eventos podem ser causas. Se isto for assim, é necessário modificar as traduções e substituir M por uma propriedade abstrata que seja adequadamente restrita de modo a só referir-se a um evento.

Segundo o segundo método, as traduções de "matar" e de "abrir" são colocadas como constantes simples da lógica intensional e são acrescentados "postulados de significado" como:

$$\text{nec [matar'}(x, P) \langle - \rangle P(\text{int } y [(E M) [M(\text{ext } x) \text{ CAUSAR} \\ \text{ETORNAR-SE [vivo'}(y)]]]])]$$
$$\text{nec [abrir'}(x, P) \langle - \rangle P(\text{int } y [(E M) [M(\text{ext } x) \text{ CAUSAR} \\ \text{ETORNAR-SE [aberto'}(y)]]]])].$$

Uma possibilidade que o postulado de significação oferece é o enfraquecimento da bicondicional para uma condicional, como por exemplo:

$$\text{nec [matar'}(x, P) \rightarrow P(\text{int } y) [(E M) [M(\text{ext } x) \text{ CAUSAR}$$



[TORNAR-SE [~ vivo' (y)]] ] ] }].

Isto pode ser considerado interessante se for aceito que "matar" sempre implica em "causar tornar-se não vivo", mas que a inversa nem sempre valha.

### 7.3.5 Uma extensão para locuções preposicionais

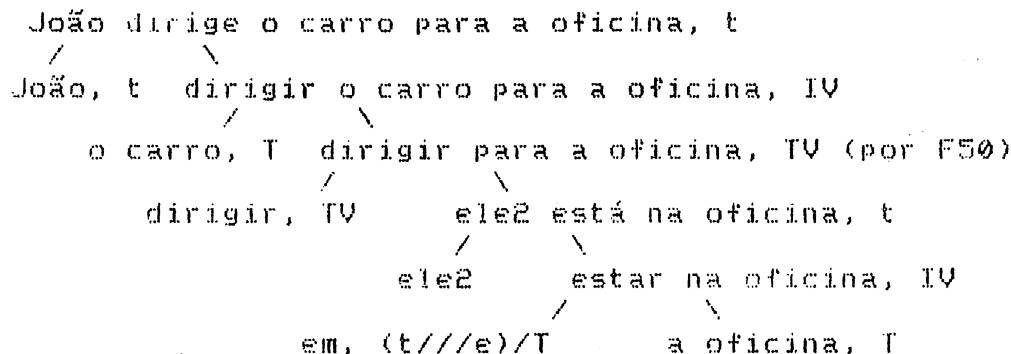
Um outro campo de aplicação do cálculo aspectual sobre predicados (na forma de decomposição lexical) é sobre as locuções preposicionais que expressam alvo. A aplicação aqui é contudo problemática.

Em relação a isso, em primeiro lugar, uma locução verbal que não é do tipo resultativo com tarefa associada pode ser transformada num resultativo com tarefa associada através do emprêgo de certas locuções preposicionais. Assim a sentença "João dirige o carro" não tem um verbo resultativo com tarefa associada, mas "João dirige o carro para a oficina" tem. Esta colocação obriga então que a gramática gere sentenças da forma [elen está na oficina], onde "elen" é uma variável do tipo dos termos nos moldes do PTQ e a locução [está na oficina] é um IV. Considerando que as preposições do tipo "em" seriam asinaladas a uma categoria do tipo (t///e)/T, é possível estabelecer a seguinte regra sintática:

se a é um TV, b é um t e c tem a forma [elen é d], então  $F50(a,b)$  é um TV, onde  $F50(a,b) = ad$ .

Exceção: se d tem a forma [em k], então  $F50(a,b) = a \text{ em } k$ .

Assim a sentença "João dirige o carro para a oficina" teria como árvore de análise:



sendo que a tradução disso implicaria que João dirige o carro e que o carro vai para a oficina como resultado de João dirigir o carro, o que parece correto.

Por outro lado, uma sentença como "João dirigiu o carro de Bagé para Ijuí" teria como árvore de análise:

João dirigiu o carro de Bagé para Ijuí, t, 4  
 / \  
 João, T dirigir o carro de Bagé para Ijuí, IV, 5  
 / \  
 dirigir de Bagé para Ijuí, IV, 7 o carro, T, 2  
 / \  
 de Bagé para Ijuí, TV/TV, 6 dirigir, TV  
 / \  
 de Bagé, (TV/TV)/(TV/TV), 5 para Ijuí, TV/TV, 5  
 / \  
 de, (TV/TV)/(TV/TV)/T Bagé, T para, (TV/TV)/T Ijuí, T

Assim fazendo a tradução de "de" é:

de => lambda P lambda G lambda R lambda Q lambda F (int y Q  
 (int z [ext G (R) (int Pi Pi(z)) (x) AND [ext R  
 (int P P(z)) (x) CAUSAR TORNAR-SE "estar-  
 em'(z,y) ]])

onde G é uma variável do tipo <s, f(TV/TV)>; a tradução correspondente a sentença seria então:

(E x) [carro'(x) AND [dirigir'(j, x) CAUSAR TORNAR-SE  
 "estar-em'(x, i) AND [dirigir'(j, x) CAUSAR  
 TORNAR-SE estar-em'(x, ii) ] ] ].

Uma outra possível proposta é que uma sentença com um verbo de movimento como "João foi caminhando para Canoas" possua a seguinte árvore de análise:

João foi caminhando para Canoas, t, 4  
 / \  
 João, T foi caminhando para Canoas, IV, 7  
 / \  
 para Canoas, IV/IV, 5 caminhar, IV  
 / \  
 para, IAV/T Canoas, T

sendo que neste caso a preposição "para", que é um IAV (ou seja, (IV/IV)/T) possui como tradução:

para => lambda Pi lambda F lambda x Pi(int y [P(x) AND  
 TORNAR-SE [estar-em'(x,y) ] ]])

onde estar-em' é uma constante do tipo <e, <e,t>>.

A tradução da sentença acima seria algo como:

caminhar'(j) AND TORNAR-SE [estar-em'(j, c) ] ].

Já uma sentença como "João foi caminhando de Gravataí para Canoas" teria como árvore de análise:

João foi caminhando de Gravataí para Canoas, t, 4  
 / \  
 João, T caminhar de Gravataí para Canoas, IV, 7  
 / \  
 caminhar de Gravataí, IV, 7 para Canoas, IV/IV, 5  
 / \  
 caminhar, IV de Gravataí, IV/IV, 5 para, IAV/T Canoas, T  
 / \  
 de, IAV/T Gravataí, T

Supondo a tradução das preposições como abaixo:

de =>  $\lambda P_1 \lambda P \lambda x P_1 (\text{int } y [ F(x) \text{ AND TORNAR-SE "estar-em" (x,y) AND (Ez) [TORNAR-SE estar-em'(x,z) ] ] )$

para =>  $\lambda P_1 \lambda P \lambda x P_1 (\text{int } y [ P(x) \text{ AND (Ez)[TORNAR-SE "estar-em" (x,z) AND TORNAR-SE estar-em'(x,y) ] ] )$

a tradução de "João foi caminhando para Canoas" seria algo como:  
[caminhar'(j) AND (Ez)[TORNAR-SE "estar-em'(j,z) AND TORNAR-SE estar-em'(j, c)]]

e "João veio caminhando de Gravataí" teria como tradução:  
[caminhar'(j) AND TORNAR-SE "estar-em'(j, g) AND (Ez) [TORNAR-SE "estar-em'(j, z)]]

A tradução de "João foi caminhando de Gravataí para Canoas" seria então:

[caminhar'(j) AND TORNAR-SE "estar-em'(j, g) AND (Ez) [TORNAR-SE estar-em'(j, z) AND (Ez) [TORNAR-SE "estar-em'(j, z) AND TORNAR-SE estar-em'(j,c)]]

a qual é equivalente a

[caminhar'(j) AND TORNAR-SE "estar-em'(j, .g) AND TORNAR-SE estar-em'(j, c)]]

Deve ser observado que a primeira proposta assumiu que [de locução nominal] modifica [para locução nominal], constituindo um só constituinte /ZWI 78/; neste caso, as traduções das preposições ficaram um pouco diferentes das utilizadas na segunda proposta. A opção pela primeira ou pela segunda proposta não parece todavia clara.

Como comentário final, dentro do campo do cálculo aspectual, deve ser observado que a idéia de decomposição lexical de verbos está intuitivamente ligada a um processo de dividir o significado de uma palavra (um verbo) numa configuração (lógica) finita de blocos semânticos primitivos. Estes blocos (predicados atômicos estativos ou operadores como TORNAR-SE e CAUSAR) poderiam ser pensados como universais cognitivos, entretanto não é claro no momento para o autor deste trabalho se todo verbo do Português pode sofrer uma decomposição exaustiva em expressões bem formadas através de algum conjunto pequeno de primitivos semânticos. De qualquer forma, a proposta aqui feita serve para capturar uma classe importante e enorme de relações entre sentenças, a qual pode ser formalizável e implementável num processador para o Português. Além disso, e de um modo singelo, a estratégia aqui defendida para o tratamento da semântica dos predicadores não foi a de trabalhar sobre o "significado total" de uma palavra (verbo), mas sim a de estabelecer implicações entre pares de sentenças a partir de classes de verbos. Crê o autor que esta estratégia é válida para um processador de uma linguagem natural na medida em que trabalha com generalizações e sobre relações sobre sentenças.

## 7.4 TRATAMENTO COM LÓGICA DO TEMPO VERBAL

### 7.4.1 A interação dos tempos verbais principais com os advérbios de tempo: o modelo E-E melhorado

A proposta aqui é a discussão do modelo do PTQ que utiliza os operadores priorianos F e P (ou G e H), para a especificação da interação entre os tempos verbais principais (presente simples, pretérito simples e futuro simples) com advérbios de tempo.

Em primeiro lugar é assumido aqui que os tempos verbais do Português são parasitas primários dos advérbios e não podem ser compreendidos corretamente sem a compreensão de sua interação com advérbios de tempo. Por isso aqui eles são tratados conjuntamente.

Assim "João chegou ontem" não pode ser tratado como o resultado da composição de duas regras separadas: a regra para tempo verbal e a regra para advérbio. Isto é assim porque se o tempo ficar sob o escopo do advérbio de tempo ou vice-versa, o arranjo total é um resultado incorreto. Isto é: se  $[Contem' x]$  for interpretado como verdade no tempo  $t$  sse  $x$  é verdade no dia anterior a  $t$ , e se  $Px$  for interpretado como no PTQ (onde aparece trocado por H), então

$ontem' [P [chegar' (j)]]$

não representa o significado de "João chegou ontem", porque a fórmula acima diz que chegar' (j) é verdade (agora) se existe um tempo  $t_1$  no dia anterior ao de hoje tal que exista um outro tempo  $t_2$  ainda anterior a  $t$  no qual "João chegou é verdade". Isto é: a fórmula acima é verdade se João chegou dois dias antes, ou um mês, ou um ano atrás. Por outro lado

$P [Contem' [chegar' (j)]]$

sofre dos mesmos defeitos, porque a sentença "João saiu ontem" afirma que "João saiu" foi verdadeiro ontem e a interpretação dada acima diz que "João chegou" em algum tempo no passado para trás de ontem.

Uma primeira tentativa de solução seria subcategorizar os advérbios de tempo em advérbios de tempo presente, de tempo passado e de tempo futuro. Uma consequência correta disto seria a de bloquear "\*João chegará ontem". Porém esta subcategorização traz problemas porque existem advérbios como "hoje", "esta semana", "esta manhã", "este ano" que funcionam igualmente como advérbio de tempo presente, passado e futuro, como fica evidenciado nos exemplos seguintes:

João está no Rio hoje.

João esteve no Rio hoje.

João estará no Rio hoje.

Um modo simples de tentar resolver isso, dentro da proposta do modelo aqui chamado de Modelo F-P Melhorado, é a utilização da seguinte maquinaria:

- variáveis e quantificadores sobre tempo;
- operador EM, onde  $EM(t_1, x)$  é verdade num certo tempo  $t$  sse  $x$  é verdade no tempo denotado por  $t_1$ ;
- predicados de tempo como PAS, PRES e FUT onde:

PAS (t1) é verdade num certo tempo t sse  $t_1 < t$ ,  
 PRES (t1) é verdade num certo tempo t sse  $t_1 = t$ ,  
 FUT (t1) é verdade num certo tempo t sse  $t_1 > t$ .

Assim a sentença "João chegou ontem" pode ter como tradução:

$(\exists t) [PAS(t) \ \& \ (t \ C= \text{ontem}') \ \& \ EM(t, \text{sair}'(j))]$

onde C= deve ser lido como "está contido". Por outro lado, as sentenças "João está no Rio hoje", "João esteve no Rio hoje" e "João estará no Rio hoje" podem possuir como traduções respectivamente:

$(\exists t) [PRES(t) \ \& \ (t \ C= \text{hoje}') \ \& \ EM(t, \text{estar-no-Rio}'(j))]$

$(\exists t) [PAS(t) \ \& \ (t \ C= \text{hoje}') \ \& \ EM(t, \text{estar-no-Rio}'(j))]$

$(\exists t) [FUT(t) \ \& \ (t \ C= \text{hoje}') \ \& \ EM(t, \text{estar-no-Rio}'(j))]$

Aqui as variáveis t, t1, t2, etc e a constante "hoje" são entendidas como denotando intervalos de tempo com valores; por outro lado,  $[t < t_1]$  afirma que cada momento dentro de t é anterior a cada momento dentro de t1.

Se for colocada a constante indexal t\* como denotando, para qualquer intervalo i, o próprio intervalo i, é possível estabelecer as seguintes propriedades:

$x = EM(t^*, x)$

$Px = (\exists t) [t < t^* \ \& \ EM(t, x)]$

$Fx = (\exists t) [t > t^* \ \& \ EM(t, x)]$

$EM(t, \sim x) = \sim EM(t, x)$

$EM(t, EM(t_1, x)) = EM(t_1, x)$

onde a primeira fórmula indica que se x é verdade em t\*, então x é verdade. Por outro lado, a segunda e a terceira fórmulas mostram a definição dos operadores priorianos de passado e futuro em termos de EM, t\*, e variáveis e quantificadores sobre tempo; assim dizer "passado x" (ou seja, Px) é dizer que existe um tempo anterior a t\* no qual x é verdade; dizer "futuro x" (ou seja Fx) é dizer que para algum t mais tardio do que t\*, x é verdade em t. Por seu turno a quarta fórmula diz que a negação distribui sobre o operador EM. A última fórmula diz que quando dois ou mais operadores EM são encaixados, todos menos o mais interno são vacuosos, se a expressão de tempo deste EM mais interno não for t\*.

Existe um problema contudo com esta proposta simples: as expressões usadas acima para traduzir as sentenças do Português não são exatamente adequadas como representações dos significados destas sentenças. Isto ocorre porque PAS(t), PRES(t) e FUT(t) devem ser tomados como implicações convencionais (ou pressuposições) das sentenças, ficando o resto das expressões denotando propriamente a asserção feita pela sentença. Isto deve ser assim porque "\*João não virá ontem" deveria ser vista como falsa e também como inapropriada. Isto iria requerer então duas denotações para a sentença: uma referencial e outra implicada, como é a proposta que está em /KAR 75/.

Se este problema é afastado como purismo, deve ser constatado que a incorporação de traduções do tipo das feitas acima ao PIQ implica em modificações importantes, porque o PIQ não usa expressões para denotar diretamente o tempo: o tempo no PIQ está envolvido na definição do modelo mas funcionando como membros de índices, os quais são argumentos de funções denotadas pelas expressões do tipo  $\langle s, a \rangle$  para algum tipo a.

Existem em princípio três modos de modificação da lógica intensional do PTQ de modo a torná-la capaz de referir-se diretamente a tempos. A primeira forma é incluir tempos no domínio das entidades básicas De; isto implica em usar uma lógica intensional com classificação dentro de um tipo, tal que possa existir um conjunto especial de variáveis que apontem somente para um subconjunto próprio de De, qual seja o subconjunto dos tempos. Esta proposta está embrionariamente em Cooper /COO 75/ e em Carlson /CAR 77/.

A segunda possibilidade é o estabelecimento de que uma proposição que é verdadeira só no tempo  $t$  em cada mundo possível possa "representar" o tempo  $t$ . Aqui também existe necessidade de algum tipo de classificação dentro dos tipos.

A terceira possibilidade é mais simples e envolve a introdução de um novo tipo primitivo: o tipo  $i$  de expressões que denotam intervalos de tempo. A definição de tipos fica então:

- $e, t, i$  são tipos;
- se  $a$  e  $b$  são tipos, então  $\langle a, b \rangle$  é um tipo;
- se  $a$  é um tipo, então  $\langle s, a \rangle$  é um tipo;
- nada mais é um tipo.

A última proposta será a desenvolvida aqui. Para tanto, é necessário redefinir um índice como um par ordenado  $\langle w, i \rangle$  onde  $w$  é um mundo possível e  $i$  é um intervalo de tempo.

Avançando um pouco mais a proposta, deve ser notado que existem expressões temporais que envolvem claramente quantificação sobre o tempo, mais do que referência sobre um intervalo único de tempo, como por exemplo as sentenças como "João bebe quando Maria bebe", "João canta algumas vezes", "Maria canta frequentemente". Isto faz com que seja útil possuir uma categoria  $T_m$  de expressões do Português que denotem conjuntos de propriedades de tempo; isto é análogo à categoria  $T$  de Montague para denotar conjuntos de propriedades de indivíduos.

Desta forma, expressões como "quarta-feira", "meia-noite" e "Natal" podem ficar como categorias básicas da categoria  $T_m$ , de tal modo que  $T_m$  fique distinta da categoria dos advérbios temporais  $T_mAv$ . Isto é feito assim porque expressões como "quarta-feira" são usadas antecedidas de preposição quando funcionam como  $T_m$ , como em "João chegou na quarta-feira"; porém outras expressões como "ontem" e "amanhã", funcionando como  $T_mAv$ , não requerem o uso da preposição ("\*João chegou em ontem").

Além disso o significado de "na quarta-feira", em relação ao intervalo  $i$ , é o conjunto de todos os conjuntos, onde cada conjunto contém um intervalo dentro de quarta-feira, isto é:

$$\lambda \text{Pt} (E t) [t \subseteq \text{quarta-feira} \ \& \ \text{Pt}(t)]$$

Por outro lado, algumas expressões em  $T_m$  também podem ocorrer adverbialmente sem preposição, como "João chegou quarta-feira"; estas expressões poderiam vir de  $T_mAv$  sem preposição, porém isto nem sempre pode ser feito (por exemplo "\*Pedro chegou noite"). As condições que governam este "apagamento" são obscuras no momento.

Aceitando as propostas feitas acima em bloco, é possível estipular as seguintes regras de concatenação entre os tempos principais e os advérbios de tempo. A numeração das regras da gramática está iniciando ao acaso, no momento.

Regra do Pretérito Perfeito Simples com Advérbio

S36. Se a é um TmAv e b é um t, então bia é um t, onde bi é o resultado da mudança do verbo principal para o pretérito perfeito simples.

T36. Se a é um TmAv e b é um t, então (bia)' é igual a a' (lambda t [PAS (t) & EM (t, b')]).

Regra do Presente Simples com Advérbio

S37. Se a é um TmAv e b é um t, então bia é um t, onde bi é o resultado da mudança do verbo principal para o presente simples.

T37. Se a é um TmAv e b é um t, então (ba)' é igual a a' (lambda t [PRES (t) & EM (t, b')]).

Regra do Futuro do Presente Simples com Advérbio

S38. Se a é um TmAv e b é um t, então bia é um t, onde bi é o resultado da mudança do verbo principal para o futuro do presente simples.

T38. Se a é um TmAv e b é um t, então (bia)' é igual a a' (lambda t [FUT (t) & EM (t, b')]).

As regras específicas para os advérbios temporais são:

S39. hoje, ontem, de-noite, na-quarta-feira pertencem a TmAv.

T39. hoje => lambda Pt (Et)[(t C= hoje') & Pt (t)]

ontem => lambda Pt (Et)[(t C= ontem') & Pt (t)]

de-noite => lambda Pt [Pt (noite)]

na-quarta-feira =>

lambda Pt (Et)[(t C= quarta-feira') & Pt (t)]

onde Pt é uma variável usada para indicar propriedades de tempo e => é lido como "traduz-se em".

Assim "João chega hoje" é analisado como:

João chega hoje, t, 36

```

      /      \
    hoje, TmAv  João chegar, t, 4
                /      \
              João, T   chegar, IV
  
```

e tem como tradução:

lambda Pt (Et)[(t C= hoje') & Pt(t)] (lambda t [PAS(t) & EM(t, lambda P [P(j)] ('chegar'))])

que é reduzida para

(E t) [(t C= hoje') & PAS(t) & EM (t, chegar'(j))].

A sentença "João chegará à noite" tem como árvore de análise:

João chegará de noite, t, 38

```

      /      \
    de-noite, TmAv  João chegar, t, 4
                    /      \
                  João, T   chegar, IV
  
```

e como tradução:

lambda Pt [Pt(noite)] (lambda t [FUT(t) & EM(t, chegar'(j))])

que é reduzida para

FUT (noite') & EM (noite', chegar(j)).

A tradução de "João chegou na quarta-feira" fica como:

$(E t) [(t C= \text{quarta-feira}') \& \text{PAS}(t) \& \text{EM}(t, \text{chegar}(j))]$

onde PAS(t) é o mesmo que  $t < t^*$ .

Um primeiro problema com esta proposta é a existência de sentenças como "Pedro encontrou João às duas horas da tarde da quarta-feira na primeira semana de abril de 1936", porque aqui deve existir um só tempo que satisfaça simultaneamente todas as quatro especificações de tempo (duas horas da tarde, quarta-feira, primeira semana, abril de 1936). Deve ser lembrado que, quando os operadores tipo EM (t2, a) são iterados, todos os operadores EM exceto o mais interno são vazios. Isto é: EM(t1, EM(t2, a)) é verdade num tempo t sse a é verdade em t2, não dependendo de t1. E com isso o único tempo que sobrar será "abril de 1936", o que é errado.

Um outro problema é que "\*João chegará na segunda-feira na quarta-feira" pode ser gerado pelas iterações sobre as regras 36-38, levando à interpretação que João chegará na quarta-feira.

Além disso, como sentenças sem advérbios existem, devem existir regras correspondentes a 36-38 para tal. Uma solução é postular um advérbio "fonologicamente nulo" (constante vazia) que cumpriria o papel do a das regras 36-38; uma solução aparentemente melhor é adicionar regras com uma só entrada, como abaixo:

Regra do Pretérito Perfeito Simples

S40. Se b é um t, então b1 é um t, onde b1 é o resultado da substituição do verbo principal de b pelo seu pretérito perfeito simples.

T40. Se b é um t, então  $(b1)' = (Et) [\text{PAS}(t) \& \text{EM}(t, b')]$ .

Regra do Futuro do Presente Simples

S41. Se b é um t, então b1 é um t, onde b1 é o resultado da substituição do verbo principal de b pelo seu futuro do presente simples.

T41. Se b1 é um t, então  $(b1)' = (Et) [\text{FUT}(t) \& \text{EM}(t, b')]$ .

#### 7.4.2 Advérbios temporais aspectuais no modelo E-E melhorado

Os advérbios como "hoje", "ontem", etc discutidos antes são como tempos na localização do tempo de verdade do verbo com relação ao tempo de fala. Já os advérbios tipo "por uma hora", "em uma hora", "frequentemente" são como funções que funcionam como operadores aspectuais.

Estes advérbios aspectuais não criam problemas na interação com os advérbios de tempo, mas só apontam para significados diferentes. Assim "João dorme no seu emprego frequentemente em seis semanas" diz que João dorme várias vezes dentro de um período de seis semanas; por outro lado "João dorme no seu emprego por uma hora frequentemente" diz que João dorme por períodos de uma hora de um modo frequente.



Assim sendo, as expressões "uma hora" e "seis semanas" podem ser tratadas como expressões básicas que denotam conjuntos de intervalos; isto é: "seis semanas" denota, para um certo índice, o conjunto de intervalos que possuem exatamente seis semanas de duração. Desta forma as preposições temporais "em" e "por" ficarão pertencendo à categoria (IV/IV)/(t/i), de modo que elas se combinam com uma expressão denotando uma propriedade de intervalo (t/i) e geram uma locução verbal com advérbio (IV/IV).

Por razões que ficarão evidenciadas abaixo, na tradução das regras de aplicação funcional desta categoria, será aqui definida uma constante indexal n (para "agora"), denotando, para um certo índice, a coordenada de tempo deste índice. Assim:

Para um índice  $\langle w, i \rangle$ , a denotação de n é i.

Aplicando esta definição para um caso específico, se n ocorrer no escopo de EM (t, x) e se não estiver encaixado sob algum operador de tempo, então n denota t.

Um comentário final antes de propor as regras de tradução para "por" e "em" é que em alguns casos eles podem desambiguar uma locução verbal que poderia ter uma leitura perfeitaiva (como por exemplo "Pedro leu um livro em uma hora") e uma leitura imperfeitaiva (como em "Pedro leu um livro por uma hora"). Esta questão não é desenvolvida contudo aqui, ficando o tema principal por conta das interações das formas [em advérbio] e [por advérbio] com as quatro classes de predicados apresentadas acima. Assim, inicialmente será apresentada uma discussão sobre [por advérbio] e depois sobre [em advérbio].

As regras relativas a "por" são:

S42. por pertence a (IV/IV)/(t/i)

T42. por traduz-se em

$\lambda$  Pt  $\lambda$  P  $\lambda$  x [Pt(n) & (Et)C(t C= n) -> EM(t, P(x)) ] ]

Com estas regras "João dormiu por uma hora" tem a seguinte árvore de análise:

João dormiu por uma hora, t, 36

!

João dorme por uma hora, t, 4

João, t dorme por uma hora, IV, 7

por uma hora, IV/IV dorme, IV

por, (IV/IV)/(t/i) uma hora, t/i

e a tradução é:

(Et1)[PAS(t1) & EM(t1, [uma-hora'(n) & (Et2)C(t2 C= n) -> EM(t2, dorme'(j)) ] ] ) ]

e, pelo princípio da interpretação de n, é possível eliminar n, e a tradução fica:

(Et1)[PAS(t1) & EM(t1, [uma-hora'(t1) & (Et2)C(t2 C= t1) -> EM(t2, dorme'(j)) ] ] ) ]

e, como uma-hora' é um designador rígido (porque ele é referido a diferentes tempos num mesmo mundo possível), é possível eliminar o primeiro operador EM e gerar:

(Et1)[PAS(t1) & uma-hora'(t1) & (Et2)C(t2 C= t1) -> EM(t2, dorme'(j)) ] ]

O significado de "por" proposto em T42 não possui um caráter geral para todas as expressões do Português. Em primeiro lugar as expressões da forma [por advérbio] são usadas habitualmente para várias atividades de modo que elas não fazem referência a literalmente todos os subintervalos do intervalo medido, mas meramente para todos os subintervalos maiores do que os intervalos mínimos para estas atividades em questão. Em segundo lugar, mesmo esta modificação apresenta problemas quando aplicada à sentenças como "João trabalha na cidade por quatro anos, mas ele usualmente passa suas férias em Tramandai". Em terceiro lugar a duração especificada por [por advérbio] pode ser a duração da união de intervalos não contíguos, como em "João serviu ao comitê eleitoral por quatro anos", a qual pode ser verdade mesmo se João serviu por quatro não consecutivos períodos de um ano. Uma saída possível para [por x] é talvez assertar que alguma coisa é verdadeira para cada um dos conjuntos dos possivelmente não contíguos intervalos de tempo, tal que a duração total seja x. A escolha exata dos membros destes conjuntos fica para a interpretação contextual.

Em relação a preposição "em", as regras seriam:

S43. em é um (IV/IV)/(t/i)

T43. em traduz-se como

$\lambda P t \lambda P \lambda x [Pt(n) \ \& \ (Et)[(t \ C= n) \ \& \ Em(t, P(x))]]$

A regra T43 especifica que o tempo da verdade do verbo é algum subconjunto do intervalo mencionado, embora não necessariamente um subconjunto próprio. Porém usualmente, por razões ligadas à implicação conversacional,  $t=n$  no caso de existirem múltiplos objetos de ação, como em "João lavou as janelas em uma hora". Mas se o verbo é um simples verbo de mudança, o tempo esperado normalmente para este tipo de verbo é bem menor do que o tempo especificado pelo advérbio, como em "João fechou a porta em uma hora"; neste caso, se for dito que João fechou a porta em uma hora e ele de fato fechou a porta dentro de um intervalo de cinco minutos, a sentença "João fechou a porta em uma hora" não é falsa, porém uma declaração mais restrita seria mais apropriada à situação no caso de uma conversação. Por outro lado, "Eu o encontro em uma hora", em oposição, não estabelece que o intervalo é ao menos uma hora, mas esta sentença pode conversacionalmente implicar que este tempo não é tão longo; um outro ponto neste último caso é a determinação do início da medida do intervalo, no caso onde o verbo é verdadeiro somente no subintervalo final: este ponto parece ser determinado apenas na conversação.

Um problema importante com T43 é que ela não explica porque [em advérbio] não ocorre naturalmente com verbos estativos. Assim "João dormiu em uma hora" não é muito natural para uma visão estativa de dormir, só parecendo verdadeira em uma interpretação que João ficou acordado por uma hora e depois dormiu. Para resolver isso, a tradução de "em" deve ser mudada para T42':

T42'. em traduz-se como

$\lambda P t \lambda P \lambda x [Pt(n) \ \& \ (Et)[(t1 \ C= n) \ \& \ Em(t1, P(x)) \ \& \ (Et2)[[(t1 \ C= n) \ \& \ Em(t2, P(x))] \rightarrow t2 = t1 ] ] ]$

que diz que o verbo é verdade num subintervalo único, embora não

necessariamente um subintervalo próprio do intervalo medido. Além disso, o requerimento da unicidade de  $t_1$  é parte da implicação conversacional, e não parte da asserção.

Assim "João adormeceu em uma hora" fica traduzido, segundo T42', em:

$$\begin{aligned} & (E t_1) [ P A S (t_1) \ \& \ E M (t_1, \ [ \text{uma-hora}'(n) \ \& \ (E t_2) [ (t_2 \ C = n) \ \& \\ & \quad E M (t_2, \ [ T O R N A R \ \text{adormecer}'(j) ] ] ) \ \& \\ & \quad (E t_3) [ [ (t_3 \ C = n) \ \& \ E M (t_3, \ [ T O R N A R \ \text{adormecer}'(j) ] ] ) ] \ ] \ ] \rightarrow \\ & \quad (t_2 = t_3) ] ] \ ] \end{aligned}$$

sendo usada a forma  $[TORNAR \text{adormecer}'(j)]$  para denotar a leitura incoativa; após a eliminação da constante  $n$  e do operador  $EM$  correspondente, a fórmula acima fica:

$$\begin{aligned} & (E t_1) [ P A S (t_1) \ \& \ \text{uma-hora}'(t_1) \ \& \ (E t_2) [ (t_2 \ C = t_1) \ \& \\ & \quad E M (t_2, \ [ T O R N A R \ \text{adormecer}'(j) ] ] ) \ \& \\ & \quad (E t_3) [ [ (t_3 \ C = t_2) \ \& \ E M (t_3, \ [ T O R N A R \ \text{adormecer}'(j) ] ] ) ] \ ] \rightarrow \\ & \quad (t_2 = t_3) ] ] \ ] \end{aligned}$$

porém, mesmo assim, não é claro como "João dormiu em uma hora" requer o tratamento de "dormir" como ambíguo entre uma interpretação estativa como "dormir" e uma interpretação incoativa como  $[ \lambda x \ [ TORNAR \text{dormir}'(x) ] ]$ . Por suposição, se a interpretação usada for a estativa, então deve ser tomado o intervalo de uma hora de duração tal que o último momento deste intervalo seja um tempo no qual "João dorme" é verdade, mas antes deste intervalo "João dorme" deve ser falso. Isto quer dizer então que existe um único intervalo no qual a hora na qual "João dorme" é verdade, embora isso não seja verdade se existirem dois momentos de sono de João dentro desta hora; só levando em conta todos estes casos a interpretação de "em uma hora" pode ser satisfeita.

Por último, se for requerido que o tempo de elocução não seja maior do que um momento, então a interpretação correta de "João está em Recife por seis semanas" (usa "em" e "por") não pode ser obtida pela aplicação das regras acima descritas, porque aqui o presente deve ocorrer em um intervalo. A única leitura que esta sentença tem é a leitura como futuro simples, na qual é estabelecido um evento determinado como referência. Por outro lado, devem existir razões pragmáticas (e não só semânticas) para que "João chega em Recife em uma hora" tenha só uma leitura no futuro simples. Estes problemas ficam aqui em aberto.

Numa análise geral sobre todo o material aqui apresentado sobre a semântica proposta para  $[por \text{ advérbio}]$  e para  $[em \text{ advérbio}]$ , deve ter ficado claro que  $[por \text{ advérbio}]$  é apropriado para estativos e atividades, mas não para resultativos com ou sem objeto associado à ação: os primeiros predicados são verdadeiros para todos (ou a maioria) dos subintervalos de um intervalo; os segundos por outro lado, não são predicados de subintervalos. Por outro lado,  $[em \text{ advérbio}]$  é adequado para resultativos com ou sem objeto associado à ação, porque eles satisfazem o requerimento de unicidade na semântica de "em"; mas isso não ocorre para atividades e estativos, exceto no caso da leitura incoativa destes últimos. Dentro dessa visão simplificada,  $[por \text{ advérbio}]$  e  $[em \text{ advérbio}]$  separam estativos/atividades de resultativos com ou sem objeto associado à ação.

7.4.3 O significado de [estar [presente simples] + gerúndio] e o modelo I=0

Um problema pode ser estabelecido já para estas formas gerundivas, através da comparação das sentenças "João está desenhando um círculo" e "João está empurrando um carro": a primeira sentença não implica que João desenhou um círculo, mas a segunda implica que João empurrou um carro. A primeira é um tipo de resultativo sem objeto associado à ação, podendo ser analisado como [X CAUSAR [TORNAR Y]] e a segunda é uma atividade analisável como [FAZER (a, X)].

A proposta de análise a ser feita aqui envolve postular a existência do operador FG, o qual toma uma fórmula e gera sua forma gerundiva. Assim para resultativos sem objeto associado à ação vale apenas a inferência:

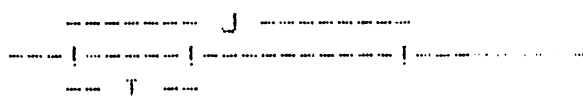
FG[X CAUSAR [TORNAR Y]] → X,

sendo que as formas gerundivas para eles não são comuns, existindo porém sentenças como "João está morrendo", as quais não implicam em [TORNAR Y].

Para os casos com FG, a verdade de uma sentença não pode ser definida em relação a um instante de tempo, mas sim em relação a um intervalo de tempo. Uma saída poderia ser a utilização dos operadores G e H de Prior, ou mesmo dos operadores SINCE e UNTIL de Kamp /KAM 68/ que expressam verdade sobre um intervalo. Porém, com eles, a sentença ligada ao intervalo fica verdadeira só nos casos onde a sentença encaixada (em relação ao operador) é verdadeira em todos os momentos durante este intervalo. Isto contudo vale para predicados estativos com advérbios durativos ("João viveu em Mato Grosso por seis semanas"), mas não vale quando uma sentença com resultativo de qualquer tipo é verdadeira em intervalo maior do que um momento, como em "Eu dei uma hora para João desenhando um círculo". Neste caso, [João desenhando um círculo] não precisa ser verdadeira em todos os momentos durante o intervalo de uma hora, como foi discutido no início desta seção.

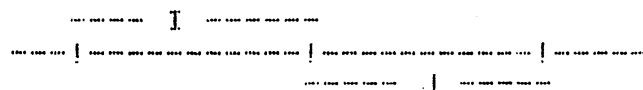
Para resolver casos como estes, Bennet e Partee /BEN 78/ propõem uma semântica intensional que não é baseada em índices instantâneos (como existe no PTQ), mas baseada em intervalos, num ordenamento linear denso (aqui chamado de Modelo I-D). Assim seja T o conjunto de momentos de tempo, <= uma operação de ordenamento linear denso standard em T, I um intervalo. É possível convencionar que [t1, t2] denota um intervalo fechado, isto é {t: t1 <= t <= t2}. Por outro lado (t1, t2) denota um intervalo ligado, ou seja {t: t1 < t < t2}. Além disso [t] denota um momento, abreviando [t, t], ou seja {t}. Assim é possível estabelecer as seguintes definições:

- I é subintervalo de J sse (I C= J);
- I é subintervalo próprio de J sse (I C J);
- I é subintervalo inicial de J se (I C= J) e não existe t que pertença a (J - I) para t1 pertencente a I e (t <= t1);



- I é subintervalo final de J sse (I C= J) e não existe t que pertença a (J - I) para t1 pertencente a I e (t )= t1);

- I é intervalo ligado inicial de J sse I e J são disjuntos, a união de I e J é um intervalo e I é um subintervalo inicial da união de I com J;

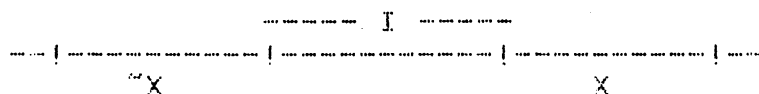


- I é intervalo ligado final de J sse I e J são disjuntos, a união de I e J é um intervalo e I é um subintervalo final da união de I com J.

Com estas definições, [TORNAR X] é verdade em I sse:

- existe um intervalo J contendo o subintervalo ligado inicial I, tal que "X é verdade em J;

- existe um intervalo K contendo o intervalo ligado final de I tal que X é verdade em K.



Analisando agora a significação da forma gerundiva, por comparação com a significação do pretérito perfeito simples, uma proposta inicial é que ela seja usada para descrever um evento mais relevante do que quando este evento é descrito pelo pretérito perfeito simples; um problema aqui é que a noção de "relevância atual" não é muito clara, e por outro lado não é muito claro também porque "João está visitando o Rio" descreve um evento que tem maior relevância atual do que o evento descrito por "João visitou o Rio". Uma segunda proposta é discutir se a forma gerundiva implica numa asserção menos definida do que a feita com o pretérito perfeito; o problema aqui é o de especificar esta menor definição.

Uma outra maneira de comparar a significação da forma gerundiva com a significação do pretérito perfeito simples é observar a interação destes tempos verbais com os advérbios; em relação a isso é fácil constatar que alguns advérbios como "ontem" ocorrem com o pretérito perfeito simples, mas não com a forma gerundiva; já advérbios como "desde 1971" ocorrem com a forma gerundiva mas não com o pretérito perfeito simples. Isto é mostrado nos exemplos a seguir:

- João chegou ontem / \*João está chegando ontem
- João viveu em Canela por seis semanas / João está vivendo em Canela por seis semanas
- \*João viveu em Canoas desde 1971 / João está vivendo em Canoas desde 1971.

Uma classificação de advérbios entre os tempos pretérito perfeito simples (pps) e a forma gerundiva (fg) seria:

+ pps/- fg: cinco anos, uma vez, na última noite, em 1945, no passado;

+ pps/+ fg: nunca, as tres horas, tardiamente, depois da guerra, antes da guerra, recém agora, por tres anos, as vezes, sempre, já, antes, até agora, durante os últimos anos, nos últimos anos;



A tradução respectiva é:

(Et1)[AX(t1) & (Et2)[(t2 C= t1) & uma-hora'(t2) &  
(Et3)[(t3 C= t2) -> EM(t3, dormir'(j))]] ] ]

devendo ser observado que o último momento do intervalo "por uma hora" é o instante presente. Além disso, o intervalo "agora estendido" (que começa no passado e que se estende até incluir o tempo de fala) é identificado pelo advérbio.

Por outro lado, uma sentença como "João está parando aqui desde quarta-feira, mas ele saiu um minuto atrás" é inaceitável, porque o intervalo apontado por "desde quarta-feira" necessariamente atinge o instante presente. Considerando que "desde" faz parte dos advérbios marcados com [- pps/+ fg], podem ser propostas as seguintes regras para "desde":

S45. desde é um TmAv/Tm.

T45. desde traduz-se em

lambda P2t lambda Pt (lambda t1 [(Et2)[[(t1 < t2) & AX(t2)] ->  
Pt (t2) ]])

onde P2t denota propriedades de propriedades de tempo, [(t1 < t2) & AX(t2)] tem o efeito de fazer com que t2 (o tempo de verdade do verbo) varie sobre todo o subintervalo final do intervalo do advérbio.

Em relação a construções tipo [por advérbio], como em "João está dormindo por uma hora", as regras para "por" seriam:

S46. por é um TmAv/(t/i) (diferente da regra 45 acima).

T46. por traduz-se em

lambda Pt lambda Qt (Et1) [AX(t1) & Pt (t1) & (Et2) [ [(t2 C= t1) & AX(t2) ] -> Qt (t2) ] ]

onde [(t2 C= t1) & AX(t2)] tem o efeito de fazer com que t2 (o tempo de verdade do verbo) varie sobre os subintervalos finais do intervalo do advérbio.

Quando estativos e atividades simples são envolvidos, é obrigatório que o verbo seja verdadeiro em todos os subintervalos finais do intervalo do advérbio. Neste caso porém não é correto restringir T45 e T46 a [t1 < t2 =< n] e [t2 C= t1] respectivamente em lugar das cláusulas dadas, porque a tradução de "estar" via T44 iria adicionar a asserção de que t2 é um AX e isso ficaria contraditório.

Considerando o exposto acima, uma sentença como "João está dormindo desde meia-noite" tem como árvore de análise

João está dormindo desde meia-noite, t, 4

João, T está dormindo desde meia-noite, IV, 45

desde meia-noite, TmAv dorme, IV

desde, TmAv/Tm meia-noite, Tm

e tem como interpretação

(Et2) [ [(meia-noite' < t2) & AX(t2)] -> [AX(t2) & EM(t2, dormir'(j)) ] ]

e por outro lado uma sentença como "João está dormindo por uma hora" tem como árvore de análise

João está dormindo por uma hora, t, 4  
 João, T está dormindo por uma hora, t, 45  
     por uma hora, TmAv   dormir, IV  
     por, TmAv/(t/i)   uma hora, t/i  
 e como interpretação  
 (Et1) [AX(t1) & uma-hora'(t1) & (Et2) [ [ (t2 =C t1) &  
     AX(t2) ] -> [AX(t2) & EM(t2, dormir'(j)) ] ] ]  
 Um problema nestas duas traduções é que a última  
 ocorrência de AX(t2), que deriva de T45, é ociosa. Porém ela não  
 pode ser eliminada porque caso contrário as regras permitiriam  
 gerar "João está chegando ontem".

#### 7.4.4 Uma outra proposta: o modelo D-I

A idéia inicial aqui será tentar uma análise da  
 interação entre tempo verbal e advérbio numa abordagem  
 inteiramente afastada da lógica do tempo de Prior, que é a usada  
 no PTQ. A alternativa será o uso de uma lógica bidimensional,  
 baseada em /KAM 71/.

A proposta da lógica de Kamp foi introduzida  
 originalmente para representar a diferença entre "Um menino que  
 seria rei nasceu" e "Um menino que será rei nasceu". A primeira  
 sentença é representada em lógica prioriana como

$P (Ex) [menino'(x) \& nascer'(x) \& F (rei(x)) ]$

onde ocorre o encaixamento do operador para futuro F dentro do  
 operador para passado P, permitindo corretamente que o tempo de  
 ser rei esteja ou antes ou mais tarde do que o tempo da elocução,  
 e que o tempo de ser rei seja posterior ao tempo de ter nascido.

Porém a segunda sentença não pode ser representada com os  
 operadores de Prior, porque ela coloca de um modo não ambíguo que  
 o tempo de ser rei é mais tardio que o tempo da elocução. Para  
 representar a segunda sentença, Kamp usou o operador N, onde [Nx]  
 é lido como "agora x". Este operador tem, por definição, o efeito  
 de declarar que x é verdade em relação ao tempo da elocução.  
 Assim a segunda sentença pode ser representada como

$P (Ex) [criança(x) \& nascer'(x) \& N F (rei'(x)) ]$

A proposta a ser desenvolvida aqui assume que a  
 definição de verdade de uma fórmula em relação ao tempo j de  
 elocução é dada através de uma definição intermediária de verdade  
 relativa ao par de tempos <i, j>. Assim uma sentença atômica x  
 (sem nenhuma referência para tempo) é definida como verdade  
 relativamente a <i, j> (chamado aqui verdade2) sse as condições  
 apropriadas valem para i, para algum j. Uma sentença formada pela  
 adição de um advérbio temporal y à sentença x é verdade2 em <i,  
 j> sse x é verdade em <i, j> e i é o tempo com a propriedade  
 especificada por x. Uma sentença formada pela adição de um tempo  
 a x é verdade2 em <i, j>, de outro lado, sse x é verdade1 em <i,  
 j> e i está numa relação apropriada para j (por exemplo: no caso  
 do tempo pretérito perfeito, i é anterior a j). Finalmente, a  
 definição de verdade relativa a um único tempo (verdade1) é dada



pela declaração de que  $y$  é verdade<sub>2</sub> em relação ao tempo  $i$  se existe algum tempo  $k$  tal que  $x$  é verdade<sub>1</sub> relativo a  $\langle i, j \rangle$ .

A seguir esta nova proposta será discutida a partir de um caso concreto que é o da interpretação de sentenças em que o verbo está no tempo pretérito perfeito simples. Partee /PAR 73a/ propõe que o tempo pretérito perfeito simples seja indexado com seu tempo de referência; assim uma sentença como "Pedro não retornou do Rio" não significa que existe um tempo no passado no qual Pedro não retornou do Rio, nem que não existe um tempo no passado no qual Pedro retornou do Rio. A sentença referir-se-ia basicamente, pela proposta de Partee, a um intervalo particular definido cuja identidade seria clarificada pelo contexto extralinguístico. Se isto for considerado correto, o modo de interpretar o tempo de referência do pretérito perfeito não deveria ser com um quantificador existencial, como na Regra do Pretérito Perfeito mostrada abaixo:

S47. Se  $b$  pertence a  $t$ , então  $b_1$  pertence a  $t$ , onde  $b_1$  é o resultado da substituição do verbo principal de  $b$  pelo seu pretérito perfeito

T47. Se  $b$  pertence a  $t$ , então  
 $(b_1)' = (Et) [PAS(t) \& EM(t, b')]$

mas a solução envolveria o uso de uma constante indexical que ficaria interpretada como um parâmetro contextual, semelhante aos indexicais "eu", "este", etc.

Aceitando isto e ligando com a proposta oferecida acima, no caso de uma sentença no pretérito perfeito simples com um certo número de advérbios de tempo, esta sentença pode ser verdadeira somente no caso em que ao menos exista um tempo passado apropriado para todos os advérbios e em que a sentença sem tempo seja verdadeira /KAM 71/.

Existe aqui uma similaridade entre o uso destes dois índices e a distinção de Reichembach /REI 47/ entre tempo de fala e tempo de referência, sendo que  $i$  é análogo ao tempo de referência e  $j$  é análogo ao tempo de fala. A diferença aqui é que  $j$  não tem nenhum papel na interpretação da sentença.

Aceitando a proposta de Partee /PAR 73/ então é necessário introduzir um segundo índice de tempo, não só para interpretar corretamente os advérbios, mas também para levar em conta as condições da pragmática. Esta característica é mesmo uma diferença pela qual o pretérito perfeito simples difere do pretérito perfeito composto ("tem chegado"), pois a forma [tem + participio] não requer que o falante indique precisamente o tempo, ao contrário da forma simples.

A diferença entre usar um modelo kampioniano e usar um modelo reichembachiano é que a diferença entre o pretérito perfeito simples e o pretérito perfeito composto é vista, na proposta de Reichembach /REI 47/, através de três índices de tempo distintos: tempo de fala (S), tempo do evento (E) e tempo de referência (R). Segundo esta proposta, o pretérito perfeito simples tem seu R e seu E juntos e antes de S; já o pretérito perfeito composto tem S e R juntos e E antes.

R, E ----- S : pretérito perfeito simples  
E ----- R, S : pretérito perfeito composto

Porém McCoard /McC 78/ mostrou que a proposta de Reichembach falha completamente na abordagem semântica da diferença entre os dois tempos porque ela fornece para as duas formas as mesmas condições de verdade. Sem diminuir a proposta de Reichembach, ela pode ser vista como uma teoria sobre uma diferença pragmática entre as duas formas, em especial porque ela usa a noção de tempo de referência a qual teria um lugar importante numa teoria da narração. A proposta desenvolvida aqui será contudo a kampiana.

Nesta proposta, os operadores PAS e PRES não trocam o ponto de referência para o qual a fórmula é avaliada, como o tempo prioriano faz: agora os dois índices  $i$  e  $j$  estão numa certa relação, tal que o operador passado diz que  $i$  é anterior a  $j$ , o operador presente diz que  $i$  é igual a  $j$ , o operador futuro diz que  $i$  é posterior a  $j$ . Assim sendo, os advérbios de tempo podem ser tratados sintaticamente como operadores de sentença, tendo o efeito semântico de estabelecerem que o tempo de referência está localizado num certo tempo (por exemplo: ontem). Desta forma é possível encaixar um operador de tempo, sem produzir o efeito semântico indesejável que resultava com os operadores priorianos referido acima.

Nesta proposta, a definição de verdade<sup>2</sup> (o significado da sentença  $x$  em relação a  $\langle i, j \rangle$ ) é verdade sse:

- expressões básicas: uma função de interpretação  $\$$  assinala um valor para cada expressão básica  $x$ , para cada tempo  $i$ ; depois o significado de  $x$  em relação a  $\langle i, j \rangle$  é igual a  $\$(x, i)$  para cada expressão básica  $x$ ;

obs.: o índice  $j$  não tem nenhum papel na determinação da interpretação das expressões básicas;

- sentenças atômicas (sem tempo): o significado da expressão da forma  $(x (y))'$  em relação a  $\langle i, j \rangle$  vale 1 sse  $x'$  em relação a  $\langle i, j \rangle$  aplicado a  $b'$  em relação a  $\langle i, j \rangle$  valer 1;

obs.: assumindo  $x'$  como o significado de  $x$  em relação a  $\langle i, j \rangle$ , então:

$(PRES x)'$  = 1 sse  $x' = 1$  e  $i = j$ ;

$(PAS x)'$  = 1 sse  $x' = 1$  e  $i < j$ ;

$(FUT x)'$  = 1 sse  $x' = 1$  e  $i > j$ ;

- (ESTAR  $x$ )' = 1, onde "estar" está no presente do indicativo e  $x$  está no gerúndio, sse o significado de  $x$  em relação a  $\langle ii, j \rangle$  for igual a 1, onde  $ii$  é um subintervalo final do intervalo  $i$ ;

obs.: o intervalo  $ii$  é algo análogo ao tempo do evento; assim considerando, o operador ESTAR funciona como um operador prioriano pois ele troca o ponto de avaliação de sua sentença de  $i$  para  $ii$ ;

- (EM  $\langle t, x \rangle$ )' = 1 sse a interpretação de  $x$  em relação a  $\langle ii, j \rangle$  for igual a 1, onde  $ii$  é igual a interpretação de  $t$  em relação a  $\langle i, j \rangle$ ;

obs.:  $j$  não exerce nenhum papel nesta definição;

-  $(t*)'$  = 1,

isto é:  $t*$  denota o tempo de referência.

Com estas definições, a sentença "João chegou" tem como tradução PAS [ chegar'(j) ]; esta fórmula será verdade quando o tempo de referência for anterior ao tempo de fala e a chegada de João ocorrer no tempo de referência. Por outro lado, "João chegou ontem", tem como tradução uma expressão formada sintaticamente pela tradução da sentença "João chegou" adicionada à tradução do advérbio de tempo, ou seja:

ontem' (lambda t [ (t = t\*) & PAS[ chegar'(j) ] ])

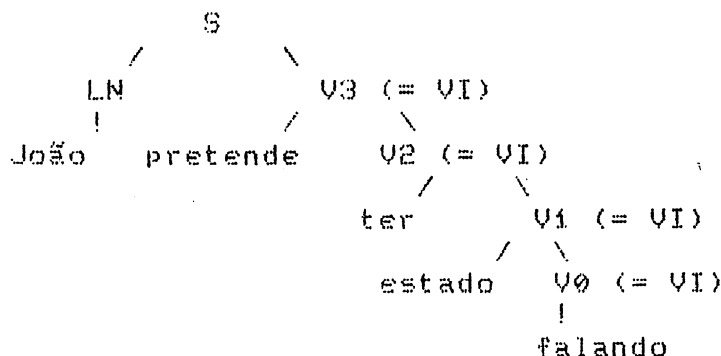
a qual será verdade quando ontem' contiver o tempo que é idêntico com t\* (isto é: com o tempo de referência) e o tempo de referência for anterior ao tempo de fala e João chegou no tempo de referência.

A definição de verdadei relativa a um único tempo de elocução é:

o significado de x em relação a j é igual a 1 (x é verdadei em j) sse existe algum i tal que x' = 1.

Devido a esta definição de verdadei, a tradução de "João chegou" fica equivalente à representação prioriana tradicional.

Considerando o sistema de auxiliares, as regras para as locuções verbais, podem ser estabelecidas, considerando como exemplo a sentença "João pretende ter estado comendo", que possui a seguinte árvore sintagmática (simplificada):



As regras do presente, passado e futuro passam a ser as seguintes:

S48. Se x é um IV, F48(x) pertence a V3, onde F48(x) é o tempo presente do indicativo de x.

T48. lambda F [ PRES ( x' ) ] 1.

obs.: a tradução está sendo dada de forma simplificada.

S49. Se x é um IV, F48(x) pertence a V3, onde F48(x) é o pretérito perfeito de x.

T49. lambda F [ PAS ( x' ) ] 1.

S50. Se x é um IV, F48(x) pertence a V3, onde F48(x) é o futuro simples de x.

T50. lambda F [ FUT ( x' ) ] 1.

Regra de ter + participípio passado:

S51. Se x é um IV, F51(x) pertence a V3, onde F51(x) = [teri x1], onde teri é o presente do indicativo de ter e x1 é o participípio passado de x.

T51. lambda F [ TEM ( x' (F) ) ] 1.

Regra do advérbio de tempo:

S52. Se  $x$  é um TmAv,  $y$  é um  $t$ , então  $F52(x, y) = yx$ .

T52.  $x'$  ( $\lambda t [ (t = t*) \& y' ]$ ).

A regra 48 muda um IV num V3 pela adição do morfema do tempo presente simples do indicativo a ela; a regra semântica correspondente forma um predicado de terceira ordem da tradução de IV, no qual o significado da locução nominal sujeito (que será o valor da variável P) aparece no escopo do operador PRES.

A regra 49 produz um V3 no tempo pretérito perfeito e a regra 50 produz um V3 no futuro simples do indicativo, com suas respectivas traduções.

A regra 51 forma um IV pela introdução de "ter" no presente simples do indicativo e a morfologia adequada para o verbo. A regra T51 introduz o operador de tempo TER, deixando o predicado no primeiro nível.

A regra 52 combina um advérbio de tempo com uma sentença gerando uma sentença; já T52 diz que o significado do advérbio temporal (isto é:  $x'$ ) identifica um tempo  $t$  tal que  $t$  é idêntico com o tempo da referência  $t*$ , e  $y$  é verdade.

Existem motivações importantes para o sistema de dupla indexação (modelo D-I) apresentado aqui. Uma delas é que este modelo fornece um correto tempo de referência para advérbios como "agora", quando eles aparecem encaixados dentro de outros operadores temporais. Assim em "João pensou que Pedro acreditou que Carlos estaria aqui agora" o tempo em que foi acreditado que Carlos estava aqui é o tempo de fala, e não o tempo da crença nem o tempo do pensamento, apesar do fato de que "agora" está sintaticamente encaixado num passado (pretérito perfeito) e num futuro. Num sistema prioriano como o usado anteriormente, esta conduta de "agora" não pode ser explicada porque estes outros operadores mudariam o ponto de avaliação fora do tempo de fala, tal que o tempo de fala ficasse como não recuperável no contexto encaixado. Num sistema de dupla indexação entretanto o valor do advérbio indexical de tempo (como "agora") pode depender do tempo de fala  $j$ , em vez do tempo de referência  $i$  (exatamente como a proposta original de Kamp). Assim as denotações de agora' e de ontem' ficam sendo:

a interpretação de agora' em relação a  $\langle i, j \rangle$  é igual a (o conjunto de todos os conjuntos contendo) o intervalo  $j$ ;

a interpretação de ontem' em relação a  $\langle i, j \rangle$  é igual a (o conjunto de todos os conjuntos contendo) um intervalo dentro do dia precedente ao dia que contém  $j$ .

Embora "agora" possa ser o único advérbio que se refira ao tempo de fala, mesmo em contextos encaixados, ele não é o único advérbio de tempo cuja denotação dependa do tempo de fala em contextos encaixados. Se for substituído "agora" por "ontem" na sentença anterior, gerando "João pensou que Pedro acreditou que Carlos estaria aqui ontem", pode ser observado que "ontem" refere-se a algum tempo ao dia precedente ao tempo de fala, mas não necessariamente no dia precedente ao tempo de crença ou ao tempo de pensamento.

Outro argumento a favor do modelo D-I leva em consideração o caráter indexical do pretérito perfeito. Este caráter indexical pode ser notado em sentenças como "João não pegou o trem", cuja interpretação não é devidamente representada pela fórmula prioriana "[PC João pega o trem]", que diz que não existe um tempo passado no qual João pegou o trem, nem pela outra fórmula prioriana P "[João pega o trem]", que diz que existe um tempo passado no qual João não pegou o trem.

Outra vantagem do modelo D-I é no tratamento de sentenças com vários advérbios temporais, como "João encontrou Pedro às 15 horas da terça-feira da segunda semana de maio de 1987". Na abordagem prioriana, ou deveriam ser adicionadas regras para inserção simultânea de dois, três, ou n advérbios junto com o tempo verbal, ou deveria ser adicionada uma regra que combinasse dois ou mais advérbios num advérbio composto e daí ligasse isso às regras de tempo verbal. Segundo o modelo D-I, a regra do tempo adverbial forma uma nova sentença pela adição de um advérbio temporal a uma sentença, e isso pode ser feito recursivamente. Assim a tradução da sentença acima seria:

de-1987' (lambda t [(t = t\*) & de-maio' (lambda t [(t = t\*) & da-segunda-semana' (lambda t [(t = t\*) & da-terça-feira' (lambda t [(t = t\*) & às-15-horas (lambda t [(t = t\*) & PAS (João-encontrar-Pedro') ] ) ] ) ] ) ] ) ] ) ] )

Nesta tradução, o tempo de referência t\* é idêntico com um tempo que está em 1987, um tempo que está em maio, que está na segunda semana, que está na terça-feira, que está às 15 horas, um tempo que é anterior ao tempo de fala, e finalmente um tempo no qual João encontra Pedro. A sentença então será verdadeira se for encontrado um tempo no qual todos estes requerimentos forem cumpridos. Por outro lado, uma sentença como "João encontrou Pedro na quarta-feira na terça-feira" não conseguirá ter um referência única de tempo que cumpra as condições para ambos os advérbios, e logo esta sentença é predita ser contraditória.

Uma outro local importante de utilização do modelo D-I diz respeito ao tratamento de sequências de tempo como encontrado na interação do tempo da oração principal com o tempo da oração subordinada. A esse respeito, Ladusaw /LAD 77/ mostrou que o fragmento do PTQ com seus operadores priorianos de tempo prediz que as sentenças "João achará um unicórnio que andou", "João achará um unicórnio que está andando" e "João achará um unicórnio que andar" fiquem todas ambíguas em relação ao tempo de referência da oração principal versus ao da oração subordinada. Assim por exemplo para a sentença "João achará um unicórnio que está andando", se [um unicórnio que está andando] for introduzido diretamente na árvore, como em

```

João achará um unicórnio que está andando
 /      \
João  achar um unicórnio que está andando
      /      \
      achar  um unicórnio que está andando
          /      \
          .....

```

então a tradução será

FC(Ex)Eunicórnio'(x) & andar'(x)] & achar'(j,x)]  
e logo a tradução da oração subordinada aparecerá dentro do escopo do operador de tempo da oração principal. Se, por outro lado, a locução nominal [um unicórnio que está andando] for introduzida via a regra de quantificação (S14), como abaixo:

João achará um unicórnio que está andando

um unicórnio que está andando João achará ele@

então a tradução será

(Ex)Eunicórnio'(x) & andar'(x) & F[achar'(j,x)]]

e a tradução da oração subordinada aparecerá fora do escopo do operador de futuro. Esta predição de ambiguidade está correta para as três sentenças referidas acima, porém esta predição está incorreta para sentenças como "Maria achou um unicórnio que andava na floresta", "Maria achou um unicórnio que está andando na floresta" e "Maria achou um unicórnio que andarà na floresta" como refere Ladusaw, pois a interpretação de "Maria achou um unicórnio que está andando" requer que o tempo de andar seja o mesmo que o tempo de fala.

A solução de Ladusaw envolve a utilização da sintaxe proposta por Cooper e Parson /COO 76/, a qual possui uma estrutura profunda não ambígua (produzida por regras de estrutura de frase) e esta estrutura profunda é convertida na estrutura superficial via transformações. Neste método, a estrutura profunda de uma leitura com escopo amplo para uma locução nominal é tal que esta locução nominal existirá numa estrutura de alto nível, e uma transformação (de rebaixamento do quantificador) irá inserir mais tarde esta alta locução nominal numa posição baixa da estrutura superficial. Ladusaw, assim fazendo, na verdade postula uma sequência de transformações de tempo, a qual copia um tempo passado num outro nodo de tempo que é comandado assimetricamente pelo passado; Ladusaw também ordena esta sequência de transformações após as transformações de rebaixamento de locução nominal. Apesar de todo este esforço na linha transformacional, Ladusaw reconhece que esta análise faz predicções erradas para exemplos como "João disse que alguém estará em seu apartamento" e "João disse que alguém estaria em seu apartamento", pois nestes exemplos não parece haver motivação para uma regra de quantificação que permita que a oração relativa tenha um escopo amplo, paralela à motivação para permitir que a locução nominal tenha um escopo amplo. Toda esta problemática leva Ladusaw a estabelecer que a sequência das transformações seja obrigatória, exceto quando uma fonte ad hoc de escopo amplo para a oração relativa é postulada. Outro problema com a proposta de Ladusaw é que a aplicação da sequência de transformações depende de a locução ter um escopo amplo ou não.

Por tudo isso, Ladusaw conclui seu artigo especulando que uma análise diferente possa tratar o tempo das orações subordinadas, chegando a sugerir que dois tempos passados encaixados sintaticamente não funcionem realmente como operadores priorianos (isto é: um sob o escopo do outro), mas sim que só ocorra um ordenamento entre os eventos em relação ao tempo de fala. Em relação a isso, Smith /SMI 75/ propõe que os dois

eventos (o descrito pela oração principal e o descrito pela oração subordinada) não fiquem ordenados entre si, mas só em relação ao tempo de fala, e assim não há necessidade de usar transformações.

Olhando do ponto de vista da lógica do tempo, a diferença aqui é que para um operador prioriano vale:

$$P [y \& P w] \neq [P y \& P w],$$

onde y e w são fórmulas, e para um operador kampiano vale:

$$PAS [y \& PAS w] = [PAS y \& PAS w]$$

isto é, y e w são verdade no tempo de referência passado. O que está em discussão aqui é se "O menino que caiu viu o gato" diz que a queda do menino foi anterior a êle (menino) ter visto o gato ou não, ou se a sentença é ambígua em relação a isso. Se for assumido que existe o ordenamento entre os dois eventos, então deve existir um tempo de referência para cada oração e a idéia de um índice para o tempo de referência cai por terra. A esse respeito, Cooper /COO 83/ por exemplo, propõe abandonar a idéia de tempo de referência e tratar a tradução do tempo verbal através de variáveis livres sobre o tempo; assim o tempo passado seria traduzido por alguma variável livre cujo valor seria restrito aos tempos passados. Nesta proposta de Cooper /COO 83/, algumas vezes os tempos verbais ficariam ligados a quantificadores temporais de advérbios de tempo, algumas vezes êles permaneceriam livres, e em outras êles seriam interpretados contextualmente (como os pronomes).

A proposta do modelo D-I é diferente pois aceita que uma oração subordinada introduza sistematicamente um novo tempo de referência. Formalmente isto é feito através do emprego do operador EM para cada oração subordinada, e assim vale

$$PAS y \& EM (t_0, PAS w) = [PAS y \& EM (t_0, PAS w)]$$

o que é relativamente próximo de uma abordagem kampiana pura. Assim sendo, a regra de tradução para uma oração relativa deve ser alterada de

$$\text{lambda } x_n [a' (x_n) \& b'],$$

como está no PTQ, para

$$\text{lambda } x_n [a' (x_n) \& (Et) (EM (t, b'))].$$

Similarmente para orações complementos de verbos como "dizer", "acreditar", etc, deve ser introduzido o operador EM no objeto proposicional, tal que a tradução do PTQ

$$\text{acreditar-que', dizer-que'}$$

deva ser alterada para

$$\text{lambda } p \text{ lambda } x (Et) [\text{acreditar-que' } (x, \text{int } EM(t, \text{ext } p))] ]$$

$$\text{lambda } p \text{ lambda } x (Et) [\text{dizer-que' } (x, \text{int } EM(t, \text{ext } p))] ].$$

Assim "A mulher que pegou o livro viu João" tem como tradução

$$PAS [ (Ex) [(Vy) [\text{mulher' } (x) \& (Et) [EM(t, PAS \text{ pegar-o-livro' } (y))]] \langle - \rangle (x = y) \& \text{ver' } (j, x)]]].$$

Outra problema referente a ordenação de tempo entre oração principal e oração subordinada é o tratamento das orações com "quando", como "Pedro chegou quando João tinha saído".

## 7.5 A noção de modalidade

### 7.5.1 O problema geral

Tradicionalmente, a modalidade tem sido quase exclusivamente associada aos modos verbais e aos verbos modais, enquanto categorias gramaticais de expressão da atitude do locutor, quer em relação ao conteúdo da elocução, quer em relação ao valor verdade do enunciado, quer em relação ao ouvinte. Porém, enquanto interação, todo enunciado apresenta certo grau de modalização. Tal modalização consiste numa modificação colocada pelo falante na predicação que ele está fazendo, em função das condições colocadas à disposição da elocução. Estas condições relacionam-se a noções como falante, ouvinte, tempo-espaco, discurso anterior, universo de referência, etc. Contudo isto foge ao escopo deste trabalho, o qual não atinge o campo da pragmática: a restrição forte aqui envolve trabalhar apenas dentro da sintaxe e da semântica das sentenças.

Dentro desta restrição séria, uma primeira aproximação, do ponto de vista lógico envolve classificar as modalidades sob tres tipos, comportando cada tipo quatro categorias:

- modalidades aléticas (funcionam ao nível do estado de coisas): necessário, contingente, possível, impossível;
- modalidades epistêmicas (de conhecimento): certeza, excluído, plausível, contestável;
- modalidades deônticas: obrigatório, interdito, permitido, facultativo.

Por outro lado, do ponto de vista da relação entre os estados de coisas e o locutor, é possível distinguir as modalidades "de re" e "de dicto". A definição ao nível da lógica destas modalidades é:

uma proposição  $p$  contendo um operador modal  $M$  exprime uma modalidade de re se o escopo de  $M$  contém uma variável individual em ocorrência livre; de outra forma,  $p$  exprime uma modalidade de dicto /HUG 68/.

Adaptando isto ao ponto de vista linguístico, uma modalidade de necessidade ou de possibilidade de dicto é sempre atribuída a uma proposição (dictum), enquanto que uma modalidade de re é atribuída a algo (res) devido a este algo ter certa propriedade.

Assim se a modalidade é de dicto, a proposição é necessariamente ou possivelmente verdadeira; se a modalidade é, ao contrário, de re, uma coisa necessariamente ou possivelmente tem certa propriedade. Um exemplo de "de re" é "Solteiro é um homem que tem a propriedade necessariamente de não ser casado"; no caso de modalidade "de dicto",  $x$  não é uma propriedade mas uma proposição necessariamente verdadeira ("É necessariamente não casado um homem que é solteiro").

A distinção de re/ de dicto é de difícil tratamento através de técnicas que usem meramente a sintaxe e a semântica. Uma saída aqui seria utilizar a sugestão de Von Wright /VON 51/ de que as propriedades sejam divididas em formais (essenciais) e materiais (contingentes): isto permitiria então eliminar as



modalidades de re em favor das modalidades de dicto. A análise do Português parece favorecer esta hipótese, pois a distinção entre propriedades formais e propriedades materiais parece relacionar-se com a distinção entre predicados que exprimem propriedades individuais e predicados que exprimem manifestações temporalmente limitadas de indivíduos. No entanto o problema fica aqui como não resolvido.

### 7.5.2 Tratamento dos modos verbais

A modalidade no Português pode ser expressa pelo modo do verbo. Assim o modo indicativo está ligado a um estado de coisas reconhecido pelo falante como necessário (ou com alto grau de possibilidade). Já o modo subjuntivo está ligado a um estado de coisas reconhecido pelo falante como possível ou contingente. O modo imperativo, por seu turno, está ligado indiretamente à modalidade alética contingente, uma vez que ele expressa uma relação entre o falante e o ouvinte num ato ilocutório diretivo; porém deve ser lembrado que este modo é frequentemente substituído pelo indicativo e mesmo pelo subjuntivo.

A substituição referida acima revela que a relação falante/ouvinte depende diretamente de modalidades deônticas: se a modalidade é obrigatória, o enunciado vai para o indicativo; se a modalidade é do tipo permitido, o enunciado vai para o subjuntivo. Esta regra é geral e pode ser implementada por um predicado Prolog que teste se o modo verbal é o indicativo e, em caso de resposta afirmativa, coloca uma marca na cláusula correspondente à sentença em questão colocando-a como obrigatória; no caso do subjuntivo, a marca aponta para permissão.

Algumas restrições sintáticas podem ser usadas para testar a corretude do uso dos modos verbais; assim o indicativo ocorre obrigatoriamente em:

- orações principais;
- orações subordinadas quando o verbo da oração principal cria um universo de referência com modalidade do tipo necessário; são exemplos: parece/juro/sinto que vais sair;
- perguntas indiretas: pergunto se vais sair.

Por seu turno, o subjuntivo aparece em orações subordinadas quando o predicador da oração principal é avaliativo ("Basta que venhas cedo"), volitivo ("Desejo que venhas"), modal ("É preciso que venhas"), epistêmico ("Acredito que venhas"). Estas condições podem ser acrescentadas ao predicado de teste.

### 7.5.3 Futuro e modalidade

Uma questão interessante diz respeito ao valor verdade de proposições contidas em enunciados declarativos que descrevem estados de coisas localizados no futuro. Uma proposta aqui é que o valor verdade seja colocado como indeterminado, no momento da elocução. Assim "Choverá amanhã" só pode receber um valor verdade no intervalo designado por "amanhã", como é discutido na seção sobre tempos verbais.

Esta idéia aceita que o futuro linguístico exprime sempre um valor modal de "não factualidade", porém deve ser esclarecido que a asserção de proposições que descrevem estados de coisas localizados no futuro decorre fundamentalmente da avaliação que o falante faz sobre a possibilidade, a necessidade e a probabilidade da ocorrência destes estados de coisas.

A asserção de tais proposições funda-se então na relação de crença (relação de natureza epistêmica) do falante para com o estado de coisas. Explicitando mais claramente esta relação de crença em relação às várias formas que assume a indicação de futuro em Português, deve ser notado que a seleção dos tempos e dos modos verbais utilizados na expressão do futuro é determinada pela modalidade na qual a proposição é assertada.

Assim para a indicação de um "futuro iminente" é utilizada em geral a construção

ir (presente indicativo) + verbo (infinitivo) + agora

ir (presente indicativo) + verbo (infinitivo) + imediatamente  
a qual expressa um alto grau de certeza do falante de que a ação (expressa pelo predicador) ocorra imediatamente após a enunciação.

Expressando um grau menor de certeza estão as construções do tipo

ir (presente indicativo) + verbo (infinitivo)

as quais não exprimem a proximidade relativa ao intervalo da elocução ("Maria vai casar agora", "Maria vai casar"). Estas últimas construções expressam a necessidade (para o falante) da ocorrência dos estados de coisas que elas descrevem.

Outra forma é a construção

não + verbo (presente indicativo) + advérbio temporal

como "João não chega hoje", a qual expressa que o falante acredita que o estado de coisas descrito está excluído ou é impossível.

A forma com presente do indicativo ("Pedro casa amanhã") é selecionada pelo falante quando este crê que a ocorrência do estado de coisas descrito é altamente provável.

A base comum destas três construções que usam o presente do indicativo é então a crença do falante de que o estado de coisas descrito é necessário (ou impossível no caso de negativa).

O uso do futuro do indicativo é em geral muito limitado no discurso, ocorrendo por exemplo numa consequência causal de um condicional real ("Se não chover, haverá seca"). Outra condição é quando o falante expressa um compromisso em realizar um ato futuro ("O diretor receberá Pedro amanhã").

O uso do presente do subjuntivo para expressão de estados de coisas futuras ocorre em construções do tipo

é + adjetivo epistêmico + que + sentença  
onde os adjetivos epistêmicos são "provável", "necessário", "duvidoso", como em "É necessário que Pedro case com Maria".  
Outro uso do presente do subjuntivo para indicar futuro ocorre na construção

verbo de ordem/pedido + que + sentença  
como em "Peço que respondas a pergunta".

## 8 A GUIA DE CONCLUSÃO: O CAMINHO DA PESQUISA

Este trabalho é um dos resultados de um projeto de investigação científica que já dura sete anos, cuja história é contada resumidamente no prefácio. Este projeto tem como nó duro a construção de uma versão computacional da Gramática de Montague, em regras livres de contexto e outras maquinarias.

O presente trabalho é composto de três partes. Inicialmente existe uma introdução geral ao problema da construção de um processador automático de linguagem natural (no caso, o Português), envolvendo um capítulo inicial de situação do problema, seguido por um capítulo sobre questões formais (aonde é argumentado em favor de uma gramática com regras livres de contexto com alguma maquinaria adicional). Depois se segue um capítulo que envolve uma introdução geral sobre a Gramática de Montague.

A segunda parte é composta por um conjunto de capítulos sobre o tratamento de problemas específicos do Português, sendo elaboradas uma sintaxe e uma semântica nos moldes do modelo proposto. Até aonde conhece o autor, este tratamento é o mais completo até hoje sobre a língua portuguesa.

O capítulo inicial aqui envolve o tratamento das sentenças (declarativas, relativas e interrogativas) junto com o tratamento da coordenação. Depois é mostrado o tratamento das locuções nominais, adjetivais, preposicionais e adverbiais. A isto se segue um capítulo sobre o tratamento da locução verbal, envolvendo a resolução do problema da passiva. Por último, existe um capítulo sobre o tratamento do verbo, no que diz respeito ao tempo, ao modo e ao aspecto. Em relação ao aspecto, existe a elaboração de um cálculo lógico sobre os predicadores; no que diz respeito ao tempo, são discutidas três propostas (F-P melhorada, I-D e D-I), sendo argumentado a favor de um modelo que usa dupla indexação. Em relação ao modo, o tratamento proposto implica na utilização de um conjunto de operadores lógicos.

Cabe colocar agora, na presente terceira parte, os aspectos que estão hoje mais próximos de uma formalização, nos moldes do modelo proposto, mas sobre os quais ainda o autor não apresenta soluções totalmente sólidas e completas. Estes aspectos envolvem o tratamento

- dos funtores naturais,
  - dos quantificadores naturais e
  - das pressuposições e dos acarretamentos das sentenças,
- e, por fim, uma avaliação final do trabalho aqui apresentado.

## 8.1 O problema dos funtores naturais

Em relação aos funtores naturais, deve ser lembrado que Cohen /COH 71/ argumenta contra Grice /GRI 75/ (originalmente publicado em 1967) que a negação da linguagem natural não pode ser identificada com o operador funcional para verdade "¬": o problema básico é que a dupla negação costuma ser vista pelos falantes do Português como uma variante enfática da negação simples, pois "Pedro não bebeu cerveja hoje" e "Pedro não bebeu cerveja hoje não" significam a mesma coisa.

Duas estratégias podem ser consideradas aqui para tornar idênticos o "não" linguístico e o "não" lógico. Uma é tratar este dialeto como tendo duas formas idênticas, mas enfaticamente diferentes, de negação: uma sendo um morfema simples e a outra sendo um par de morfemas que não podem ser adjacentes /WAL 75/. Uma segunda estratégia é não identificar "¬" com um morfema de estrutura superficial da língua portuguesa, mas com um elemento da estrutura profunda ou mesmo da representação semântica. Aí não haveria problema em mapear um elemento da estrutura profunda em um ou dois "não" na estrutura superficial. A solução aqui adotada, em coerência com o modelo defendido, deve ser a primeira, implicando isto a escrita de várias regras além das aqui mostradas.

Em relação à negação ainda, considerando a sentença "O rei da França não é careca", Russel /RUS 05/ argumenta que ela é ambígua, na medida em que não existe hoje nenhum rei da França, valendo

$$(Ex)[rei(x) \ \& \ \sim(Ey)[y \neq x \ \& \ rei(y)] \ \& \ \sim careca(x)]$$

ou

$$\sim(Ex)[rei(x) \ \& \ \sim(Ey)[y \neq x \ \& \ rei(y)] \ \& \ careca(x)]$$

implicando isso que a negação linguística ficaria como idêntica à negação da lógica bivalente.

Porém Strawson /STR 50/ argumentou contra Russel, colocando que esta sentença não é ambígua e tem uma paráfrase como "O rei da França é não-careca"; assumindo isso, Strawson propôs que a sentença em questão não seja verdadeira nem falsa se não existir um rei da França.

Alguns neo-strawsonianos, como Keenan /KEE 72/ e van Frassen /VAN 69a/, formalizaram a segunda parte da posição de Strawson em termos de uma semântica trivalente (Keenan) ou de uma semântica com um buraco de valor verdade (van Frassen). Esse tipo de semântica possui três formas de negação (N1 x, N2 x, N3 x), assim definidas:

N1 x é verdade sse x é falso, e falsa de outra forma;

N2 x é verdade sse x é falso, e falsa sse x é verdade;

N3 x é verdade sse x não é verdade, e falsa de outra forma.

O problema com esta proposta é que o Português não possui dois tipos de negação e logo deve existir só um operador de negação. Kempson /KEM 75/, quanto a isso, argumenta que a negação em linguagem natural não é ambígua, e que as abordagens russeliana e neo-strawsoniana são igualmente inadequadas.

Em aditamento a esta última observação, deve ser colocado aqui que, embora a negação em linguagem natural seja semanticamente idêntica ao "¬" do cálculo proposicional, ela possui uma propriedade que dá à sentença negativa um status diferente de sua contraparte afirmativa. Givon /GIV 75/ mostra, quanto a isto, que a negação em linguagem natural é restrita a contextos em que ela pode ocorrer: uma asserção negativa é usada em linguagem natural em contextos onde sua contraparte afirmativa já foi falada ou quando o ouvinte assume (erroneamente) a crença da verdade desta afirmativa. Isto é: uma negativa não ocorre a menos que exista a possibilidade de sua afirmativa ser verdadeira. Neste momento, isto contudo é de difícil formalização.

Em relação aos outros operadores lógicos sobre um único argumento, devem ser lembradas as seguintes definições:

valor do argumento	operador T	operador N	operador P	operador Q
1	1	0	1	0
0	0	1	1	0

sendo que, destes, só N ocorre na linguagem natural.

A exclusão do operador T ocorre devido provavelmente à máxima do modo de Grice /GRI 75/: como  $[T x \langle - \rangle x]$  e como Tx é uma expressão mais complexa do que x, o uso de Tx sobre x envolveria uma prolixidade desnecessária, violando a máxima. Por outro lado, P e Q são eliminados provavelmente pela máxima da relevância, porque no caso  $[P x \langle - \rangle P y]$  e  $[Q x \langle - \rangle Q y]$  para quaisquer sentenças x e y, tal que os argumentos desta funções seriam irrelevantes para a avaliação do valor verdade das referidas sentenças.

As funções com dois argumentos são:

A	B	C	D	E	F	G	H	I	J	K	L	M	O	V	X	argumentos	
1	1	1	0	1	0	0	1	1	0	1	0	0	0	1	0	1	1
1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	0
1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	0	1
0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0

sendo que os primeiros candidatos para conetivos com valor verdade são "e" e "ou". Porém Cohen /COH 71/ argumenta que existem diferenças entre eles e suas contrapartes lógicas. Os argumentos de Cohen envolvem o uso de sentenças condicionais: ele argumenta que a sentença

"Se o antigo rei morresse de um ataque do coração e um republicano fosse declarado rei, então Pedro ficaria contente."

pode ser verdade, mas a sentença

"Se um republicano fosse declarado rei e o antigo rei morresse de um ataque do coração, então Pedro ficaria contente."

pode ser falsa. O problema aqui é ver se a ordem temporal (ou de causalidade) dos eventos é importante (ou não) para afetar o valor verdade da sentença.

Aceitando o argumento de que o valor verdade é alterado, deve ser levado em conta contudo que isto não tem nada a ver com uma sentença com "e", mas sim com a semântica das sentenças condicionais.

Além disso, assumindo que a ordem em que os argumentos são apresentados não altera o valor verdade da sentença, o conjunto acima de 16 operadores se reduz bastante. Ainda aí, colocando que os conetivos funcionais para valor verdade sejam funções que possam pegar um conjunto de valores verdade (e não somente dois argumentos), este conjunto de possíveis argumentos fica composto pelo conjunto dos subconjuntos não vazios de  $\{0, 1\}$  e logo possui três membros:

$W: \{ \{0\}, \{1\}, \{1, 0\} \}$

e o conjunto dos conetivos é então  $w \rightarrow w$ .

Isto tudo reduz o número de conetivos para oito:

A*	D*	F*	J*	K*	O*	V*	X*	argumento
1	0	1	0	1	0	1	0	{ 1 }
1	1	0	1	0	0	1	0	{ 0, 1 }
0	1	1	0	0	0	1	1	{ 0 }

porém nenhum destes conetivos ocorre assim em linguagem natural.

Deve ser observado, quanto a isso, que existem conetivos que fazem com que, quando o valor verdade de cada argumento é falso, o valor verdade da sentença fique verdadeiro (D\*, F\*, V\*, X\*): estes conetivos devem ser afastados por razões pragmáticas. Por outro lado, a máxima da relevância elimina O\*.

Dos restantes, K\* deve corresponder ao uso do "e" e A\* ao uso do "ou". Halbash /HAL 75/, quanto a isso, fornece uma demonstração formal de que uma estrutura da forma

"nem ... nem ... nem ..."

não pode ser semanticamente associada à ocorrência do conetivo binário X, sendo que o funtor lógico E iterado poderia capturar estas construções.

Por outro lado, a questão da existência de morfemas que correspondam à disjunção exclusiva não foi ainda plenamente respondida, porque a significação da disjunção exclusiva pode, às vezes, parecer muito exclusiva. A esse respeito, Horn /HOR 72/ mostra que existe uma pressão concreta na linguagem natural contra a lexicalização independente de noções tornadas redundantes pela implicação: isto é, se falando [X a Y] regularmente, na maioria dos contextos, implica [X b Y], então não há razão para lexicalizar b. Se isto é verdade, então a disjunção exclusiva não deveria ser lexicalizada, o que parece ser verdade. Com isto, a proposta aqui é que uma expressão como "ou" não é ambígua, tendo, isto sim, uma leitura inclusiva: a leitura exclusiva que aparece em certos casos, seria explicada por referência à implicação proposta por /HOR 72/.

Além disso, reforçando esta proposta, considerando a sentença "João não é patriota ou é elegante", se o "ou" fosse ambíguo, haveriam ( $p$  = "João é patriota",  $q$  = "João é elegante") então duas interpretações:

$\sim(p \vee q)$

$\sim(p \wedge q)$

sendo a primeira é equivalente a  $[(\sim p) \& (\sim q)]$  e a segunda é equivalente a  $[((\sim p) \& (\sim q)) \vee (p \& q)]$ . Esta última, e logo

$\sim(p \wedge q)$ ,

é verdade quando  $[p \& q]$  é verdade. Assim a aceitação da dupla leitura sobre "ou" assegura que "João é patriota e é elegante" é verdadeiro, o que não pode ser aceito e logo a ambiguidade é falsificada. A proposta final aqui é então que não existe um morfema para a disjunção exclusiva, mas isto deve ser melhor estudado.

Em relação à condicional, o correspondente para "se" seria o funtor  $C$  da tabela anterior; mas deve ser observado que este funtor é uma implicação material, tal que

$Cxy$  é equivalente  $Ax Ny$  ( $x$  ou não  $y$ ).

Porém "Se Pedro tem seu joelho removido, o chute de Pedro falhará" não possui o mesmo significado que "O chute de Pedro falhará ou Pedro não terá seu joelho removido", o que dificulta a identificação de "se" com  $C$ .

Existem ainda mais argumentos contra esta igualdade propostos por Grice /GRI 75/, Strawson /STR 67/ e Stalnaker /STA 75/. Um argumento é que a negação de  $Cxy$  implica em  $x$ , mas para

"Não é verdade que se um tratado de paz for assinado, a guerra será evitada. Então um tratado de paz será assinado" isto não parece valer.

Grice /GRI 75/ propõe tratar  $N(\text{se } x \text{ então } y)$  semanticamente como  $N Cxy$ , mas pragmaticamente como  $ANx Ny$  - o que só complicaria a maquinaria e não atingiria todos os casos, como o próprio Grice observa.

Um outro argumento utiliza as expressões contrafatuais e discute a transitividade /STA 68/. Assim

"Se Reagan tivesse nascido na Rússia, Reagan seria um comunista."

"Se Reagan tivesse sido um comunista, Reagan seria um traidor."

deveriam implicar em

"Se Reagan tivesse nascido na Rússia, Reagan seria um traidor."

o que não é o caso, ficando assim estabelecido que a transitividade falha nos contrafatuais (ao contrário da implicação material, a qual não pode fazer isso).

Um outro argumento /McC 74/ coloca que a identidade entre  $C$  e "se" não fornece uma base para a explicação do porque "somente se" pode ser usado como sua conversa.

Uma proposta a ser estudada aqui é que "se" tenha um significado aproximadamente como o da locução "no evento que" ou "no caso em que". Este tipo de abordagem parece conseguir pegar o uso de "se" como implicação material e o uso de "se" nos contrafatuais (onde a implicação material não provê uma base coerente). Mas isto deve ser melhor estudado também.



## 8.2 O problema dos quantificadores naturais

Os quantificadores da lógica padrão de primeira ordem parecem inadequados para tratar as sentenças quantificadas das linguagens naturais porque existem sentenças que não podem ser simbolizadas numa lógica que é restrita aos quantificadores universal ( $\forall x$ ) e existencial ( $\exists x$ ), como é mostrado a seguir.

Em relação às limitações dos quantificadores da lógica de primeira ordem, Mostowsky /MOS 57/ propôs alguns quantificadores, matematicamente interessantes, que não são definíveis em termos dos quantificadores da lógica de primeira ordem ( $\forall$ ,  $\exists$ ), iniciando o estudo dos chamados "quantificadores generalizados". A partir daí, muitos trabalhos foram feitos, devendo ser citados os sobre quantificadores de cardinalidade /KEI 69/ e os sobre quantificadores topológicos /SGR 77/. Na avaliação do autor deste trabalho, estes quantificadores não parecem contudo muito relevantes para a linguagem natural.

No que se refere especificamente à linguagem natural, deve ser notado que as sentenças:

a) Existe um número finito de estrelas.

Nenhum coração bate um número infinito de vezes.

b) Mais de metade dos eleitores votaram em Pedro.

c) A maioria das pessoas votou no Ulisses.

são dificilmente definíveis em termos do quantificador universal ( $\forall$ ) e do quantificador existencial ( $\exists$ ).

Estes exemplos poderiam ser tratados contudo como:

a) (finito  $x$ ) [ $F(x)$ ]

b) (mais que  $1/2 x$ ) [ $F(x)$ ]

c) (maioria  $x$ ) [ $F(x)$ ].

O problema aqui é que a lógica de primeira ordem somente faz quantificação sobre objetos num conjunto  $K$  não vazio de coisas (indivíduos ou entidades), e não sobre conjuntos arbitrários de coisas ou mesmo outro tipo de objeto abstrato que não esteja em  $K$ . No caso específico de "mais de metade", uma possibilidade é expandir o domínio  $K$  de quantificação para um domínio  $K \cup A$ , onde  $A$  inclua números e funções de subconjuntos de  $K$  para números, como faz /PER 83/ por exemplo.

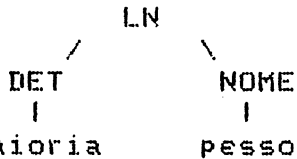
Um problema com esta proposta é o tratamento do quantificador "mais da metade", pois (ao contrário do que é possível fazer com o quantificador "a maioria") não é possível definir "mais da metade do braço de Pedro" a partir de "mais da metade de todas as coisas" /BAR 81/; isto é, o quantificador "mais da metade" não pode ser definido a partir de

mais da metade  $x$  (...  $x$  ...).

Uma abordagem diferente, e mais adequada à Teoria dos Modelos, seria realizar uma definição formal na meta-linguagem, de modo a tratar com quantificadores generalizados, sem colocar todos os problemas da Teoria dos Conjuntos na sintaxe e na semântica da lógica.

Nesta abordagem, "mais da metade" poderia ser colocado como um "determinador especial" que se combinaria com uma expressão de conjunto produzindo então um quantificador:

quantificador :- determinador\_espec, expressão de conjunto e assim [a maioria das pessoas] teria como estrutura algo como:



onde a locução nominal [a maioria das pessoas] funcionaria como um quantificador lógico. Isto resolveria o problema da dificuldade de parafrasear uma sentença como "A maioria das pessoas votou no Ulisses" como

a maioria das coisas tal que se elas são pessoas então ...

Uma outra alternativa é que os quantificadores citados seriam tais que não fossem tratados necessariamente como símbolos lógicos: quando o determinante especial "mais da metade" aplicar-se a conjuntos infinitos como em:

Mais da metade dos inteiros não são primos.

Mais da metade dos números reais entre 0 e 1, expressos em notação decimal, não começam por 7.

a verdade ou falsidade destas sentenças não dependeria a priori da lógica em si, mas sim da medida usada para conjuntos infinitos. Se assim fosse feito, então esta medida deveria fazer parte do modelo proposto.

Em relação especificamente a esta medida, um contexto deveria ser fixado para a determinação do significado das expressões básicas: isto é tradicionalmente feito em Teoria dos Modelos, mas, no caso presente, isto deveria ser feito de tal forma que este contexto provesse uma interpretação para os determinantes especiais (ou "não lógicos"). Isto teria como consequência eliminar o problema da especificação vaga destes determinantes especiais.

Assim pensando, é possível colocar que os quantificadores seriam usados para declarar que um conjunto possui alguma propriedade. Isto é:  $[(\text{Ex}) F(x)]$  declararia que o conjunto das coisas que satisfaz  $F(x)$ , informalmente

$(x \mid F(x))$

ou formalmente

$\lambda x [ F(x) ],$

é um conjunto não vazio. Por outro lado  $[(\forall x) F(x)]$  declararia que o conjunto  $K$  contém todos os indivíduos. Além disso  $[\text{Finito } x F(x)]$  declararia que o conjunto é finito. Deste modo, um quantificador seria visto como dividindo a família dos conjuntos providos pelo modelo: quando combinado com alguns conjuntos, ele produziria o valor "verdade" e quando combinado com os outros, ele produziria o valor "falso". Para capturar esta idéia, pode ser colocada a idéia simples (mas diferente do que está no PTQ) de que os quantificadores denotam a família dos conjuntos para os quais eles geram o valor "verdade".

Colocando a denotação do símbolo para quantificador  $Q$  como  $Q'$ , onde  $K$  representa o conjunto das entidades provido pelo modelo, pode ser estipulado então que:

$(E)'$  =  $\{X =C K \mid X \neq \emptyset\}$  ( $=C$  : está contido)

$(V)'$  =  $\{K\}$

$(\text{Finito})'$  =  $\{X =C K \mid X \text{ é finito}\}$

$(\text{Mais da metade de } N)'$  =  $\{X =C K \mid X \text{ contém mais da metade de } N\}$

$(\text{A maioria de } N)'$  =  $\{X =C K \mid X \text{ contém a maioria de } N\}$ .

Para enfatizar o papel do conjunto, deve ser escrito

$Q \text{ lambda } x [F(x)]$

em vez de

$Q x [F(x)]$ .

Assim "Alguma pessoa espirrou" teria como tradução:

$(\text{alguma pessoa}) \text{ lambda } x [\text{espirrou}(x)]$

e "Pedro espirrou", por seu turno, seria traduzido naturalmente como:

$\text{pedro} \text{ lambda } x [\text{espirrou}(x)]$

onde "pedro" denotaria a família dos conjuntos que contém Pedro.

Esta modificação proposta colocaria uma locução nominal como dividindo os conjuntos correspondentes às locuções verbais em duas classes: aquelas nas quais a referida locução nominal é verdadeira e aquelas onde ela é falsa. A denotação de uma locução nominal seria então o conjunto que ela faz verdadeiro.

Esta proposta apresenta contudo alguns problemas bastante sérios. Um deles é que os determinantes especiais são funções de denotações de nomes comuns (conjuntos de coisas) para denotações de locução nominal (conjunto de conjuntos), e toda função tem um domínio. O problema é que este conjunto pode não existir. Isto é importante para os determinantes "o", "ambos", "nenhum": [o homem sulfuroso] pode não denotar nada, por exemplo. Uma saída pode ser tratar a denotação do determinante "o" como uma função cujo domínio é o conjunto dos conjuntos que possui um elemento; "ambos", por seu turno, seria definido sobre conjuntos de dois elementos.

Um modo formal de pensar isso seria colocar que, para cada número natural  $n$ , onde " $\geq$ " denota "maior ou igual",

$(n)'$  (A) =  $\{X =C K \mid (X \text{ intersecção } A) \geq n\}$

$(o \ n)'$  (A) =  $\{(cada)'$  (A) se cardinalidade (A) =  $n\}$   
indefinido, de outra forma

$(\text{ambos})'$  (A) =  $(\text{os } 2)'$  (A)

e, por esta proposta, para cada determinante  $D$

$(D)'$  (A) é uma família de conjuntos  $Q$  com a propriedade que  $X$  pertence a  $Q$  sse  $(X \text{ intersecção } A)$  pertence a  $Q$ .

Isto é: se  $x$  é um membro de  $(D)'$  (A) ou não depende, só de  $[X \text{ intersecção } A]$ . Esta propriedade dos determinantes diz que eles assinalam, a um conjunto  $A$ , uma família de conjuntos que "vive" em  $A$ . Além disso, seria necessário colocar que

$(D(X))' = (D)'(X')$ .

Olhando, a partir daí, os quantificadores como relações binárias num universo K, pode ser observado se eles cumprem certas condições globais. Assim, simplificando a notação para

$Q(K) A B$

onde Q é o quantificador e

A e B são subconjuntos do conjunto universo K, uma restrição recorrente poderia valer sobre o primeiro argumento, tal que

$Q(K) A B \Leftrightarrow Q(K) A (B \text{ intersecção } A)$

aqui chamada de "conservatividade".

Esta propriedade diria por exemplo que

"alguns homens comem"  $(\rightarrow)$  "alguns homens são homens que comem"

isto é: se B é um membro de  $Q(K) (A)$  ou não depende somente de que parte de B é comum com A. Isto é proposto por /WES 85/ como um universal linguístico.

Deve ser observado que esta propriedade não vale para alguns quantificadores lógicos; assim, por exemplo, o quantificador de Resher

$\text{Re } X Y \text{ (onde } |X| > |Y|)$

não é conservativo. Mas em compensação a sentença lógica

$\text{Re } X (R_1(X), R_2(X))$

não pode ser expressa através de uma sentença da forma

sentença :- locução\_nominal, locução\_verbal.

com a locução nominal na forma

locução\_nominal :- determinante, nome.

mas somente através da forma

existe mais R1 do que R2.

O princípio da conservatividade, por outro lado, parece ser epistemologicamente válido: ele pega a idéia intuitiva de que a locução nominal  $Q(R)$  restringe o universo K à denotação

$A = R'$

do substantivo. Dizendo as coisas de outro modo: os objetos fora de A em K (isto é,  $K-A$ ) não possuem nenhuma influência se B pertence ou não a  $Q(A)$ . Se, por exemplo, fosse o caso de

$Q(K) (A) = \{X =C K \mid (A \text{ intersecção } X) = (K - (A \text{ intersecção } X))\}$

o princípio valeria, porém se fosse o caso de

X pertence ou não a  $Q(K)$  depende de  $(K-A)$

então o princípio (da conservatividade) não valeria.

O que o princípio da conservatividade diz então é que se pode restringir, sem perda, a família dos conjuntos  $Q(K) (A)$  aos subconjuntos de A. Esta idéia restrita diz que os subconjuntos de A são os mesmos, quer o universo seja K ou A.

Um outro princípio, epistemologicamente interessante, seria o da "constância", do tipo

se  $B =C A =C K$  então

B pertence a  $[Q(K) (A)] \Leftrightarrow B$  pertence a  $[Q(A) (A)]$

e combinando o princípio da conservatividade com o da constância pode ser estipulado que

se  $A =C K$  e se  $B =C K$  então

B pertence a  $[Q(K) (A)] \Leftrightarrow$

$(A \text{ intersecção } B)$  pertence a  $[Q(K) (A)]$

como propõe van Benthem /VAN 84/.

No entanto, ao princípio da conservatividade poderiam ser colocados alguns contra-exemplos. Assim algumas pessoas falantes do Português aceitam que "Algumas meninas são risonhas" possa ser falso e que "Algumas meninas são meninas risonhas" possa ser verdadeiro (testes feitos pelo autor). Outro contra-exemplo envolveria o uso de expressões intensionais: "Todos os alegados machos são fêmeas" pode ser verdade, mas "Todos os alegados machos são machos fêmeas" é uma falsidade analítica.

Outras restrições gerais parecem difíceis, à primeira vista. Uma candidata poderia envolver o segundo argumento da relação quantificada. Assim poderia ser proposto:

$$(K \neq \emptyset)$$

$$(E A) (E B) D(K) A B \quad \vee \quad (E A) (E B) \sim D(K) A B$$

aqui chamado de princípio da variedade. Os quantificadores "algum", "dois", "ambos", "a maioria" parecem satisfazer este critério, porém ele não funciona para o quantificador "ao menos dois" num universo de apenas um objeto.

Outro princípio, não necessariamente geral, pode ser o da monotonicidade /BAR B<sub>i</sub>/. Existe a monotonicidade para conjuntos que aumentam (aqui denominada (mon +)); isto é, vale:

$$Q(K) A B \quad B = C B_i$$

---


$$Q(K) A B_i$$

e os quantificadores que são mon + possuem a propriedade da estabilidade: quando  $Q(K) A B$  vale, mesmo sob uma base de conhecimento parcial sobre a denotação do predicado B, esta declaração permanece válida quando mais membros da denotação de B são conhecidos. São exemplos de mon + os quantificadores "todos", "a maioria", "alguns", "muitos", "o", "este", "ao menos dois".

Existe também a monotonicidade para conjuntos que diminuem (mon -), cumprindo a implicação:

$$Q(K) A B \quad B_i \text{ está contido em } B$$

---


$$Q(K) A B_i$$

São mon - os quantificadores "nenhum", "poucos", "ao menos dois" e não são monótonos "exatamente dois", "exatamente a metade".

Porém, se "um pouco" é usado no sentido de [alguns mas não muitos], "uns poucos homens" não é monótono; mas se é usado como [ao menos uns poucos], ele é mon +.

Pode ser observado que existem mais mon + do que mon -, sendo que estes últimos são basicamente negações de quantificadores de aumento. Uma proposta a ser estudada aqui é se a negação com Q, do tipo

$$\sim Q = \{X = C K \mid X \text{ não pertence a } Q\}$$

ou do tipo

$$Q^{\sim} = \{X = C K \mid (K-X) \text{ pertence a } Q\}$$

reverte sempre a monotonicidade.

Uma observação interessante aqui é que apenas os quantificadores complexos parecem ser não monotônicos.

Por outro lado, poderia ser considerada a aplicação da monotonicidade sobre o argumento da esquerda do quantificador, isto é:

$$\begin{array}{c} Q\langle K \rangle \quad A \quad B \quad \quad \quad A = C \quad A_i \\ \hline Q\langle K \rangle \quad A_i \quad B \end{array}$$

a qual parece valer para "alguns", mas não para "todo" e "a maioria".

Uma visão interessante seria adotar agora uma postura construtivista, fixando o universo K e fixando uma classe inicial de relações básicas de quantificação. A partir daí poderiam ser realizadas combinações booleanas e estes "geradores" mais as operações booleanas gerariam uma classe de relações de quantificação (aqui denominada Q-GEN). Um trabalho interessante seria investigar se Q-GEN cumpre o princípio da conservatividade, por exemplo.

Um problema com esta proposta é que [a maioria X Y] é conservativo, mas só pode ser definido em função de "algum"; por outro lado, "todo" só poderia ser definido em termos de uma enumeração bruta. Mas nenhuma definição existe em linguagem de primeira ordem para [a maioria], precisamente porque as definições em lógica de primeira ordem possuem uma certa "insensibilidade de limiar" ao cruzarem sobre diferentes universos (ao contrário de "a maioria").

Poderiam ser estudados além disso princípios que pegassem os quantificadores naturais com "características lógicas". Assim, estes quantificadores são insensíveis aos traços individuais dos objetos:

$$Q\langle K \rangle A B \text{ somente depende do número de indivíduos em } A, B, A \text{ intersecção } B, \text{ e } K$$

e colocando

$$\begin{array}{ll} |K| = e & |A \text{ intersecção } B| = c \\ |A \cap B| = a & |B \cap A| = b \end{array}$$

poderiam ser definidos:

$$\begin{array}{l} \text{todo: } a = \emptyset \\ \text{algum: } c \neq \emptyset \\ \text{a maioria: } c > a \\ \text{algum: } c > \text{função}(e). \end{array}$$

Outro postulado para estes quantificadores seria:

$$\begin{array}{l} \text{se } Q \text{ vale para } a, b, c, e, \\ \text{então } Q \text{ vale para } a+1, b, c, e \end{array}$$

sendo que este postulado vale para "todos", "nenhum", mas não vale para "a maioria", "a minoria", "muitos", "poucos".

Outro postulado para estes quantificadores envolveria certos padrões inferenciais. Assim a monotonicidade implica em:

$$\begin{array}{ll} Q\langle K \rangle X Y & \rightarrow Q\langle K \rangle X (Y \vee Z) \\ Q\langle K \rangle X (Y \& Z) & \rightarrow Q\langle K \rangle X Y \end{array}$$

Por outro lado, "algum" e "nenhum" são simétricos:

$$Q\langle K \rangle X Y \rightarrow Q\langle K \rangle Y X$$

e "todo" é reflexivo e transitivo. No conjunto dos quantificadores investigado até agora, não foi possível detectar nenhum quantificador assimétrico, exceto o vazio.

### 8.3 O problema da pressuposição e do acarretamento

A representação semântica de uma sentença, neste trabalho, foi colocada como a explicitação de uma forma lógica. Foi aceito que o sistema da lógica era bivalente. Agora esta bivalência vai ser posta em cheque.

Uma noção aparentemente simples é a de "acarretamento" entre sentenças. Por definição, os acarretamentos de uma sentença são compostos pelas sentenças cuja verdade se segue da verdade dessa primeira sentença. Porém, numa lógica trivalente (verdade, falso, nem verdade nem falso), é possível instituir o conceito de "pressuposição": as pressuposições de uma sentença são todas as sentenças que se seguem tanto da verdade desta sentença quanto de sua falsidade.

O debate acarretamento/pressuposição pode ser germinado aqui a partir do debate entre Russel e Strawson sobre a sentença "O rei da França é careca", debate este referido acima. O primeiro autor trabalha numa lógica bivalente, o segundo com uma lógica trivalente: se for aceita a existência desta lógica trivalente, existem pressuposições; em caso contrário, não existem diferenças entre pressuposições e acarretamentos.

Numa visão mais lógica, uma sentença S1 acarreta uma sentença S2 quando:

- se S1 é verdade, S2 é verdade;
- se S2 é falsa, S1 é falsa.

Um exemplo é "Pedro é solteiro" (S1) que acarreta "Pedro é um homem" (S2).

Porém se S1 for falsa, daí nada se segue em relação a S2: se existe uma relação de acarretamento entre S1 e S2, o valor verdade de S2 independe do valor verdade de S1 quando S1 é falsa.

A pressuposição ("S2 é uma condição para a bivalência de S1") difere do acarretamento sob dois modos: sendo falsa a consequência de S1 e sendo falsa a consequência de S2. Para que S1 pressuponha S2, a verdade de S2 deve seguir da verdade de S1, mas se S2 for falsa então S1 não tem valor verdade. O exemplo disso é exatamente "O rei da França é careca" (S1): S1 pressupõe a sentença "Há um rei da França" (S2). Por outro lado, se S2 é falsa (não existe um rei da França), então S1 não é nem verdadeira nem falsa.

As "tabelas verdade" para acarretamento e pressuposição são:

S1		S2	
V	->	V	Acarretamento
F	<-	F	
F	->	?	

S1		S2	
V	->	V	Pressuposição
?	<-	F	
F	->	V	

A argumentação original de Strawson a favor da noção de pressuposição limitou-se a locuções nominais definidas. Porém a noção de pressuposição pode ser defendida também pela discussão de verbos como "lamentar" e "entender": estas sentenças pressupõem a verdade da sentença complementar que se segue ao verbo. Assim "João lamenta que Pedro esteja preso" pressupõe a sentença "Pedro está preso". A idéia aqui é que "João lamenta que Pedro esteja preso" e "João não lamenta que Pedro esteja preso" possuem algumas implicações em comum: as duas pressupõem que Pedro está preso.

Esta questão diz respeito a como as pressuposições de uma sentença complexa são determinadas pelas pressuposições das orações que elas contêm. A esse respeito, Langendoen e Savin /LAN 71/ argumentaram que as pressuposições de uma sentença complexa poderiam ser definidas como a soma lógica das pressuposições de suas sentenças constituintes (hipótese chamada aqui de hipótese cumulativa). Porém isto não vale para sentenças como "Se Maria tem filhos, então todos os filhos de Maria são carecas", pois a oração conseqüente pressupõe que Maria tem filhos, mas a condicional como um todo não faz tal pressuposição.

Karttunen /KAR 73/ argumenta contra essa hipótese cumulativa e a favor de uma classificação, envolvendo três diferentes tipos de predicados com complemento: as tomadas (plugs), os buracos (holes) e os filtros (filters). As tomadas são predicados que bloqueiam todas as pressuposições da sentença complemento; os buracos fazem com que todas as pressuposições da sentença complemento tornem-se pressuposições da sentença principal; os filtros sob certas condições cancelam algumas das pressuposições da sentença complemento.

As tomadas envolvem os verbos de dizer (dizer, mencionar, prometer, acusar, perguntar, pedir, ordenar, criticar). São exemplo:

a) Pedro prometeu mostrar o rei da França para Antonio  
(não pressupõe que tal rei exista)

b) Maria acusa Pedro de bater em sua esposa  
(não pressupõe que Pedro tenha uma esposa)

c) Maria pediu para Pedro que ele a beijasse de novo  
(não pressupõe que ele a tenha beijado antes)

Os buracos envolvem os seguintes predicados: conhecer, surpreender, começar, parar, continuar, ser capaz, forçar, prevenir, parecer. Assim em

(a) Pedro tem uma esposa

(b) Pedro tem batido em sua esposa

(c) Pedro parou de bater em sua esposa

(d) Pedro hesitou em parar de bater em sua esposa

(e) Surpreendeu Maria que Pedro hesitou em parar de bater em sua esposa

(f) Cecília sabe que surpreendeu Maria que Pedro hesitou em parar de bater em sua esposa

deve ser observado que (b) pressupõe (a), (c) pressupõe (b), (d) não pressupõe (c), (e) pressupõe (d) e é pressuposta por (f). Assim da verdade de (f) é possível concluir que (a), (b), (d), (e) são verdade, mas nada sobre (c).



Os filtros contêm só conetivos lógicos: e, ou, se ... então. O problema aqui é distinguir o caso:

(a1) Se a calvície é hereditária, então todos os filhos de Pedro serão carecas

(a2) Se todos os filhos de Pedro são carecas, então a calvície é hereditária

do caso:

(b1) Se Pedro tem filhos, então todos os filhos de Pedro são carecas

(b2) Se todos os filhos de Pedro são carecas, então Pedro tem filhos.

Em (a1), a conseqüente "Todos os filhos de Pedro são carecas" pressupõe que Pedro tem filhos, e então a condicional funciona como um todo. Em (a2), é o antecedente que carrega esta pressuposição e toda a sentença também.

Mas, em (b1), a pressuposição do conseqüente é filtrada, porque a sentença total não pressupõe que Pedro tenha filhos; (b2) é similar a isso.

Uma condição de filtragem seria então /KAR 73/:

Seja a sentença S na forma "Se A então B"

- se A pressupõe C, então S pressupõe C;

- se B pressupõe C, então S pressupõe C a não ser que A implica semanticamente B,

onde A implica semanticamente B sse B é verdade quando A é verdade.

A condição de filtragem para "e" seria /KAR 73/:

Seja a sentença S da forma "A e B"

- se A pressupõe C, então S pressupõe C

- se B pressupõe C, então S pressupõe C a não ser que A implica semanticamente C.

A condição de filtragem para "ou" seria /KAR 73/:

Seja a sentença S da forma "A ou B"

- se A pressupõe C, então S pressupõe C

- se B pressupõe C, então S pressupõe C a não ser que não A implica semanticamente C.

Existem contudo vários exemplos que não satisfazem a solução dada para os filtros /WIL 75/, devendo aqui ser feito um melhor estudo.

Dentro do "espírito das idéias" de Montague, as pressuposições podem ser vistas como constituindo um aspecto do significado que seja distinto da "semântica do conteúdo" das sentenças /KAR 79/. Uma idéia inicial aqui é que os condicionais contrafáticos (sempre no modo subjuntivo) pressupõem a falsidade de sua oração antecedente se a sentença é verdade (tese defendida originalmente em /LAK 70/). Um exemplo é "Se estivesse chovendo, o sono de Pedro seria fácil".

Porém sentenças como "Se Maria fosse alérgica à penicilina, ela apresentaria os sintomas que ela mostra agora" tendem a sugerir que o antecedente é verdadeiro. Isto mostra que as condicionais no subjuntivo não tem como regra que sua antecedente seja falsa.

Além disso "Se Shakespeare fosse o autor de Macbeth, haveriam provas nos registros de teatro do ano de 1605" é uma sentença condicional no subjuntivo, onde a conseqüente não é verdadeira nem falsa.

Uma solução provisória (até melhor análise) aqui é que uma condicional no subjuntivo do tipo "Se A então B" implica que é epistemologicamente possível não-A.

Um modelo de análise da pressuposição aqui deve ser contudo mais explícito e formalizado dentro do "espírito das idéias" do PTQ. No PTQ, os significados são representados numa lógica intensional, onde são tratadas as condições de verdade dos significados.

Para tratar das pressuposições, seria então necessário que cada sentença em linguagem natural seja associada a duas expressões em lógica intensional: uma (chamada de extensão) seria a tradução tradicional de Montague; a outra expressaria o que a sentença implica, construindo então uma lógica a duas dimensões /KAR 73/. Dentro dessa proposta é possível então conceber quatro valores semânticos compostos:

T : sentença verdadeira e implicação verdadeira

F : sentença falsa e implicação verdadeira

t : sentença verdadeira e implicação falsa

f : sentença falsa e implicação falsa.

Assim, seja, para cada locução x, estabelecido que x' denota a extensão de x e que x'' denota a implicação de x.

Além disso, seja a sentença "Pedro ama Maria": a implicação aí introduzida pelo verbo "amar" relaciona Pedro com a extensão da locução nominal objeto direto mais do que com a locução [Maria]. Utilizando ama'' para a expressão de implicação de [ama], a denotação da implicação da locução verbal é então ama'' (int maria').

Contudo pode ser associada ao verbo "amar" uma terceira expressão ama\* em lógica intensional, a qual expressa o que a locução nominal objeto implica antes de ser adicionada o resultado da implicação do verbo. Isto é: ocorre a união de

ama\* (int maria')

mais do que maria', com a expressão

ama'' (int maria')

e de um modo formal é preciso escrever então

ama'' (int maria') & ama\* (int maria')

como a implicação de [ama Maria].

Por razões técnicas, esta expressão não está bem formada: é preciso usar a variável x para unir as duas partes da expressão e ama-maria'' é então igual a

lambda x [ama'' (x, int maria') & ama\* (x, int maria')].

Isto expressa a propriedade característica dos indivíduos que estão para Maria na relação de implicação por "amar".

Se isto fosse aceito, a regra 5 do PTQ deveria ser reescrita como:

S5: se a é um VT (verbo transitivo) e b é um T (termo no sentido do PTQ), então ab é um VI (verbo intransitivo);

T5: <a'(int b') ; lambda x [a''(x, int b') & a\*(x, int b')]>

e assim "Pedro ama Maria" teria como tradução:

```

pedro_ama_maria' = pedro'(int ama' (int maria'))
pedro_ama_maria" = [pedro" (int ama' (int maria')) &
                    pedro* (lambda x [ama" (x, int maria') &
                    ama* (x, int maria')] ) ] .

```

Com esta maquinaria, /KAR 73/ propõe para as expressões com conectivos as seguintes interpretações triviais:

```

se_a_então_b' = [a' -> b']
a_e_b' = [a' & b']
a_ou_b' = [a' v b'] .

```

No que se refere às implicações, Karttunen e Peters sugerem a existência de um conjunto S de hipóteses que assegura para as implicações de "se a então b" o seguinte:

(1) S assegura que as implicações de a são verdadeiras e  
(2) S U (a) assegura que as implicações de b são verdadeiras, onde a primeira condição requer que a sentença condicional implique tudo que a oração antecedente implica e a segunda condição estabelece que quando a e cada sentença em b são verdadeiras, então todas as implicações de b são verdadeiras.

Porém como a é verdade só no caso em que a' é verdadeira, e como as implicações de b adicionam b", a segunda condição diz no fundo que S U (a') implica b". Dadas as propriedades lógicas do acarretamento e da condicional material, isto é equivalente a dizer que S acarreta [a' -> b"] . Isto tudo pode então ser generalizado para:

```

S acarreta se_a_então_b' só quando
(1) S acarreta a' e
(2) S acarreta [a' -> b"] que vale só quando
    S acarreta [a" & [a' -> b"]].

```

Isto significa então que a implicação de se\_a\_então\_b vale [a" & [a' -> b"]]. E adicionando a esta implicação a possibilidade epistêmica aceita antes para as condicionais subjuntivas, vale então

```

se_a_então_b" = ["K "a' & a" & [a' -> b"]].

```

onde K é o operador epistêmico "conhecer".

As regras para conjunção e disjunção podem ser:

```

a_e_b" = [a" & [a' -> b"]].
a_ou_b" = [ [a" v b'] & [a' v b"] ]

```

e a regra tipo PTQ que gera as orações com "se" pode ser expressa por:

Sxx: se a é uma conjunção subordinativa (do tipo ((t/t)/t)) e se b é uma sentença (do tipo t), então ab é uma sentença conjuntiva t/t (ela precisa se unir a outra sentença t para gerar t);

Txx: (a'(int b') ; lambda q [a' (int b') & a\* (int b")] (q)).

Aplicando isso às pressuposições das sentenças, é possível sugerir as seguintes fórmulas (onde p[a] é o conjunto total das pressuposições de a):

```

p[a -> b] = ["b -> p[a]] & [a -> p[b]]
p[a & b] = [b -> p[a]] & [a -> p[b]]
p[a v b] = ["b -> p[a]] & ["a -> p[b]]

```

Outra linha de trabalho diz respeito ao tratamento das implicações de uma sentença a partir de uma "escala quantitativa" entre estas implicações /GAZ 79/. A idéia chave aqui é que, dada uma sentença  $a$ , é possível construir uma função  $f$  com argumento  $a$  que retorne o conjunto de implicações quantitativas escalares como seu valor:

$$f(a) = \{x \mid x = K \sim li\}$$

onde  $K$  é a abreviatura de "o falante sabe que" e  $li$  são as implicações onde  $i$  pertence aos naturais, sendo que

$$K a \langle - \rangle "P \sim a$$

onde  $P$  é a abreviatura de "todo falante sabe que", como faz Hintikka /HIN 62/.

Como avaliação final desta discussão, pode ser dito que, embora não exista um completo acordo sobre a definição de pressuposição, um padrão emerge: as pressuposições parecem associadas com itens lexicais particulares e com certas construções gramaticais /KAR 73/. Isto sugere que um processador e um léxico podem prever as pressuposições de uma sentença, usando uma computação puramente estrutural.

Por outro lado, alguns exemplos de acarretamentos foram também associados com itens lexicais /GIY 73/. Isto também pode ser gerado enquanto uma sentença está sendo processada.

Então, nos limites deste trabalho, os conceitos de "pressuposição" e de "acarretamento" podem ser referidos aos exemplos que aparecem associados a itens lexicais particulares e a construções sintáticas especiais. Não são aqui levadas em conta noções como texto prévio, regras culturais, leis universais, conhecimento do falante - atribuíveis ao campo da pragmática: esta limitação sempre foi aqui assumida. Contudo o método aqui proposto assume que o objetivo de um processador de linguagem natural seja o de traduzir as sentenças do Português para representações semânticas em forma lógica (do tipo do PTQ) e que as pressuposições e os acarretamentos de uma sentença sejam subfórmulas derivadas desta sentença. Enquanto técnica, esta proposta assume que as pressuposições e os acarretamentos sejam computados enquanto ocorre o processamento da referida sentença.

Um problema que pode ser levantado contra esta técnica inferencial é o de como guiar o processo de modo a só realizar inferências "importantes" e não todas as inferências. A resposta que pode ser dada é que não está ainda resolvido o problema de como só gerar as inferências "importantes" /CHA 76/.

No que se refere às pressuposições baseadas em construções sintáticas, a proposta assume que quando existe ambiguidade (uma sentença tem duas estruturas sintáticas), o processador não resolve esta ambiguidade, gerando ambas as leituras.

Um exemplo sintático bem claro aí é formado pelas locuções nominais definidas, as quais pressupõem que algum indivíduo (pessoa ou coisa) exista. Assim quando uma locução nominal tipo "o cão grande" é processada, deve ser inferido que

$$(E x) (x = \text{cão grande}).$$

Uma análise semelhante pode ser proposta para as implicações de uma oração relativa ligada a uma locução nominal: esta construção pressupõe que exista algum indivíduo (pessoa ou coisa) que possua a propriedade mencionada na oração relativa. Assim quando "o homem que estava aqui ..." é processada, é inferido que

(E x) (x tem a propriedade de "estava aqui ...")  
devendo ser aqui recordado que esta idéia é usada na formulação, proposta neste trabalho, da interpretação semântica das locuções nominais com oração relativa.

Em relação às pressuposições ligadas a itens lexicais, um exemplo é a utilização da idéia de que cada palavra pode ser classificada como fazendo parte de uma categoria, onde cada categoria é definida por um traço semântico. Assim se a palavra "Pedro" possui como traço "humano", então a sentença "Pedro ama Maria" tem como pressuposição "Um humano ama Maria", por exemplo. Por outro lado "humano" pode ser visto como uma especificação de "ser vivo" e, como "Pedro ama Maria" pressupõe também que "Um ser vivo ama Maria", as supercategorias dos traços semânticos devem também ser usadas. Uma forma simples de fazer isso funcionar é estabelecer regras de implicação no processamento.

Esta idéia (de utilizar os traços ligados às palavras que compõem a sentença em processamento para gerar as pressuposições dela) também pode ser usada para o tratamento das pressuposições de sentenças compostas por orações: os verbos, por exemplo, com o traço plug bloqueiam as pressuposições da sentença complemento, como foi discutido acima.

Um outro exemplo de pressuposição ligada ao léxico diz respeito ao tratamento da negação: a proposta aqui, assumindo a discussão feita acima, é que numa sentença do tipo

locução\_nominal não locução\_verbal  
a locução nominal sujeito é assumida como existente (a negação é só interna à sentença), mas na forma em que a locução verbal desdobra-se num verbo seguido por uma outra sentença ("Maria não veio quando João fez aniversário") deveria ser assumido que as pressuposições da sentença encaixada são verdadeiras.

Outro problema diz respeito ao tempo verbal de uma pressuposição, pois "João parou de bater em Maria às dez horas" pressupõe que "João batia em Maria até imediatamente antes das dez horas". A hipótese forte aqui é que o tempo da pressuposição é completamente determinado pelo tempo da sentença em análise, mas isto requer maior estudo.

## 8.4 Uma avaliação

Numa avaliação do que aqui é apresentado, deve ser observado que a maquinaria proposta não possui nenhuma faceta ad hoc, sendo que de um lado não é nova (a noção de traço, por exemplo) e de outro, sob certos propósitos, é nova (por exemplo, a noção de categoria derivada no tratamento das extracções de locução).

Em relação ao nó duro do modelo proposto, uma idéia importante é a de tomar nodos como símbolos complexos (com conjuntos de traços) ou termos lógicos, sendo isso usado em todas as regras propostas. Outra idéia, mais original, do modelo envolve a interpretação das regras sintáticas a partir da noção de "admissibilidade": uma regra admite um conjunto de árvores locais (onde uma árvore local é o nó pai e seus filhos) e um árvore inteira é tida como bem formada só no caso onde suas sub-árvores serem bem formadas. Esta idéia não é usada na gramática proposta, pois envolve um processamento a base de "daemons" - idéia viável num Prolog em processamento paralelo.

Uma outra idéia aqui pouco desenvolvida envolve a noção de meta-regra. Como deve ter ficado claro na discussão feita nos capítulos anteriores (em especial o do tratamento das construções verbais com verbos auxiliares), não está ainda estabelecido se esta maquinaria pode ser totalmente substituída por regras de natureza lexical, onde os itens lexicais são vistos como termos envolvendo traços.

Outra área de investigação futura diz respeito ao tratamento do controle da concordância, via traços, para as locuções verbais encaixadas em construções tipo "equi" e "raising", no sentido de fazer estas locuções verbais concordarem sintaticamente com o seu sujeito. O tratamento desenvolvido neste trabalho envolve adotar uma versão lexicalista para este controle (o fato de que "Maria" é o sujeito de "viver" em "João persuadiu Maria a viver" depende das regras associadas com "persuadir"). Isto envolveu também utilizar um princípio de que a locução nominal que entra nesta relação de implicação de controle, é sempre vista como o próximo argumento semântico do verbo, isto é, "persuadir" é analisado informalmente como

`<loc_verbal', <loc_nominal', loc_nominal'>>`,  
devendo ter ficado claro que muitas outras regras aqui devem ser escritas e testadas, para uma maior abrangência de sentenças do Português.

Uma outra peça central do modelo aqui proposto envolve o tratamento da dependência não ligada, sem o uso de transformações: esta contribuição é talvez uma das mais relevantes contribuições isoladas deste trabalho. Aqui é mostrado que o fenômeno da extração à distância é compatível com o modelo com regras livres de contexto e que, além disso, existem (como é discutido) um conjunto de fatos que são problemáticos para a abordagem transformacional e que são tratáveis facilmente pelo modelo proposto. Um exemplo onde esta análise é superior à proposta transformacional diz respeito ao tratamento das orações relativas e das perguntas; outra área de aplicação é a topicalização.

Uma aplicação também importante da noção de dependência não ligada é o tratamento das estruturas coordenadas, onde a idéia básica é que só categorias idênticas podem ser coordenadas.

Uma inovação a ser melhor estudada no futuro envolve uma aproximação maior com a proposta das Gramáticas Categrais, implicando o estudo da necessidade da composição de funções do tipo "se A/B e se B/C então A/C"; falando em termos de regras, isto pode ser lido como "se existe a regra  $X \rightarrow W V$  e se existe a regra  $V \rightarrow Y Z$ , então existe a regra  $X/Y \rightarrow W Z$ ", implicando isso na possibilidade da existência da regra abaixo

$v/n(\text{masc, sing, ...}) :- v(\dots), v(\text{masc, sing, ...})$ .

No que diz respeito à abordagem da interpretação semântica, o modelo apresentado aqui coloca uma inovação realmente importante, no sentido de um tratamento formal (nos moldes do PTQ), sem a maquinaria ad hoc utilizada na maioria dos processadores de linguagem natural. A implementação desta semântica é bastante fácil em Prolog, desde que exista definido um predicado "lambda" do tipo

$\text{lambda}(\text{variável, expressão, valor\_da\_variável, resultado})$ .

Uma área bastante desenvolvida aqui é o tratamento do verbo, no que diz respeito ao tempo, ao aspecto e ao modo. O tratamento do tempo em processador de linguagem natural é algo inovador e um campo onde este trabalho muito contribuiu; no que diz respeito ao cálculo lógico (aspectual) sobre os predicadores, aqui apresentado, ele é extremamente importante no sentido de aumentar a capacidade de inferência do processador, o mesmo se aplicando ao tratamento do modo e à discussão sobre pressuposições e acarretamentos.

Por último, o autor quer deixar aqui registrado que desconhece qualquer trabalho sobre processadores automáticos de língua portuguesa que possua a extensão do que aqui é apresentado. O mesmo pode dizer o autor sobre os trabalhos em Linguística Formal sobre o Português. Porém o modelo aqui desenvolvido não deve ser tomado como uma exposição definitiva de uma teoria da sintaxe e da semântica do Português, mas como uma contribuição dentro de um programa continuado de pesquisas, programa esse que procura explorar as possibilidades de uma teoria restrita e computacional da sintaxe acoplada a uma teoria explícita e computacional da semântica.

"Life is short, science is long."  
/HIP 76/

9 BIBLIOGRAFIA

- /AHO 68/ AHO, A. Indexed grammars - an extension of context free grammars. *Journal of the ACM*, 15:647-71, 1968.
- /AHO 68a/ AHO, A. and ULLMAN, J. Translations on a context-free grammar. *Proceedings of the Association Computing Machine Symposium on Theory of Computing*, May, 1968.
- /AHO 72/ AHO, A. and ULLMAN, J. The theory of parsing, translation, and compiling. Englewood Cliffs, Prentice Hall, 1972.
- /AJD 67/ AJDUKIEWICZ, K. Syntactic connexion. In: McCALL, S. *Polish logic*. Oxford, Clarendon Press, 1967. p. 207-31.
- /AJD 74/ AJDUKIEWICZ, K. *Pragmatic logic*. Dordrecht, Reidel, 1974.
- /AKM 79/ AKMAJIAN, A.; STEELE, S. and WASOW, T. The category AUX in universal grammar. *Linguistic Inquiry*, 10:1-61, 1979.
- /AND 73/ ANDERSON, J. *An essay concerning aspects*. The Hague, Mouton, 1973.
- /AND 77/ ANDERSON, J. Induction of augmented transition networks. *Cognitive Sciences*, 1:125-57, 1977.
- /ARB 68/ ARBIB, M. and GIVE'ON, Y. Algebra automata I: Parallel programming as a prolegomena to the categorial approach. *Information and Control*, 12:331-45, 1968.
- /BAC 80/ BACH, E. In defense of passive. *Linguistics and Philosophy*, 3:290-341, 1980.
- /BAR 64/ BAR-HILLEL, Y. *Language and information*. Reading, Addison Wesley, 1964.
- /BAR 65/ BAR-HILLEL, Y.; PERLES, M. and SHAMIR, E. On formal properties of simple phrase structure grammars. In: LUCE, R. et alii. eds. *Readings in Mathematical Psychology*. New York, Wiley, 1965. p. 75-105.
- /BAR 81/ BARWISE, J. AND COOPER, R. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159-219, 1981.



- /BAT 78/ BATES, M. The theory and practice of augmented transition networks grammars. In: BOLC, L. ed. *Natural language communication with computers*. Berlin, Springer Verlag, 1978. p. 191-259.
- /BEA 79/ BEAR, J. and KARTTUNEN, L. PSG: a simple phrase structure parser. *Texas Linguistic Forum*, 15:1-46, 1979.
- /BEC 66/ BECHARA, E. *Moderna gramática portuguesa*. 10 ed. São Paulo, Editora Nacional, 1966.
- /BEN 78/ BENNETT, M. and PARTEE, B. *Toward the logic of tense and aspect in English*. Bloomington, Indiana University Linguistic Club, 1978.
- /BEN 79/ BENNETT, M. *Questions in Montague grammar*. Bloomington, Indiana University Linguistic Club, 1979.
- /BIC 75/ BICKERTON, D. Creolization, linguistic universals, natural semantax and the brain. Paper presented at the International Conference on Pidgins and Creoles, University of Hawaii, Honolulu, 1975. (mimeo).
- /BIG 78/ BIGELOW, J. Believing in semantics. *Linguistics and Philosophy*, 2:101-45, 1978.
- /BIG 78a/ BIGELOW, J. Semantics on thinking, saying and translation. In: GUENTHNER-REUTTER, M. and GUENTHNER, F. eds. *Meaning and translation: Philosophical and linguistic approaches*. London, Duckworth, 1978.
- /BIR 79/ BIRNBAUM, L. and SELFRIDGE, M. *Problems in conceptual analysis of natural language*. Department of Computer Science, Yale University, 1979.
- /BLA 81/ BLACKWELL, S. Processing conjunctions in an ATN parser. University of Cambridge, Department of Philosophy, 1981.
- /BOB 69/ BOBROW, D. and FRASER, J. An augmented state transition network analysis. *Proceedings of the First International Joint Conference on Artificial Intelligence*, 1969. p. 557-67.
- /BOB 85/ BOBROW, D. and HAYES, P. Artificial Intelligence: where are we? *Artificial Intelligence*, 25:375-415, 1985.

- /BOG 83/ BOGURAEV, Recognising conjunctions with the ATN framework. In: JONES, K. and WILKS, Y. eds. Automatic natural language parsing. Chichester, Ellis Horwood, 1983. p. 39-45.
- /BOO 73/ BOOK, R. Topics in formal language theory. In: AHO, A. ed. Currents in the theory of computing. Englewood Cliffs, Prentice Hall, 1973.
- /BOR 83/ BORGIDA, A. Some formal results about stratificational grammars and their relevance to Linguistics. Mathematical Systems Theory, 16:29-56, 1983.
- /BRA 68/ BRAINERD, W. The minimalization of tree automata. Information and Control, 13:484-91, 1968.
- /BRA 76/ BRAME, M. Conjectures and refutations in syntax and semantics. Amsterdam, North-Holland, 1976.
- /BRA 80/ BRACHMAN, R. and SMITH, B. Introduction. SIGABI Newsletter, 70:1-7, 1980. (Special issue on knowledge representation).
- /BRE 73/ BRESNAN, J. Syntax of the comparative clause construction in English. Linguistic Inquiry, 4:275-343, 1973.
- /BRE 78/ BRESNAN, J. A realistic transformational grammar. In: HALLE, M.; BRESNAN, J. and MILLER, G. eds. Linguistic theory and psychological reality. Cambridge, MIT Press, 1978. p. 1-59.
- /BRI 83/ BRISCOE, E. Determinism and its implementation in PARSIFAL. In: JONES, K. and WILKS, Y. eds. Automatic natural language parsing. New York, Ellis Horwood, 1983. p. 61-8.
- /BUE 68/ BUENO, F. Gramática normativa da língua portuguesa. 7 ed. São Paulo, Saraiva, 1968.
- /BUN 78/ BUNNEMAN, P. and LEVY, L. Proper analysis algorithmics. Proceedings of the 1978 Conference on Information Sciences and Systems. Baltimore, Johns Hopkins University, 1978.
- /CAM 64/ CAMARA JUNIOR, J. Dicionário de filologia e gramática. 2 ed. Rio de Janeiro, Ozon, 1964.
- /CAM 79/ CAMARA JUNIOR, J. História e estrutura da língua portuguesa. Rio, Padrão, 1979.
- /CAR 47/ CARNAP, R. Meaning and necessity. Chicago, University of Chicago Press, 1947.

- /CAR 77/ CARLSON, G. A unified analysis of the English bare plural. *Linguistics and Philosophy*, 1(3):413-56, 1977.
- /CAR 83/ CARDEN, G. The non-finite-state-ness of the word formation component. *Linguistic Inquiry*, 14:537-41, 1983.
- /CHA 76/ CHARNIAK, E. Inference and knowledge. In: CHARNIAK, E. and WILKS, Y. eds. *Computational semantics*. Amsterdam, North-Holland, 1976. p. 1-21.
- /CHA 83/ CHARNIAK, E. A parsing with something for everyone. In: KING, M. ed. *Parsing natural language*. New York, Academic Press, 1983. p. 117-49.
- /CHO 57/ CHOMSKY, N. *Syntactic structures*. The Hague, Mouton, 1957.
- /CHO 62/ CHOMSKY, N. A transformational approach to syntax. *Third Texas Conference on Problems of Linguistic Analysis in English*, 1962. p. 138-44.
- /CHO 63/ CHOMSKY, N. Formal properties of grammars. In: LUCE, R.; BUSH, R. and GALANTER, E. eds. *Handbook of Mathematical Psychology*. volume 2. New York, Wiley, 1963. p. 323-418.
- /CHO 65/ CHOMSKY, N. *Aspects of the theory of syntax*. Massachusetts, MIT Press, 1965.
- /CHO 72/ CHOMSKY, N. *Studies on semantics in generative grammar*. The Hague, Mouton, 1972.
- /CHO 75/ CHOMSKY, N. *Questions on form and interpretation*. New York, Elsevier, 1975.
- /CHO 75a/ CHOMSKY, N. *The logical structure of linguistic theory*. New York, Plenum, 1975.
- /CHO 77/ CHOMSKY, N. *Essays on form and interpretation*. New York, Elsevier, 1977.
- /CHO 77a/ CHOMSKY, N. On wh-movement. In: CULICOVER, P.; WASOW, T. and AKMAJIAN, A. *Formal syntax*. New York, Academic Press, 1977. p. 71-132.
- /CHO 77b/ CHOMSKY, N. and LASNIK, H. Filters and control. *Linguistic Inquiry*, 8:425-504, 1977.
- /CHO 80/ CHOMSKY, N. *Rules and representations*. New York, Columbia University Press, 1980.
- /CHO 82/ CHOMSKY, N. *Lectures on government and binding*. Dordrecht, Foris, 1982.

- /CHU 40/ CHURCH, A. A formulation of a simple theory of types. *Journal of Symbolic Logic*, 5:56-68, 1940.
- /CHU 51/ CHURCH, A. A formulation of the logic of sense and denotation. In: HENLE, P.; KALLEN, H. and LANGER, S. eds. *Structure, meaning and method*. New York, Liberal Arts, 1951. p. 3-24.
- /CHU 80/ CHURCH, K. *On memory limitations in natural language processing*. Bloomington, Indiana University Linguistics Club, 1980.
- /CHU 83/ CHURCH, K. *Phrase structure parsing: a method for taking advantage of alloebonic constraints*. Bloomington, Indiana University Linguistic Club, 1983.
- /CLA 77/ CLARK, H. and CLARK, E. *Psychology and language: An introduction to psycholinguistics*. San Francisco, Freeman, 1977.
- /COC 65/ COCCHIARELLA, N. *Tense and modal logic: A study in the typology of temporal reference*. San Diego, UCLA, 1965. (mimeo).
- /COH 71/ COHEN, L. The logical particular of natural language. In: BAR-HILLEL, Y. ed. *Pragmatics of natural languages*. Dordrecht, Reidel, 1971. p. 50-68.
- /COL 70/ COLMERAUER, A. *Les systemes ou un formalisme pour analyser et synthetiser des phrases sur ordinateur*. Université de Montréal, Département d'Informatique, 1970.
- /COL 75/ COLMERAUER, A. *Les grammaires de metamorphose*. Groupe d'Intelligence Artificielle, Université d'Aix-Marseille-Luminy, 1975.
- /COL 79/ COLMERAUER, A. Un sous-ensemble intéressant du français. *BAIRD Informatique Théorique*, 13(4):309-36, 1979.
- /COM 76/ COMRIE, B. *Aspect. An introduction to the study of verbal aspect and related problems*. Cambridge, Cambridge University Press, 1976.
- /CON 63/ CONWAY, N. Design of a separable transition-diagram compiler. *Communications of the ACM*, 6/7:396-408, 1963.
- /COO 75/ COOPER, R. *Montague's semantics framework and transformational syntax*. Amherst, University of Massachusetts, 1975. (mimeo)

- /COO 76/ COOPER, R. and PARSON, T. Montague grammar, generative semantics and interpretative semantics. In: PARTEE, B. ed. *Montague grammar*. New York, Academic Press, 1976. p. 311-62.
- /COO 79/ COOPER, R. *Model theory for a fragment of English*. Madison, University of Wisconsin, 1979.
- /COO 83/ COOPER, R. *Quantification and syntactic theory*. Dordrecht, Riedel, 1983.
- /CRE 73/ CRESWELL, M. *Logics and languages*. London, Methuen, 1973.
- /CRE 74/ CRESWELL, Interval semantics for some event expressions. 1974. (mimeo).
- /CRE 75/ CRESWELL, M. Hyperintensional logic. *Studia Logica*, 34:261-92, 1975.
- /CRE 76/ CRESWELL, M. Review of formal philosophy, selected papers of Richard Montague, ed. with an introduction by Richmond Thomason, New Haven: Yale University Press, 1976. *Philosophia*, 6:193-207, 1976.
- /CUL 76/ CULICOVER, P. *Syntax*. New York, Academic Press, 1976.
- /CUL 85/ CULY, C. The complexity of the vocabulary of Bambara. *Linguistics and Philosophy*, 8:345-51, 1985.
- /CUN 79/ CUNHA, C. *Gramática da língua portuguesa*. Rio de Janeiro, Fename, 1979.
- /DIA 59/ DIAS, A. *Syntaxe histórica portuguesa*. 4. ed. Lisboa, Clássica, 1959.
- /DON 74/ DONELLAN, K. Speaking of nothing. *Philosophical Review*, 83:3-32, 1974
- /DOW 78/ DOWTY, D. Governed transformations as lexical rules in a Montague Grammar. *Linguistic Inquiry*, 3:393-426, 1978.
- /DOW 81/ DOWTY, D.; WALL, R. and PETERS, S. *Introduction to Montague Semantics*. Dordrecht, Reidel, 1981.
- /DOW 82/ DOWTY, D. More on the categorial analysis of grammatical relations. In: ZAENEN, A. ed. *Subjects and other subjects*. Proceedings of the Harvard Conference on the representation of grammatical relations, 1981. Bloomington, Indiana University Linguistic Club, 1982. p. 115-53.

- /DRE 76/ DRESHER, E. and HORSTEIN, N. On some supposed contributions of Artificial Intelligence to the scientific study of language. *Cognition*, 4:321-98, 1976.
- /DRE 77/ DRESHER, E. and HORSTEIN, N. Reply to Winograd. *Cognition*, 5:377-92, 1977.
- /EAR 70/ EARLEY, J. An efficient context-free parsing algorithm. *Communications of the ACM*, 13:94-102, 1970.
- /EJE 80/ EJERHED, E. A context free phrase-structure parsing for English auxiliaries. Paper presented to the Fourth Groningen Round Table Conference on Auxiliaries, Groningen, Holland, July 1980. (mimeo).
- /ELS 78/ ELSTER, J. *Logic and society: contradictions and possible worlds*. New York, Wiley, 1978.
- /EMO 76/ EMONDS, J. *A transformational approach to English syntax*. New York, Academic Press, 1976.
- /ENG 82/ ENGBAHL, E. A note on the use of lambda-conversion in generalized phrase structure grammars. *Linguistics and Philosophy*, 4:505-15, 1982.
- /EVA 85/ EVANS, R. PROGRAM: a development tool for GPSG grammars. *Linguistics*, 23:213-43, 1985.
- /FIL 75/ FILLMORE, C. *Lectures on deixis*. Indiana University Linguistic Club, 1975.
- /FOD 83/ FODOR, J. Phrase structure parsing and the island constraints. *Linguistics and Philosophy*, 6:163-223, 1983.
- /FRE 52/ FREGE, G. On sense and reference. In: GEACH, P. and BLACK, M. eds. *Translations from the philosophical writings of Gottlob Frege*. Oxford, Blackwell, 1952. p. 56-78.
- /FRI 52/ FRIES, C. *The structure of English: an introduction to the construction of English sentences*. New York, Harcourt, 1952.
- /FRI 74/ FRIEDRICH, P. *On aspect theory and homeric aspect*. Chicago, University of Chicago Press, 1974.
- /FRI 78/ FRIEDMAN, J. and WARREN, D. A parsing method for Montague Grammars. *Linguistics and Philosophy*, 2:347-72, 1978.

- /FUB 75/ FU, K. and BOOTH, T. Grammatical inference: introduction and survey. IEEE Transactions on Systems Man and Cybernetics, 5:95-111, 409-23, 1975.
- /FUN 83/ FUNK, W. On a semantic typology of conditional sentences. Paper presented at the 14 SLE Annual Meeting, Poznan, 1983. (mimeo).
- /GAL 75/ GALLIN, D. Intensional and higher-order modal logic with applications to Montague semantics. Amsterdam, North-Holland, 1975.
- /GAZ 79/ GAZDAR, G. A solution to the projection problem. In: OH, C. and DINNEEN, D. eds. Presupposition. New York, Academic Press, 1979. p. 57-89.
- /GAZ 82/ GAZDAR, G. Phrase structure grammar. In: JACOBSON, P. and PULLUM, G. eds. The nature of syntactic representation. Dordrecht, Reidel, 1982. p. 131-86.
- /GIN 69/ GINSBURG, S. A mathematical model of transformational grammars. Information and Control, 15:297-334, 1969.
- /GIV 71/ GIVON, T. Historical syntax and synchronic morphology: An archaeologist's field trip. QLS 7, Chicago, Chicago Linguistics Society, 1971.
- /GIV 72/ GIVON, T. Forward implications, backward presuppositions and time axis verbs. In: KIMBALL, J. ed. Linguistic Symposia. London, Seminar, 1972.
- /GIV 73/ GIVON, T. The time-axis phenomenon. Language, 49:890-925, 1973.
- /GIV 75/ GIVON, T. Negation in language: Pragmatics, function and ontology. Pragmatics Microfiche, V2:A2-G14, 1975.
- /GIV 82/ GIVON, T. Tense-aspect-modality: The creole prototype and beyond. In: HOOPER, P. ed. Tense-aspect between semantics and pragmatics. Amsterdam, Benjamins, 1982. p. 115-63.
- /GOE 17/ GOES, C. Metodo de analise. 2 ed. Belo Horizonte, Beltrão, 1917.
- /GOL 67/ GOLD, E. Language identification in the limit. Information and Control, 10:441-74, 1967.
- /GRI 75/ GRICE, P. Logic and conversation. In: COLE, P. and MORGAN, J. eds. Syntax and semantics 3: Speech acts. New York, Academic Press, 1975. p. 41-58.

- /GUN 80/ GUNJI, T. and SONDEIMER, N. The mutual relevance of model-theoretic semantics and artificial intelligence. *SHIL: Journal of Linguistic Calculus*, 3:5-42, 1980.
- /HAL 75/ HALBASH, K. An observation on English truth-functions. *Analysis*, 35:109-110, 1975.
- /HAM 73/ HAMBLIN, C. Questions in Montague Grammar. *Foundations of Language*, 10(1):41-53, 1973.
- /HAR 51/ HARRIS, Z. *Methods in structural linguistics*. Chicago, University of Chicago Press, 1951.
- /HAR 62/ HARRIS, Z. *String analysis of sentence structure*. The Hague, Mouton, 1962.
- /HAR 63/ HARMAN, G. Generative grammars without transformation rules. A defense of phrase structure. *Language*, 39:597-616, 1963.
- /HEN 50/ HENKIN, L. Completeness in theory of types. *Journal of Symbolic Logic*, 15:81-91, 1950.
- /HIN 62/ HINTIKKA, J. *Knowledge and belief*. Ithaca, Cornell University Press, 1962.
- /HIN 77/ HINTIKKA, J. The semantics of questions and the questions of semantics, 1977. (mimeo).
- /HIP 76/ HIPOCRATES. *Corpus Hippocraticus*. New York, Allwood, 1976.
- /HIR 76/ HIRSCHBUHLER, P. Relativized  $\bar{A}$ -over- $\bar{A}$  and sister condition. Amherst, University of Massachusetts, 1976.
- /HJE 53/ HJELMSLEV, L. *Prolegomena to the theory of language*. Bloomington, Indiana University Press, 1953.
- /HOB 78/ HOBBS, J. and ROSENSCHEIN, S. Making computational sense of Montague's intensional logic. *Artificial Intelligence*, 9:287-306, 1978.
- /HOO 79/ HOOOPER, P. Some observations on the typology of focus and aspect in narrative language. *Studies in Language*, 3(1):37-64, 1979.
- /HOO 79a/ HOOOPER, P. Aspect and foregrounding in discourse. In: GIVON, T. ed. *Discourse and syntax*. New York, Academic Press, 1979. p. 213-41.
- /HOO 82/ HOOOPER, P. ed. *Tense-aspect: Between semantics and pragmatics*. Amsterdam, Benjamins, 1982.



- /HOP 69/ HOPCROFT, J. and ULLMAN, J. Formal languages and their relation to automata. Reading, Addison-Wesley, 1969.
- /HOP 79/ HOPCROFT, J. AND ULLMAN, J. Introduction to automata theory, languages, and computation. Reading, Addison-Wesley, 1979.
- /HOR 72/ HORN, L. On the semantic properties of logical operations in English. Los Angeles, University of California, 1972.
- /HUG 68/ HUGHES, G. and CRESWELL, M. An introduction to modal logic. London, Methuen, 1968.
- /HUY 76/ HUYBREGTS, M. Overlapping dependencies in Dutch. Utrecht Working Papers in Linguistics, 1:24-65, 1976.
- /JAC 69/ JACKENDOFF, R. Speculations on pre-sentences and determiners. Bloomington, Indiana University Linguistic Club, 1969.
- /JAC 75/ JACKENDOFF, R. Morphological and semantic regularities in the lexicon. Language, 51(3):639-71, 1975.
- /JAC 77/ JACKENDOFF, R. X-syntax: a study of phrase structure. Cambridge, MIT Press, 1977.
- /KAK 71/ JAKOBSON, R. Selected writings. Mouton, The Hague, 1971.
- /JAN 77/ JANSSEN, T. Simulation of Montague Grammar. Annals of System Research, 6:127-40, 1977.
- /JAN 77a/ JANSSEN, T.; KOK, G. and MEERTENS, L. On restrictions on transformational grammars reducing the generative power. Linguistics and Philosophy, 1:111-8, 1977.
- /JOH 70/ JOHNSON, C. On the formal properties of phonological rules. Doctoral dissertation. University of California, Santa Barbara, 1970.
- /JOS 77/ JOSHI, A. and LEVY, L. Constraints on structural descriptions: Local transformations. SIAM Journal of Computing, 6:272-84, 1977.
- /JOS 80/ JOSHI, A.; LEVY, L. and YUEH, K. Local constraints in programming languages. Part I: Syntax. Theoretical Computer Science, 12:265-90, 1980.

- /JOS 82/ JOSHI, A. How much context-sensitivity is required, if any, for assigning reasonable structural descriptions? Paper presented at the Conference on Syntactic Theory and How People Parse Sentences, Ohio State University, May 14-15, 1982.
- /JOS 82a/ JOSHI, A. and LEVY, L. Phrase structure trees bear more fruit than you would have thought. *American Journal of Computational Linguistics*, 8(1):1-11, 1982.
- /JOS 85/ JOSHI, A. Tree adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In: DOWTY, D., KARTTUNEN, L. and ZWICKY, A. eds. *Natural language parsing. Psychological, computational, and theoretical perspectives*. Cambridge, Cambridge University Press, 1985. p. 206-50.
- /JUC 58/ JUCA FILHO, C. 132 restrições ao anteprojeto de simplificação e unificação da nomenclatura gramatical brasileira. Rio de Janeiro, Civilização Brasileira, 1958.
- /KAM 68/ KAMP, J. Tense logic and the theory of linear order. Doctoral dissertation. Los Angeles, University of California, 1968.
- /KAM 71/ KAMP, J. Formal properties of "now". *Iheoria*, 37:227-73, 1971.
- /KAM 75/ KAMP, J. Two theories about adjectives. In: KEENAN, E. ed. *Formal semantics of natural language*. Cambridge, Cambridge University Press, 1975. p. 123-55.
- /KAP 64/ KAPLAN, D. *Foundations of intensional logic*. San Diego, UCLA, 1964.
- /KAP 70/ KAPLAN, D. What is Russel's theory of descriptions? In: YOURGRAV, W. and BLACK, A. eds. *Physics, Logic and History*. New York, Plenum, 1970. p. 277-88.
- /KAP 81/ KAPLAN, R. and KAY, M. Phonological rules and finite state transducers. New York, Winter, 1981.
- /KAP 82/ KAPLAN, R. and BRESNAN, J. Lexical-functional grammar: A formal system for grammatical representation. In: BRESNAN, J. ed. *The mental representation of grammatical relations*. Cambridge, MIT Press, 1982. p. 173-281.
- /KAR 73/ KARTTUNEN, L. Presuppositions of compound sentences. *Linguistic Inquiry*, 4:169-93, 1973.

- /KAR 77/ KARTTUNEN, L. Syntax and semantics of questions. *Linguistics and Philosophy*, 1:33-44, 1977.
- /KAR 79/ KARTTUNEN, L. and PETERS, S. Conventional implicature. In: OH, C. ed. *Presupposition*. New York, Academic Press, 1978. p. 1-56.
- /KAT 64/ KATZ, J. and POSTAL, P. *An integrated theory of linguistic descriptions*. Cambridge, MIT Press, 1964.
- /KAT 72/ KATZ, J. *Semantic theory*. New York, Harper, 1972.
- /KAT 77/ KATZ, J. *Propositional structure and illocutionary force*. New York, Crowell, 1977.
- /KAT 81/ KATO, M. Restrições à regra da elipse verbal. *Ensaios de Linguística*, 5:93-101, 1981.
- /KAY 73/ KAY, M. The MIND system. In RUSTIN, R. ed. *Natural language processing*. New York, Algorithmics Press, 1973. p. 155-88.
- /KEE 72/ KEENAN, E. On semantically based grammar. *Linguistic Inquiry*, 3:413-62, 1972.
- /KEE 80/ KEENAN, E. Passive is phrasal (non sentencial or lexical). In: HOEKSTRA, T.; HULST, T. and MOORTGAT, M. eds. *Lexical grammar*. Dordrecht, Foris, 1980. p. 181-213.
- /KEI 69/ KEISLER, H. Logic with the quantifier 'There exist uncountable many'. *Annals of Mathematical Logic*, 1:1-93, 1961.
- /KEM 75/ KEMPSON, R. *Presuppositions and the delimitation of semantics*. Cambridge, Cambridge University Press, 1975.
- /KEN 63/ KENNY, A. *Actions, emotion, and will*. New York, Humanities Press, 1963.
- /KIL 84/ KILBURY, J. *GPSQ-based parsing and generation*. Berlin, Technische Universität Berlin, 1984.
- /KLE 80/ KLEIN, E. A semantics for positive and comparative adjectives. *Linguistics and Philosophy*, 4:1-25, 1980.
- /KOS 83/ KOSKENNIEMI, K. *Two-level morphology: A general computational model for word-form recognition and production*. University of Helsinki, Department of General Linguistics, Publication 11, 1983.

- /KOW 79/ KOWALSKI, R. Logic for problem solving. New York, North-Holland, 1979.
- /KRI 63/ KRIPKE, S. Semantical considerations on modal logic. Acta Philosophica Fennica, 16:83-9, 1963.
- /KRI 72/ KRIPKE, S. Naming and necessity. In: DAVIDSON, D. and HARMAN, G. eds. Semantics of natural language. Dordrecht, Reidel, 1972. p. 253-355.
- /KRI 77/ KRIPKE, S. Speaker's reference and semantic reference. In: FRENCH, P. et alii. eds. Midwest Studies in Philosophy, volume III: Studies in the Philosophy of Language. Marris, The University of Minnesota, 1977. p. 255-76.
- /KUN 72/ KUNO, S. Natural explanations for some syntactic universals. Report NSF-28, Cambridge, Harvard University Computation Laboratory, 1972.
- /LAD 71/ LADEFOGED, F. Preliminaires to linguistic phonetics. Chicago, University of Chicago Press, 1971.
- /LAD 77/ LADUSAW, W. Some problems with tense in PTQ. Texas Linguistic Forum, 6:89-102, 1977.
- /LAK 65/ LAKOFF, G. On the nature of syntactic irregularity. Doctoral dissertation. Bloomington, Indiana University, 1965.
- /LAK 70/ LAKOFF, G. Linguistics and natural logic. Synthese, 22:151-271, 1970.
- /LAK 71/ LAKOFF, G. Passive resistance. Papers from the 2th Regional Meeting, Chicago Linguistic Society, 1971. p. 149-63.
- /LAK 72/ LAKOFF, G. Linguistic and natural logic. In: DAVIDSON, D. and HARMAN, G. eds. Semantics for natural language. Dordrecht, Reidel, 1972. p. 545-665.
- /LAN 67/ LANGENDOEN, D. and SAVIN, H. The projection problem for presuppositions. In: FILLMORE, C. and LANGENDOEN, D. eds. Studies in linguistic semantics. New York, Holt, 1971. p. 55-60.
- /LAN 77/ LANGENDOEN, D. On the inadequacy of type-2 and type-3 grammars for human languages. In: HOOPER, P. ed. Studies in descriptive and historical linguistics. Amsterdam, John Benjamin, 1977. p. 159-71.
- /LAN 81/ LANGENDOEN, D. The generative capacity of word-formation components. Linguistic Inquiry, 12:320-2, 1981.

- /LAF 77/ LAPOINTE, S. Recursiveness and deletion. *Linguistic Analysis*, 3(3):227-65, 1977.
- /LAF 80/ LAPOINTE, S. A lexical analysis of the English verb system. In: HOEKSTRA, T.; van der HULST, H. and MOORTGAT, M. eds. *Lexical grammars*. Dordrecht, Foris, 1980. p. 215-54.
- /LAS 79/ LASNIK, H. and KUPIN, J. A restrictive theory of transformational grammar. Cambridge, Massachusetts Institute of Technology, 1979.
- /LAS 81/ LASNIK, H. Learnability, restrictiveness, and the evaluation metric. In: BAKER, C. and McCARTHY, J. eds. *The logical problem of language acquisition*. Cambridge, MIT Press, 1981. p. 1-21.
- /LEM 84/ LEMLE, M. *Análise sintática*. São Paulo, Ática, 1984.
- /LEV 74/ LEVELT, W. *Formal grammars in Linguistics and Psycholinguistics*. The Hague, Mouton, 1974.
- /LEW 68/ LEWIS, D. Counterpart theory and quantified modal logic. *Journal of Philosophy*, 65:113-26, 1968.
- /LEW 70/ LEWIS, D. General semantics. *Synthese*, 22:18-67, 1970.
- /LIT 82/ LI, C.; THOMPSON, S. and THOMPSON, R. The discourse motivation for the perfect aspect. In: HOOPER, P. ed. *Tense-aspect: between semantics and pragmatics*. Amsterdam, Benjamin, 1982. p. 19-44.
- /LIM 64/ LIMA, C. *Gramática normativa da língua portuguesa*. 10 ed. Rio de Janeiro, Briguiet, 1964.
- /MAL 82/ MALING, J. and ZAENEN, A. A phrase structure account for Scandinavian extraction phenomena. In: JACOBSON, P. and FULLUM, P. eds. *The nature of syntactic representation*. Dordrecht, Reidel, 1982. p. 229-82.
- /MAR 80/ MARCUS, S. *A theory of syntactic recognition for natural language*. Cambridge, MIT Press, 1980.
- /MAT 68/ MATES, B. Leibniz on possible worlds. In: van ROOTSELAAR, B. and STAAL, J. eds. *Logic, methodology and philosophy of science III*. Amsterdam, North-Holland, 1968.
- /MAT 83/ MATHEUS, M. et alii. *Gramática da língua portuguesa*. Coimbra, Almedina, 1983.
- /McC 68/ McCRAWLEY, J. Lexical insertion in a transformational grammar without deep structure. *QLS*, 4:71-80, 1968.

- /McC 68a/ McCRAWLEY, J. Concerning the base component of a transformational grammar. *Foundations of Language*, 4:243-69, 1968.
- /McC 71/ McCRAWLEY, J. Pre-lexical syntax. In: O'BRIEN, R. ed. *Report of the 22nd Roundtable Meeting on Linguistics and Language Studies*. Georgetown, Georgetown University Press, 1971.
- /McC 74/ McCRAWLEY, J. If and only if. *Linguistic Inquiry*, 5:632-5, 1974.
- /McC 78/ McCOARD, R. *The English perfect: tense choice and pragmatic inferences*. Amsterdam, North-Holland, 1978.
- /McC 79/ McCLOSKEY, J. *Transformational syntax and model theoretical semantics*. Dordrecht, Reidel, 1979.
- /MEL 68/ MELO, G. *Gramática fundamental da língua portuguesa*. Rio de Janeiro, Acadêmica, 1968.
- /MEZ 67/ MEZEI, J. and WRIGHT, J. Algebraic automata and context-free sets. *Information and Control*, 11:3-29, 1967.
- /MON 60/ MONTAGUE, R. On the nature of certain philosophical entities. *The Monist*, 53:159-94, 1960.
- /MON 68/ MONTAGUE, R. Pragmatics. In: KLIBANSKY, F. ed. *Contemporary philosophy: a survey*. Florence, La Nuova Italia Editrice, 1968. p. 102-22.
- /MON 70/ MONTAGUE, R. English as a formal language. In: VISENTINI, B. et alii. eds. *Linguaggi nella società e nella tecnica*. Milan, Edizioni di Comunità, 1970. p. 189-224.
- /MON 70a/ MONTAGUE, R. Pragmatics and intensional logic. *Synthese*, 22:68-94, 1970.
- /MON 73/ MONTAGUE, R. The proper treatment of quantification in ordinary English. In: HINTIKKA, J.; MORAVCSIK, J. and SUPPES, P. eds. *Approaches to natural language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*. Dordrecht, Reidel, 1973. p. 221-42.
- /MOS 57/ MOSTOWSKI, A. On a generalization of quantifiers. *Foundations of Mathematics*, 44:12-36, 1957.
- /MOS 81/ MOSCA, P. How to do things (real conversations) with speech acts: questions and answers around a Brazilian kitchen. *Encontros Linguísticos da PUC/RS*, 1981. (mimeo).

- /MOS 82/ MOSCA, P. Uma teoria de "atos de fala" para perguntas e respostas. Revista do Instituto de Filosofia e Ciências Humanas, UFRGS, 10:117-30, 1982.
- /MOS 83/ MOSCA, P. O outro igual é igual? E o outro do outro é o mesmo? Modelos lógico-matemáticos e computacionais para as estratégias de referência verbal a uma coleção de objetos empregadas por crianças pequenas. Porto Alegre, Dissertação de Mestrado em Linguística, PUC/RS, 1983.
- /MOS 84/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. 1. Sintaxe. 1.0: Apresentação. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84a/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. 1. Sintaxe. 1.1: Definições. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84b/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. 1. Sintaxe. 1.2: Argumentos para recusa de uma gramática livre de contexto. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84c/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. 1. Sintaxe. 1.3: Transformações I. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84d/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. 1. Sintaxe. 1.4: Problemas com a Teoria Standard de Chomsky. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84e/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. 1. Sintaxe. 1.5: Transformações II. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84f/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. 1. Sintaxe. 1.6: Uma modificação das regras categoriais de base: a sintaxe X-barra. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.

- /MOS 84g/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. I. Sintaxe. 1.2: Teoria Standard Extendida I: Eiltras e controles. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84h/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. I. Sintaxe. 1.8: Teoria Standard Extendida II: Ligações. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84i/ MOSCA, P. Problemas linguísticos no processamento de linguagem natural. I. Sintaxe. 1.8: Teoria Standard Extendida II: Governo e ligações. Porto Alegre, Grupo de Inteligência Artificial, Universidade Federal do Rio Grande do Sul, 1984.
- /MOS 84j/ MOSCA, P. Gramáticas transformacionais. Anais do I Simpósio Brasileiro de Inteligência Artificial. Porto Alegre, Universidade Federal do Rio Grande do Sul, 1984. p. 20 (resumo).
- /MOS 85/ MOSCA, P. Uma gramática livre de contexto para línguas naturais? Anais do II Simpósio Brasileiro de Inteligência Artificial. São José dos Campos, Instituto de Pesquisas Espaciais e Sociedade Brasileira de Computação, 1985. p. 151-2 (resumo).
- /MOS 86/ MOSCA, P. Uma gramática livre de contexto para construções com auxiliares. Anais do III Simpósio Brasileiro de Inteligência Artificial. Rio de Janeiro, Instituto Militar de Engenharia do Ministério do Exército e Sociedade Brasileira de Computação, 1986. p. 187-93.
- /MOS 87/ MOSCA, P. O homem e a máquina. In: COSTA, A. ed. Seminário de Epistemologia da Inteligência Artificial. 1986: Máquina e Inteligência. Porto Alegre, Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, 1987. p. 48-53.
- /MOS 87a/ MOSCA, P. A questão do sistema representacional no homem. In: COSTA, A. ed. Seminário de Epistemologia da Inteligência Artificial. 1986: Máquina e Inteligência. Porto Alegre, Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, 1987. p. 65-74.
- /PAR 73/ PARTEE, B. The syntax and semantics of quotation. In: ANDERSON, S. and KIPARSKY, P. eds. A festschrift for Morris Halle. New York, Holt, Rinehart and Winston, 1973.



- /PAR 73a/ PARTEE, B. Some structural analogies between tenses and pronouns in English. *Journal of Philosophy*, 70:601-9, 1973.
- /PEL 80/ PELLETIER, F. The generative power of rule ordering in formal grammars. *Linguistics*, 18:17-72, 1980.
- /PER 09/ PEREIRA, E. *Gramática expositiva: Curso Superior*. 2 ed. São Paulo, Duprat, 1909.
- /PER 78/ PERLMUTTER, D. Empirical evidence distinguishing some current approaches to syntax. Paper presented at the Winter Meeting of the LSA, Boston, 1978. (mimeo).
- /PER 80/ PEREIRA, F. and WARREN, D. Definite clause grammars for language analysis - A survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13:231-78, 1980
- /PER 83/ PEREIRA, E. *Logic for natural language analysis*. Menlo Park, SRI International, 1983.
- /PET 65/ PETRICK, S. A recognition procedure for transformational grammars. Doctoral dissertation. Massachusetts, MIT, 1965.
- /PET 69/ PETERS, S. and RITCHIE, R. Context-sensitive immediate constituent analysis. *Proceedings of ACM Symposium on Theory of Computing*, 1969.
- /PET 71/ PETERS, S. and RITCHIE, R. On restricting the base component of transformational grammars. *Information and Control*, 18:483-501, 1971.
- /PET 73/ PETERS, S. and RITCHIE, R. On the generative power of transformational grammars. *Information Sciences*, 6:49-83, 1973.
- /PET 73a/ PETERS, S. and RITCHIE, R. Non-filtering and local filtering transformational grammars. In: HINTIKKA, J. et alii. eds. *Formal languages*. Reidel, North-Holland, 1973.
- /PET 81/ PETERS, S. Comments on Lasnik (1981). In: BAKER, C. and McCARTHY, J. eds. *The logical problem of language acquisition*. Cambridge, MIT Press, 1981. p. 22-9.
- /PIB 81/ PIMENTA-BUENO, M. Os participios passados portugueses: verbos, adjetivos ou uma terceira classe? VI Encontro Nacional de Linguística, Rio de Janeiro, PUC/RJ, 1981. (mimeo).

- /PIN 79/ PINKER, S. Formal models of language learning. *Cognition*, 7:217-83, 1979.
- /PIN 82/ PINKER, S. A theory of the acquisition of lexical interpretative grammars. In: BRESNAN, J. ed. *The mental representation of grammatical relations*. Cambridge, MIT Press, 1982. p. 655-726.
- /PLA 73/ PLATH, W. Transformational grammar and transformational parsing in the REQUEST system. In ZAMPOLI, A. and CALZOLARI, N. eds. *Computational and mathematical linguistics*. vol 2. Florence, Olschki, 1973. p. 367-84.
- /POS 21/ POST, E. Introduction to a general theory of elementary propositions. *American Journal of Mathematics*, 43:163-85, 1921.
- /POS 64/ POSTAL, P. Limitations of phrase structure grammars. In: FODOR, J. and KATZ, J. eds. *The structure of language: Readings in the Philosophy of Language*. Englewood Cliffs, Prentice Hall, 1964. p. 137-51.
- /POW 78/ POWER, R. and LONGUET-HIGGINS, H. Learning to count: a computational model of language acquisition. *Proceedings of the Royal Society of London B*, 200:391-417, 1978.
- /PRI 55/ PRIOR, A. and PRIOR, M. Deontic logic. *Philosophical Review*, 64:43-59, 1955.
- /PUL 82/ PULLUM, G. and GAZDAR, G. Natural languages and context free languages. *Linguistics and Philosophy*, 4:471-504, 1982.
- /PUT 61/ PUTNAM, H. Some issues in the theory of grammar. In: JAKOBSON, R. ed. *The structure of language and its mathematical aspects*. Providence, American Mathematical Society, 1961.
- /PUT 73/ PUTNAM, H. The meaning of "meaning". In: GUNDERSON, K. ed. *Language, mind, and knowledge*. Minneapolis, University of Minnesota Press, 1973.
- /QUI 53/ QUINE, W. Reference and modality. In: QUINE, W. ed. *From a logic point of view*. New York, Harper and Row, 1953. p. 139-53.
- /QUI 60/ QUINE, W. *Word and object*. Cambridge, MIT Press, 1960.
- /RAB 67/ RABIN, M. Mathematical theory of automata. *Proceedings of Symposia of Applied Mathematics XIX*. Providence, American Mathematical Society, 1967. p. 173-5.

- /RAD 81/ RADFORD, A. Transformational syntax. Cambridge, Cambridge University Press, 1981.
- /REI 47/ REICHEMBACH, H. Elements of symbolic logic. Berkeley, University of California Press, 1947.
- /REI 71/ REIS, F. Grammatica portugueza. Maranhão, Almeida, 1871.
- /RIB 26/ RIBEIRO, J. Grammatica portuguesa: Curso Médio. 37 ed. Rio de Janeiro, Acadêmica, 1926.
- /RIB 85/ RIBEIRO, J. Grammatica portugueza. 2 ed. São Paulo, Teixeira, 1885.
- /RIE 75/ RIESBECK, C. Conceptual analysis. In: SCHANK, R. ed. Conceptual information processing. Amsterdam, North-Holland, 1975. p. 83-156.
- /ROB 82/ ROBINSON, J. DIAGRAM: a grammar for dialogs. Communications of the ACM, 25:27-47, 1982.
- /ROS 67/ ROSS, J. Constraints on variables in syntax. Bloomington, Indiana. University Linguistic Club, 1967.
- /ROS 69/ ROSS, J. Guess who? In: BINNICK, R. et al. eds. Papers from the Fifth Regional Meeting of the Chicago Linguistic Society. Chicago, Chicago Linguistic Society, 1969.
- /ROS 69a/ ROSS, J. Auxiliares as main verbs. In: TODD, W. ed. Studies in philosophical linguistics 1. Evanston, Great Press, 1969. p. 77-102.
- /ROU 75/ ROUSSEL, P. Erolon: manuel de reference et d'utilisation. Groupe d'Intelligence Artificielle, Université de Marseille-Luminy, 1975.
- /ROO 81/ ROOT, R. SMX: a program for translating English into Montague's intensional logic. Austin, Department of Linguistics, University of Texas, 1981.
- /ROU 70/ ROUNDS, W. Mappings and grammars on trees. Mathematical System Theory, 4:257-87, 1970.
- /RYL 49/ RYLE, G. The concept of mind. London, Hutchinson, 1949.
- /RUS 03/ RUSSEL, B. Principles of mathematics. London, Allen and Urwin, 1903.
- /RUS 05/ RUSSEL, B. On denoting. Mind, 14:479-93, 1905.

- /SAI 57/ SAID ALI, M. *Dificuldades da língua portuguesa*. Rio de Janeiro, Acadêmica, 1957.
- /SAI 63/ SAID ALI, M. *Gramática histórica da língua portuguesa*. 5 ed. São Paulo, Melhoramentos, 1963.
- /SAD 74/ SADOCK, J. *Toward a linguistic theory of speech acts*. New York, Academic Press, 1974.
- /SAM 79/ SAMPSON, G. What was transformational grammar? *Lingua*, 48:355-78, 1979.
- /SCH 72/ SCHIFFER, S. *Meaning*. London, Oxford University Press, 1972.
- /SCH 75/ SCHANK, R. *Conceptual information processing*. Amsterdam, North-Holland, 1975.
- /SCH 77/ SCHANK, R. and WILENSKY, R. Response to Drescher and Hornstein. *Cognition*, 5:133-45, 1977.
- /SCH 82/ SCHUBERT, L. An approach to the syntax and semantics of affixes in 'conventionalized' phrase structure grammar. 1982. (mimeo).
- /SCH 82a/ SCHUBERT, L. and PELLETIER, J. From English to logic: Context-free computation of 'conventional' logical translation. *American Journal of Computational Linguistics*, 8(1):26-44, 1982.
- /SCO 70/ SCOTT, D. Advice on modal logic. In: LAMBERT, K. ed. *Philosophical problems in logic*. Dordrecht, Reidel, 1970. p. 143-74.
- /SGR 77/ SGRO, J. Completeness theorems for topological models. *Annals of Mathematical Logic*, 11:173-93, 1977.
- /SHI 76/ SHIBATANI, M. The grammar of causative constructions: A conspectus. In: SHIBATANI, M. ed. *The grammar of causative constructions*. New York, Academic Press, 1976. p. 1-42.
- /SMI 75/ SMITH, C. The analysis of tense in English. *Texas Linguistic Forum*, 1:71-89, 1975.
- /STA 68/ STALNAKER, R. A theory of conditionals. In: RESCHER, I. ed. *Studies in Logical Theory*. Washington, Georgetown University Press, 1968. p. 98-112.
- /STA 70/ STALNAKER, R. and THOMASON, R. A semantic analysis of conditional logic. *Theoria*, 34:23-42, 1970.
- /STA 75/ STALNAKER, R. Indicative conditionals. *Philosophia*, 5:269-86, 1975.

- /STR 50/ STRAWSON, P. On referring. *Mind*, 59:320-44, 1950.
- /STR 67/ STRAWSON, P. *If and  $\Rightarrow$* . Oxford, Oxford University, 1967.
- /TAR 44/ TARSKI, A. The semantic conception of truth. *Philosophy and Phenomenological Research*, 4:341-75, 1944.
- /TAY 77/ TAYLOR, B. Tense and continuity. *Linguistics and Philosophy*, 1(2):199-220, 1977.
- /THA 67/ THATCHER, W. Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. *Journal of Computing System Science*, 1:317-22, 1967.
- /THA 68/ THATCHER, W. and WRIGHT, J. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical System Theory*, 2:57-81, 1968.
- /THA 69/ THATCHER, J. Transformations and translations from the point of view of generalized finite automata theory. *Proceedings of the Association Computing Machine Conference on Theory of Computing*, May, 1969.
- /THA 73/ THATCHER, J. Tree automata: an informal survey. In: AHO, A. ed. *Currents in the theory of computing*. Englewood Cliffs, Prentice Hall, 1973.
- /THO 68/ THORNE, J.; BRATLEY, P. and DEWAR, H. The syntactic analysis of English by machine. In: MICHIE, D. ed. *Machine intelligence 3*. Edinburgh, Edinburgh University Press, 1968. p. 281-301.
- /THO 74/ THOMASON, R. Introduction. In: THOMASON, R. ed. *Formal philosophy: Selected papers by Richard Montague*. New Haven, Yale University Press, 1974. p. 1-69.
- /VAL 75/ VALIANT, L. General context-free recognition in less than cubic time. *Journal of Computer and System Sciences*, 10:308-15, 1975.
- /VAN 69/ van WIJNGGARDEN, A. Report on the algorithmic language ALGOL 68. *Numerische Mathematik*, 14:79-218, 1969.
- /VAN 69a/ VAN FRASSEN, B. Presuppositions, supervaluations and free logic. In: LAMBERT, K. ed. *The logical way of doing things*. New Haven, Yale University Press, 1969. p. 67-92.

- /VEN 67/ VENDLER, Z. *Linguistics in philosophy*. Ithaca, Cornell University Press, 1967.
- /VIE 78/ VIEIRA, M. *A voz passiva num tratamento lexicalista*. Dissertação de Mestrado. Rio de Janeiro, UFRJ, 1978.
- /VON 51/ von WRIGHT, G. *An essay in modal logic*. Amsterdam, North-Holland, 1951.
- /VON 63/ von WRIGHT, G. *Norm and action*. New York, Humanities Press, 1963.
- /WAL 75/ WALKER, R. Conversational implicatures. In: BLACKBURN, S. ed. *Meaning, reference and necessity*. Cambridge, Cambridge University Press, 1975. p. 133-81.
- /WAS 77/ WASOW, T. Transformations and the lexicon. In: CULICOVER, P. et alii. eds. *Formal syntax*. New York, Academic Press, 1977. p. 327-60.
- /WAS 78/ WASOW, T. On contracting the class of transformational languages. *Synthese*, 39:81-104, 1978.
- /WES 85/ WESTHERTAHL, D. Logical constraints in quantifier languages. *Linguistics and Philosophy*, 8:387-413, 1985.
- /WEX 81/ WEXLER, K. and CULICOVER, P. *Formal principles of language acquisition*. Cambridge, MIT Press, 1981.
- /WIL 75/ WILSON, D. *Presupposition and non-truth-conditional semantics*. New York, Academic Press, 1975.
- /WIL 83/ WILKS, Y. and JONES, K. Introduction: a little light history. In: JONES, K. and WILKS, Y. eds. *Automatic natural language parsing*. New York, Ellis Horwood, 1983. p. 11-31.
- /WIN 72/ WINOGRAD, T. *Understanding natural language*. Edinburgh, Edinburgh University Press, 1972.
- /WIN 77/ WINOGRAD, T. On some contested suppositions of Generative Linguistics about the scientific study of language. *Cognition*, 5:151-97, 1977.
- /WOL 80/ WOLFF, G. Language acquisition and the discovery of phrase structure. *Language and Speech*, 23:255-69, 1980.
- /WOO 70/ WOODS, W. Transition network grammars for natural language analysis. *Communications of the ACM*, 13:591-606, 1970.

- /WOO 72/ WOODS, W., KAPLAN, R. and NASH-WEBER, B. The LUNAR sciences natural language information system. Cambridge, Bolt, Beranek and Newman, 1972.
- /WOO 73/ WOODS, W. An experimental parsing system for transition network grammars. In: RUSTIN, R. ed. Natural language processing. New York, Algorithmics Press, 1973. p. 111-54.
- /ZWI 65/ ZWICKY, A. et alii. The MITRE analysis procedure for transformational grammar. Proceedings of the 1965 Fall Joint Computer Conference, 1965. (mimeo).
- /ZWI 78/ ZWICKY, A. Arguing for constituents. QLS, 14:503-12, 1978.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

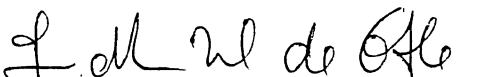
Pós-Graduação em Ciência da Computação

Programação em lógica, Montague e regras livres de contexto:

Teoria formalizada em lógica de um processador para a Língua Portuguesa

Dissertação apresentada aos Srs.

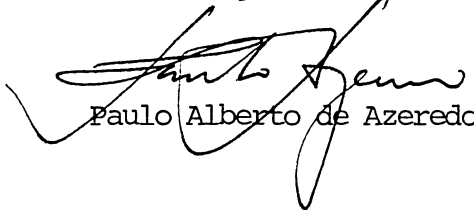
  
Antonio Carlos da Rocha Costa

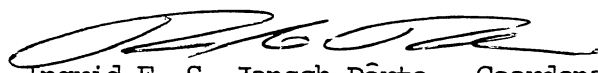
  
José Mauro Volkmer de Castilho

  
Sueli Mendes

Visto e permitida a impressão

Porto Alegre, 14/02/90

  
Paulo Alberto de Azeredo - Orientador

  
Ingrid E. S. Jansch Pôrto - Coordenadora do Curso de  
Pós-Graduação em Ciência da Computação