UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Usando bases de dados relacionais para geração semi-automática de ontologias destinadas à extração de dados

por

ORLANDO MIGUEL VIVAN

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

Prof. Dr. Carlos Alberto Heuser
Orientador

Porto Alegre, fevereiro de 2003

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Vivan, Orlando Miguel

Usando bases de dados relacionais para geração semi-automática de Ontologias destinadas à extração de dados/ por Orlando Miguel Vivan. – Porto Alegre : PPGC da UFRGS, 2003.
79p.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR – RS, 2003. Orientado: Heuser, Carlos Alberto.

1. Extração de dados. 2. Construção de ontologias. 3. Extração semântica. I. Heuser, Carlos Alberto. II. Título.

"A grandeza não consiste em receber honras,
mas em merecê-las".
Aristóteles

Agradecimentos

Em primeiro lugar, agradeço a Deus por estar sempre guiando e iluminando meus passos.

A Carla, minha querida esposa, que me apoiou, me deu forças e muitas vezes fez papel de pai e mãe para nossos filhos nesse período. Te amo muito.

A meu filho Gabriel que, apesar de seus 7 anos, procurou entender minha ausência em vários momentos de sua vida. A minha pequena Rafaela que, ainda sem entender, me dá motivação para enfrentar desafios como este. Eu amo vocês.

A minha amiga e mãe, Iris, que me ensinou que vencer é muito mais do que conquistar títulos ou bens materiais, é ter fé no que acreditamos.

Ao professor Carlos Alberto Heuser, meu orientador, que, com experiência e palavras objetivas, enriqueceu grandemente este trabalho.

Aos colegas do grupo de pesquisa do professor Heuser, principalmente a Carina, Vanessa, Rodrigo, Renata, Eduardo, Sérgio e todos aqueles com quem eu tive mais contato e que sempre me ajudaram durante minhas estadas em Porto Alegre.

Ao amigo e professor Nelson Zanete com quem discuti idéias novas para o trabalho. Também a todos os outros professores e amigos da Faculdade Paranaense – Faccar, departamente de informática, pela motivação recebida.

Ao professor e colega Fernando Accorsi pelas explicações recebidas na área de Teoria da Computação. Também ao professor Celso Kaestner, da Pontifícia Universidade Católica do Paraná em Curitiba, pela bibliografia recomendada na área de reconhecimento de padrões.

À Sercomtel, pelo incentivo financeiro e por permitir dedicação ao estudo em períodos de trabalho. Aos que nesse período foram meus coordenadores diretos, Fernando, Roberto e Sebastião, permitindo que me dedicasse aos estudos. À colega Silvana que me substituiu em tarefas da empresa que não podiam deixar de ser realizadas.

A todos os professores do mestrado com quem tivemos contato e principalmente à coordenação do curso de mestrado remoto, professores Mário Proença e José Palazzo Moreira de Oliveira, que viabilizaram a realização do curso entre UEL e UFRGS.

Aos funcionários do PPGC e biblioteca da Universidade Federal do Rio Grande do Sul, mostando-se sempre muito atenciosos e prestativos, principalmente para nós estudantes remotos.

Sumário

Lista de Abreviaturas

AFD	Autômato Finito Determinístico
AFN	Autômato Finito não Determinístico
AFε	Autômato Finito com Movimentos Vazios
DEG	Data Extraction Group
ER	Entidade Relacionamento
ExR	Expressão Regular
GR	Gramática Regular
IA	Inteligência Artificial
JEDI	Java Extraction and Dissemination of Information
LR	Linguagem Regular
NLP	Natural Language Processing
OSM	Object-Oriented System Model
OSM-L	OSM Language (Linguagem textual OSM)
PERL	Practical Extraction and Report Language
TSIMMIS	The Stanford – IBM Manager of Multiple Information Sources
W4F	World Wide Web Wrapper Factory

Lista de Figuras

Lista de Tabelas

Resumo

Extração de dados é o processo utilizado para obter e estruturar informações disponibilizadas em documentos semi-estruturados (ex: páginas da *Web*). A importância da extração de dados vem do fato que, uma vez extraídos, os dados podem ser armazenados e manipulados em uma forma mais estruturada. Dentre as abordagens existentes para extração de dados, existe a abordagem de extração baseada em ontologias. Nesta abordagem, ontologias são previamente criadas para descrever um domínio de interesse, gerando um modelo conceitual enriquecido com informações necessárias para extração de dados das fontes semi-estruturadas. A ontologia é utilizada como guia para um programa (*"parser"*) que executa a extração de dados dos documentos ou páginas fornecidos como entrada. O processo de criação da ontologia não é uma tarefa trivial e requer um cuidadoso trabalho de análise dos documentos ou páginas fontes dos dados. Este trabalho é feito manualmente por usuários especialistas no domínio de interesse da ontologia. Entretanto, em algumas situações, os dados que se desejam extrair estão modelados em bancos de dados relacionais. Neste caso, o modelo relacional do banco de dados pode ser utilizado para construção do modelo conceitual na ontologia. As instâncias dos dados armazenadas neste mesmo banco podem ajudar a gerar as informações sobre conteúdo e formato dos dados a serem extraídos. Estas informações sobre conteúdo e formato dos dados, na ontologia, são representadas por expressões regulares e estão inseridas nos chamados *"data frames"*. O objetivo deste trabalho é apresentar um método para criação semi-automática de ontologias de extração a partir das informações em um banco de dados já existente. O processo é baseado na engenharia reversa do modelo relacional para o modelo conceitual da ontologia combinada com a análise das instâncias dos dados para geração das expressões regulares nos *"data frames"*.

Palavras-chave: Extração de Dados, Construção de ontologias, Extração semântica

TITLE: "USING RELATIONAL DATABASES TO SEMIAUTOMATIC GENERATION OF DATA EXTRACTION ONTOLOGIES"

Abstract

Data extraction is the process used to gather and structure information in semi-structured documents (e.g. Web pages). The importance of data extraction is that, once extracted, data can be stored and manipulated in a more structured way. Among the existing approaches, there is the so-called ontology based data extraction. In this approach, ontologies are previously created to describe a domain of interest, generating a conceptual model enriched with information needed to identify and extract data items in the sources. An ontology is used as a guide to the parser that extracts data from the source documents. The process of creation of an ontology is not a trivial task and requires a careful analysis of documents or pages where data is on. An expert in the domain of interest of the ontology does this work manually. However, in some situations, the expected data to be extracted are already modeled in a relational database. In this case, the relational database schema can be used to the construction of the ontology conceptual model. The data instances stored in this database may help to generate information about format and content of the data to be extracted. This information about format and content, in the ontology, are represented by regular expressions in the data frames. This work presents a method for the semiautomatic creation of data extraction ontologies from an existing relational database. The process is based on reverse engineering of the relational database combined with the analysis of data instances to generate regular expressions in the data frames.

Keywords: Data extraction, Ontology construction, Semantic Extraction.

1 Introdução

A extração de dados a partir de fontes semi-estruturadas tem motivado muitas pesquisas para tratar e manipular dados que não estão armazenados em fontes estruturadas [LAE2002, SNO2001, FLO98]. A *Web* é um exemplo de fontes de dados semi-estruturados e contém uma grande quantidade de informações armazenadas sob diversas maneiras. Algumas abordagens foram criadas para extração de dados a partir da *Web* [HAM97, HUC98, SAH99, CRE2001, LAE2002a]. Dentre as várias abordagens, a mais comum é a extração sintática baseada em *"Wrappers"*. Outra abordagem é a extração semântica baseada em ontologias [EMB98, EMB99].

Ontologia é um termo utilizado na filosofia. Este termo lida com a natureza e organização da realidade. Na Ciência da Computação, o termo ontologia está sendo utilizado para descrever os conceitos do mundo real. Várias pesquisas foram feitas relacionadas com o conceito de ontologia [GRU93, FIK96, FAR96, GUA97], definindo-a como um conjunto de regras informais que têm por objetivo dar significado semântico à informação. Ontologias podem ser aplicadas na área de recuperação da informação, comércio eletrônico, Web Semântica e inteligência artificial [MAE2002]. Podem ser também utilizadas para extração de dados a partir de documentos não estruturados na *Web* [EMB98, EMB99].

Na abordagem de extração de dados baseada em ontologias, apresentada por David Embley e pelo Grupo de Extração de Dados (DEG) da *Brigham Young University*, uma ontologia é um modelo conceitual enriquecido com dados, relacionamentos e restrições, e pode ser relacionada com os dados de documentos semi-estruturados em um domínio de interesse [EMB98]. Nesta ontologia, o formato e conteúdo dos dados estão descritos nos chamados *"data frames"*. Um *"data frame"* contém expressões regulares e valores que podem ser utilizados para representar o conteúdo e formato dos dados. A ontologia é utilizada para guiar o processo de extração de dados a partir de documentos ricos em dados que possam ser descritos por ontologias relativamente pequenas [EMB99].

O extrator de dados do grupo DEG recebe como entrada a ontologia definida e o documento semi-estruturado (ex.: páginas *Web,* documentos) que contém informações sobre diversas instâncias dos conceitos de objetos não léxicos que aparecem na ontologia. A saída é um conjunto de registros com dados correspondentes aos conceitos de objetos léxicos definidos nos *"data frames"* da ontologia e que foram encontrados no documento semi-estruturado. Esta abordagem foi testada com sucesso em dois estudos de casos, extração de dados de classificados de carros e obituários, disponíveis na Web e fornecidos pelo "Salt Lake Tribune" (www.sltrib.com) e "Arizona Daily Star" (www.azstarnet.com) [EMB98, EMB99]. Além destes estudos, outros casos estão disponíveis na página do grupo (http://www.deg.byu.edu). Nos casos apresentados, as métricas utilizadas para análise dos resultados foram as taxa de recuperação e precisão que mostram, respectivamente, o percentual de dados relevantes recuperados e o percentual de dados recuperados de forma correta [SAL83]. Nos dois casos citados anteriormente, as taxas de recuperação e precisão foram aproximadamente de 90% e 98%.

A construção da ontologia para extração de dados não é uma tarefa simples e requer do usuário a análise de um grande número de documentos semi-estruturados, a

partir dos quais pretende-se extrair os dados. No caso da extração a partir de obituários, 128 exemplos foram analisados para que a ontologia de extração pudesse ser criada [EMB99] e no caso dos classificados de carros, quase 600 classificados foram analisados [EMB98]. A vantagem da extração semântica em relação à extração sintática é que, uma vez criada a ontologia, a mesma pode ser utilizada para extração de dados de um número maior de documentos, desde que pertencentes ao mesmo domínio de problema. A desvantagem é que a construção da ontologia é uma tarefa manual, lenta e sujeita a erros.

Em algumas situações, organizações que precisam extrair dados a partir de documentos semi-estruturados, possuem bancos de dados relacionais que modelam os dados a serem extraídos e podem até mesmo conter instâncias destes dados. Como exemplo, uma empresa do segmento de comércio que tenha um banco de dados sobre produtos que vende, pode estar interessada em extrair dados de páginas *Web* de seus concorrentes. Outro exemplo seria uma organização que necessita extrair dados legados a partir de fontes de dados semi-estruturadas com o objetivo de incluí-los no banco de dados. Em domínios de aplicação como estes, o processo semi-automático de construção da ontologia poderia ser muito útil.

O objetivo deste trabalho é apresentar um método para, semiautomaticamente, construir a ontologia de extração de dados utilizada pelo extrator do grupo DEG, a partir de um banco de dados relacional já existente. Este método utiliza as informações armazenadas no esquema físico do banco de dados, bem como as informações contidas nos dados armazenados neste mesmo banco.

Na abordagem do grupo DEG, uma ontologia é especificada utilizando a abordagem OSM – *"Object-Oriented System Model"*. OSM é uma abordagem orientada a objetos representada pela notação gráfica ou textual do modelo conceitual [EMB98a].

Para construir a ontologia no modelo textual OSM a partir de um banco de dados relacional, o método propõe duas tarefas: a engenharia reversa do modelo relacional para o modelo OSM, e a geração das expressões regulares a serem inseridas nos *"data frames"* a partir das instâncias dos dados no banco de dados. O processo de construção é semi-automático porque o usuário toma decisões sobre alternativas de geração da ontologia. As regras adotadas para engenharia reversa do modelo relacional são as já definidas na literatura [BAT92]. A geração das expressões regulares a partir das instâncias dos dados não é uma tarefa trivial, e envolve o estudo de algoritmos capazes de construir uma expressão regular a partir de um conjunto de dados.

O texto deste trabalho está organizado como segue: O capítulo 2 explica a extração semântica de dados, abordagens de extração existentes, com ênfase no trabalho do grupo DEG, exemplificando com um estudo de caso. O capítulo 3 mostra a solução adotada para geração da expressão regular e comenta os estudos realizados sobre algoritmos existentes na literatura para o reconhecimento de linguagens. O capítulo 4 descreve o processo semi-automático de construção da ontologia, composto pelas etapas de engenharia reversa a partir do modelo relacional e os algoritmos para geração da expressão regular. O capítulo 5 apresenta dois estudos de caso e os resultados obtidos a partir de ontologias geradas semiautomaticamente para uso no extrator do grupo DEG. O capítulo 6 apresenta as conclusões e trabalhos futuros. O protótipo da ferramenta para construção da ontologia é apresentado em Anexo.

2 Extração Semântica

Este capítulo apresenta algumas das abordagens de extração de dados existentes, com ênfase no trabalho que o grupo de extração de dados da *Brigham Young University* (DEG) vem desenvolvendo nos últimos 4 anos. O extrator do grupo DEG é apresentado com um exemplo de aplicação de extração de dados a partir de documentos semi-estruturados.

2.1 Extração de Dados

O interesse pela extração de dados a partir de documentos semi-estruturados ou em linguagem natural vem desde os anos 50, em trabalhos realizados por Zellig Harris, e mais recentemente sua idéia foi implementada para textos médicos na "New York University" [GRI97]. Exemplos no Brasil mostram que em hospitais as anamneses (textos onde consta o histórico clínico de pacientes) são armazenadas em colunas do tipo texto nos bancos de dados [NAR2001]. Estes textos são ricos em dados e vários esforços são feitos no sentido de transformá-los em uma forma mais estruturada.

Com o crescimento da *Web* nos últimos anos, uma grande quantidade e variedade de informações tornaram-se disponíveis on-line. As páginas HTML publicadas na *Web* são um exemplo de documentos semi-estruturados. Acompanhado deste crescimento, vieram alguns problemas como a localização e utilização das informações publicadas. Para solucionar estes problemas, normalmente são utilizadas buscas através de palavras-chave, navegação através de *"links"*, indexação de páginas, os quais apresentam limitações e são um tanto quanto intuitivos no uso [MOU2002]. Dentre algumas estratégias importantes para pesquisa e recuperação destas informações, como bibliotecas digitais e linguagens de consulta, existem alguns mecanismos para extração de dados a partir da *Web* [HAM97, HUC98, EMB98, EMB99, SAH99]. Estas pesquisas motivaram outros trabalhos relacionados à extração, destacando o trabalho de extração a partir de fontes semi-estruturadas para armazenamento de forma estruturada [DOR2000, MEL2000, TEI2000, SNO2001, CRE2001, LAE2002a].

A abordagem mais comum para extração de dados na *Web* é através de programas extratores denominados *"Wrappers"*, os quais mapeiam as páginas *Web* com o objetivo de extrair informações relevantes, disponibilizando-as em um formato estruturado. A utilização de *"Wrappers"* é muito dependente da estrutura do documento fonte, tornando difícil sua utilização para documentos de formatos diferentes. Outra desvantagem ocorre quando um documento é alterado, gerando novo trabalho na atualização do programa extrator [EMB98]. Ana Maria Moura [MOU2002] classifica os *"Wrappers"* por contexto ou por conteúdo. Ferramentas como W4F (*"World Wide Web Wrapper Factory"*) [SAH99] e DEByE (*"Data Extraction by Example"*) [LAE2002a] são exemplos de extratores por contexto. A ferramenta do grupo DEG [EMB98, EMB99] é citada como exemplo de extratores por conteúdo utilizando ontologias específicas de domínio.

Em recente estudo, Alberto Laender, Altigran da Silva, Berthier Ribeiro-Neto e Juliana Teixeira [LAE2002] apresentam uma taxonomia para agrupar as abordagens de extração de dados a partir da *Web*. Esta taxonomia foi feita de acordo com a principal técnica de cada ferramenta para extração dos dados, gerando os seguintes grupos:

"Languages for Wrapper Development", *"HTML-aware Tools"*, *"NLP-based Tools"*, *"Wrapper Induction Tools"*, *"Modeling-based Tools"* e *"Ontology-based Tools"*. Dentre estas, as ferramentas que motivaram este trabalho estão nos grupos de *"Languages for Wrapper Development"* com a ferramenta TSIMMIS [HAM97], *"HTML-aware Tools"* com as ferramentas W4F [SAH99] e RoadRunner [CRE2001], *"Modeling-based Tools"* com a ferramenta DEByE [LAE2002a], e *"Ontology-based Tools"* com o extrator do grupo DEG [EMB98, EMB99].

Além da classificação em grupos, Laender [LAE2002] propôs uma análise qualitativa das ferramentas de extração segundo seu grau de automação e grau de flexibilidade. O grau de automação indica o quanto o usuário participa durante o processo de geração do *"Wrapper"* para extração dos dados. O grau de flexibilidade indica se o extrator continua funcionando se ocorrer mudanças nos documentos ou páginas fontes para extração.O gráfico da Figura 2.1 mostra a análise qualitativa dos grupos de ferramentas citados.

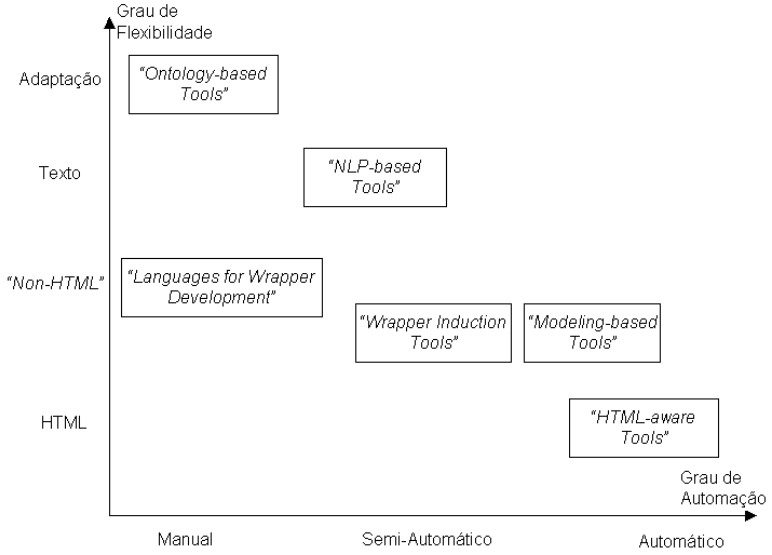

FIGURA 2.1 – Análise qualitativa das ferramentas de extração

No Grupo das *"Ontology-based Tools"*, o extrator do grupo DEG é considerado como a ferramenta mais representativa, apesar da existência de outras ferramentas como, por exemplo, a X-tract [ABA99], uma ferramenta para extração de dados a partir de descrições textuais botânicas. Nas abordagens de extração semântica, uma ontologia é previamente construída para descrever os dados de interesse, incluindo formato dos léxicos, relacionamentos e palavras-chave. A construção da ontologia requer um trabalho manual e cuidadoso por parte do usuário. Um analisador sintático (*"parser"*) analisa a ontologia. A ferramenta de extração recupera e extrai dados presentes em documentos ou páginas *Web* fornecidos como entrada no processo [EMB98, EMB99].

Analisando apenas a ferramenta do grupo DEG, seu grau de automação é baixo, pois apenas o extrator de registros está automatizado, enquanto a construção da ontologia é feita manualmente pelo usuário. Já o grau de flexibilidade é alto porque a ontologia se adapta às mudanças na página fonte. Permite, inclusive, extração a partir de outras páginas desde que pertencentes ao mesmo domínio de problema.

O foco deste trabalho concentra-se na abordagem de extração desenvolvida pelo grupo DEG. A ontologia criada de forma semi-automática pela abordagem proposta serve como entrada para o extrator do grupo DEG.

2.2 Extração de Dados baseada em Ontologia

Esta seção apresenta o extrator de dados do grupo DEG [EMB98, EMB99]. Esta abordagem de extração é aplicada a documentos semi-estruturados, ricos em dados, e que possuam uma estrutura de registros, isto é, blocos de dados que contenham os mesmos tipos de informação [EMB98]. Um conjunto de dados é semi-estruturado se sua estrutura é autodescritiva ou sem esquema, indicando que não existe descrição separada do tipo ou estrutura dos dados [ABI2000]. As páginas *Web* utilizadas nos trabalhos do grupo DEG [EMB99], como exemplo de dados semi-estruturados, foram obituários, classificados de carros e empregos, pesquisa sobre filmes, informações meteorológicas, entre outras que podem ser encontradas na página do grupo, http://www.deg.byu.edu.

O extrator será descrito através de um exemplo de aplicação utilizado em um dos estudos de caso deste trabalho. Os dados são extraídos a partir de uma página de livraria virtual – Livraria Cultura. Este tipo de aplicação para extração de dados preenche os requisitos do extrator do grupo DEG, que são: a) poder ser descrito por uma ontologia relativamente pequena e b) conter múltiplos registros sobre uma entidade principal (neste caso, livro) na ontologia.

A página utilizada para extração dos dados, na Figura 2.2, contém os seguintes dados: Título do livro, Editora, Ano de Publicação, autores do livro, entre outras informações não relevantes. Para obter esta página, foi necessário, antes, estabelecer o critério de pesquisa na livraria virtual. Neste caso, o critério de busca foi os livros do autor Jorge Amado.

FIGURA 2.2 – Página de Entrada ao Extrator do Grupo DEG

Uma vez definido o domínio de problema e os documentos semi-estruturados (páginas *Web*), a partir do qual são extraídos os dados, o próximo passo é a definição da ontologia de extração. Em seguida à definição da ontologia, o extrator do grupo DEG executa as seguintes etapas:

- Execução do analisador sintático (*"parser"*) da ontologia para geração de dois arquivos: regras de extração e uma lista de Objetos e relacionamentos;
- Execução de um processo para extração de registros a partir do documento semi-estruturado de entrada.

As próximas subseções descrevem o processo manual de definição da ontologia e as etapas executadas pelo extrator para obter os dados das páginas de interesse, que servem como entrada ao processo de extração.

2.2.1 Definição da Ontologia de Extração

Na literatura, existe muita controvérsia sobre o termo ontologia. Este conceito existe há muito tempo na área de filosofia (Aristóteles 384-322 AC), o qual está relacionado com o ramo da existência dos seres. Na ciência da computação, o termo começou a ser utilizado na área de Inteligência Artificial (IA). Atualmente, este termo é utilizado também em áreas como comércio eletrônico, processamento de linguagem Natural (NLP) [COW96], integração de informações, bibliotecas digitais, engenharia do conhecimento [MAE2002], recuperação da informação (mecanismos de busca), gerência do conhecimento [BEN98, LIA99], e extração de dados [EMB98, EMB99], foco principal deste trabalho.

Para o estudo da extração de dados baseada em ontologias, o conceito de ontologia é definido como um modelo conceitual enriquecido que auxilia na extração de dados. É enriquecido porque além de propriedades como restrições e relacionamentos, é incrementado com representação do formato e valores léxicos dos dados, permitindo que o processo de extração possa ser feito o mais automático possível [EMB98, EMB99].

A ontologia é definida manualmente pelo usuário, analisando-se as páginas das quais se deseja extrair os dados. Esta análise pode envolver um grande número de páginas. Na extração de dados a partir a partir de obituários, foram analisados 128 exemplos de obituários, e na extração a partir de classificados de carros, foram analisados quase 600 anúncios. Na construção da ontologia de livros para demonstrar o funcionamento do extrator, foram analisados apenas os exemplos de livros do autor Jorge Amado disponíveis na Livraria Cultura, num total de 142 títulos.

Uma ontologia, na abordagem do grupo DEG, é especificada utilizando a abordagem OSM – *"Object-Oriented System Model"*. O modelo é orientado a objetos, classificados em conjuntos de objetos não léxicos ou em conjuntos de objetos léxicos, com relacionamentos entre si [EMB98a]. Comparando com a abordagem ER, um conjunto de objetos não-léxicos corresponde às entidades e os conjuntos de objetos léxicos correspondem aos atributos das entidades. No modelo OSM, uma ontologia pode ser representada de uma forma mais abstrata, através de seu modelo gráfico, ou de forma textual, na qual é utilizada a linguagem OSM-L [LID95, EMB98a]. É no formato textual que a ontologia é construída para ser utilizada no processo de extração de dados. O modelo textual é enriquecido por dois conceitos: *"data frames"* e valores léxicos. Os *"data frames"* descrevem como os dados podem estar representados no documento. Os valores léxicos são uma lista dos possíveis valores para um conjunto de objetos léxicos, ou seja, um dicionário.

Na Figura 2.4, um *"data frame"* contendo um dicionário dos possíveis valores para o conjunto de objetos `TITULO` pode ser verificado nas linhas 4 a 13. Outro exemplo de *"data frame"*, nas linhas 35 a 42 da mesma figura, contém a descrição de como o

dado se encontra no documento. Alguns Aspectos do modelo OSM estão descritos a seguir.

- *Objeto* – um objeto é uma pessoa, lugar ou coisa, podendo ser algo físico ou conceitual. No modelo gráfico é representado por um ponto cheio. Pode ser léxico ou não léxico. Um objeto cuja representação é única é considerado léxico. Exemplo: Nomes de pessoas, paises, horas, título do livro (algo que os represente). Um objeto é considerado não léxico quando sua representação difere do objeto (são representados por identificadores). Exemplo: Pessoa, Cliente, Livro;
- *Relacionamentos*: relacionam dois ou mais objetos e tem um nome que descreve o relacionamento. Na Figura 2.3, a seta indica a direção de leitura no relacionamento. O relacionamento binário é descrito por uma frase que inclui o nome dos dois objetos conectados pelo relacionamento. Exemplo: "*Livro has titulo*" (livro tem título). Os relacionamentos podem ter setas nos dois sentidos, gerando múltiplos nomes para um relacionamento. Isto é devido ao fato do relacionamento poder ser lido nos dois sentidos;
- *Conjunto de objetos*: é um agrupamento de objetos. Os retângulos com o nome do objeto em seu interior representam os conjunto de objetos. Os pontilhados representam os conjunto de objetos léxicos (título, ano publicação, editora, preço) e os com linhas contínuas representam conjuntos de objetos não léxicos (livro);
- *Conjunto de relacionamentos*: um conjunto de relacionamentos é representado por um losango. O nome do conjunto de relacionamentos inclui o nome dos conjuntos de objetos conectados. Para relacionamentos binários o losango é omitido, permanecendo apenas o nome e a seta de direção;
- *Restrições de participação*: Designam o número mínimo e máximo de vezes que um objeto do conjunto participa de um relacionamento. Por exemplo, na Figura 2.3, a simbologia "*1:**" perto do conjunto de objetos *editora* indica que uma editora está associada com pelo menos um *livro* e talvez muitos. Já a simbologia *"0:1"* perto do conjunto de objetos *livro* indica que um livro não necessariamente participa no relacionamento, mas o máximo deve ser 1.

Na abordagem OSM existem mais representações e definições para restrições, como restrições de integridade referencial e restrições de cardinalidade, que não estão representadas neste estudo de caso para não torná-lo extenso. O objetivo aqui é mostrar as definições da abordagem OSM necessárias para a ontologia de extração de dados.

Para a aplicação da extração de dados a partir do domínio de problema escolhido, livraria virtual, a ontologia definida segundo o modelo OSM está representada graficamente pela Figura 2.3.

A ontologia textual está representada pela Figura 2.4. Dos aspectos apresentados anteriormente, a linha 1 representa o conjunto de objetos não léxicos. As linhas 3, 15, 27 e 34 representam os conjuntos de objetos léxicos e o relacionamento com o conjunto de objetos não léxico `LIVRO`. Junto com o relacionamento, a restrição de participação para os objeto no relacionamento está representada entre colchetes.

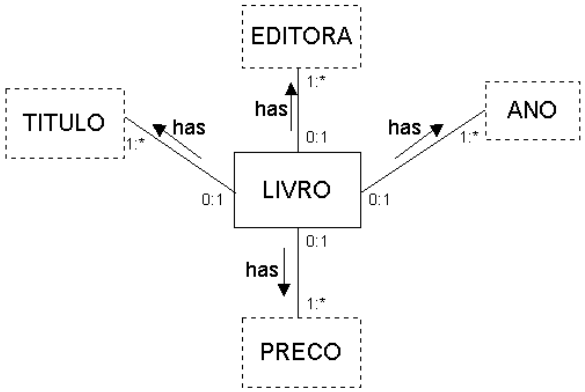

FIGURA 2.3 – Representação gráfica da Ontologia

```
 1: LIVRO [-> object];
 2:
 3: LIVRO [0:1] has TITULO [1:*];
 4: TITULO matches [100] case insensitive
 5: constant
 6:   { extract "\bO?\s*?Amor\s*(do)?\s*?soldado\b"; },
 7:   { extract "\b(Os)?\s*?Velhos\s*marinheiros\b"; },
 8:   { extract "\bO?\s*?Amor\s*(do)?\s*?soldado\b"; },
 9:   { extract
10:     "\b(Os)?\s*?Subterraneos\s*(da)?\s*?liberdade\b"; },
11:   ...
12:   ;
13: end;
14:
15: LIVRO [0:1] has ANO [1:*];
16: ANO matches [4]
17:   constant
18:   {
19:    extract "19[5-9][0-9]";
20:   };
21:   constant
22:   {
23:    extract "200[0-2]";
24:   };
25: end;
26:
27: LIVRO [0:1] has EDITORA [1:*];
28: EDITORA matches [50]
29: constant
30:   {extract “Martins”; },
31:   {extract “Record”;  };
32: end;
33:
34: LIVRO [0:1] has PRECO [1:*];
35: PRECO matches [15] case insensitive
36:    constant
37:    {
38:           context "R\$\s*?([0-9]{1,3},?)+";
39:           extract "([0-9]{1,3},?)+";
40:
41:    };
42: end;
```

FIGURA 2.4 – Representação Textual da Ontologia

2.2.2 Execução do Analisador Sintático (parsing) na Ontologia

O analisador sintático (*parser*) da ontologia é um programa que, a partir da ontologia definida, gera dois arquivos como saída: regras para constantes e palavras-chave e uma lista de objetos com seus relacionamentos e restrições. O arquivo de regras para constantes e palavras-chave é um arquivo de *"tags"* de expressões regulares em sintaxe Perl. Estas *"tags"* são um conjunto de nomes de objetos da ontologia OSM para expressões regulares que descrevem constantes. Este arquivo servirá como entrada para o programa reconhecedor de constantes e palavras-chave. A Figura 2.5 mostra um exemplo das regras e constantes.

```
TITULO : \bO\s*Amor\s*do\s*soldado\b : 0 :  : 0 : -1
TITULO : \bOs\s*Velhos\s*marinheiros\b : 0 :  : 0 : -1
TITULO : \bO\s*Amor\s*do\s*soldado\b : 0 :  : 0 : -1
TITULO : \bOs\s*Subterraneos\s*da\s*liberdade\b : 0 :  : 0 : -1
TITULO : \bSeara\s*vermelha\b : 0 :  : 0 : -1
TITULO : \bSao\s*Jorge\s*dos\s*ilheus\b : 0 :  : 0 : -1
TITULO : \bBahia\s*de\s*todos\s*os\s*Santos\b : 0 :  : 0 : -1
:
:
:
ANO : 19[5-9][0-9] : 0 : 19[5-9][0-9] : 0 :  : 0 : -1
ANO : 200[0-2] : 0 : 200[0-2] : 0 :  : 0 : -1
EDITORA : (Editora\s+Record|Martins|Record) : 0 :  : 0 : -1
PRECO : ([0-9]{1,3},?)+ : 0 : R\$\s*?([0-9]{1,3},?)+ : 0 :  : 0 : -1
```

FIGURA 2.5 – Arquivo de Regras e Constantes de Extração

O segundo arquivo é uma lista de objetos, relacionamentos e restrições. Esta lista provê um mapa dos relacionamentos na ontologia, mostrando qual é a cardinalidade que designa quais relacionamentos são um-para-um, um-para-muitos e muitos-para-muitos (neste exemplo, os relacionamentos são binários). A Figura 2.6 mostra a lista de objetos.

```
Object: LIVRO;
Nonlexical: ;
Lexical: TITULO, ANO, EDITORA, PRECO;
LIVRO: TITULO [0:1];
LIVRO: ANO [0:1];
LIVRO: EDITORA [0:1];
LIVRO: PRECO [0:1];
```

FIGURA 2.6 – Lista de Objetos, Relacionamentos e Restrições

2.2.3 Extração de Registros da Página Web

Este processo é feito por um programa Perl [CHR2001] que faz a extração dos dados do documento, gerando um arquivo que irá conter as *"tags"* referentes aos objetos da ontologia. Quando o programa Perl encontra uma cadeia de caracteres **C** de acordo com uma expressão regular **E** com a *"tag"* **T**, gera uma saída com a *"tag"* **T** como sendo o nome, a cadeia de caracteres **C** e a posição inicial e final da cadeia de caracteres na página *Web*. Utilizando uma página semelhante à da Figura 2.2 como entrada e aplicando-se as regras de constantes e palavras-chave, foram reconhecidas as cadeias de caracteres para `TITULO`, `ANO`, `PRECO` e `EDITORA`, gerando os registros demonstrados na Figura 2.7. O primeiro campo é o nome do conjunto de objetos, seguido pela constante extraída da página. Os dois últimos campos representam a posição inicial e final da cadeia de caracteres no texto.

```
TITULO|MORTE E A MORTE DE QUINCAS BERRO DAGUA|1|38
ANO|1996|65|68
PRECO|17,00|89|93
EDITORA|RECORD|210|215
#####

TITULO|CAPITAES DA AREIA|1|17
ANO|1996|41|44
PRECO|25,00|65|69
EDITORA|RECORD|186|191
#####

TITULO|GABRIELA, CRAVO E CANELA|1|24
ANO|1995|48|51
PRECO|31,00|72|76
EDITORA|RECORD|193|198
#####
```

FIGURA 2.7 – Registros extraídos a partir da página Web de entrada

O algoritmo de extração de registros é uma funcionalidade muito importante implementada no extrator do grupo DEG. De uma forma geral, o algoritmo monta uma árvore com a estrutura da página, heuristicamente faz uma busca na árvore para pesquisar sub-árvores com maior probabilidade de conter registros e, também heuristicamente, encontra separadores mais prováveis entre os registros irmãos da sub-árvore. Uma característica de páginas *Web* com múltiplos registros é que a pessoa que a criou provavelmente seguiu algum padrão de estrutura, não necessariamente rígido. Isto facilita bastante o processo de extração dos registros [EMB99].

Descrevendo com mais detalhes esta ferramenta que executa a extração dos registros, o primeiro passo é a montagem da árvore com a estrutura da página. Um documento HTML na sua maioria apresenta *"tags"* de início com uma outra *"tag"* de

fechamento. Dentro destas, outras *"tags"* podem estar aninhadas, formando uma estrutura hierárquica denominada *"árvore de tags"*. A Figura 2.8 mostra um exemplo de uma árvore de *"tags"*. HTML é a raiz da árvore, com duas *"tags"* aninhadas: *"head"*, que contém *"title"*, e *"body"* que contém *"table"*. Da mesma forma, outras *"tags"* vão se aninhando umas dentro das outras. Dentro da "*tag"* <td> existem várias *"tags"* aninhadas, mas cada uma delas sendo irmãs entre si, formando as folhas das árvores [EMB99]. Este exemplo de árvore pode ser considerado para outros tipos de páginas que, em geral, tem o mesmo tipo de estrutura. O processo de extração pode ser definido nos seguintes passos:

- Inicialmente, é identificada a região do documento que contenha os registros de interesse. Para isso, deve-se fazer uma pesquisa na árvore para encontrar a sub-árvore com o maior número de filhos. Pelos experimentos já realizados, pode-se concluir que a raiz com o maior número de filhos contém os registros de interesse;
- Dentro das *"tags"* com maior número de filhos, o próximo procedimento será identificar as *"tags"* que separam os registros. Neste exemplo, as *"tags"* candidatas a separadores são **h1**, **h4** e **hr**. São descartadas as *"tags"* que possuem poucas ocorrências, como a **h1**;
- Uma vez identificado a *"tag"* separadora dos registros, o procedimento inclui "#####" entre linhas em branco imediatamente antes de cada ocorrência do separador, conforme a Figura 2.7 acima. As *"tags"* são removidas, fazendo com que os registros de interesse sejam obtidos.

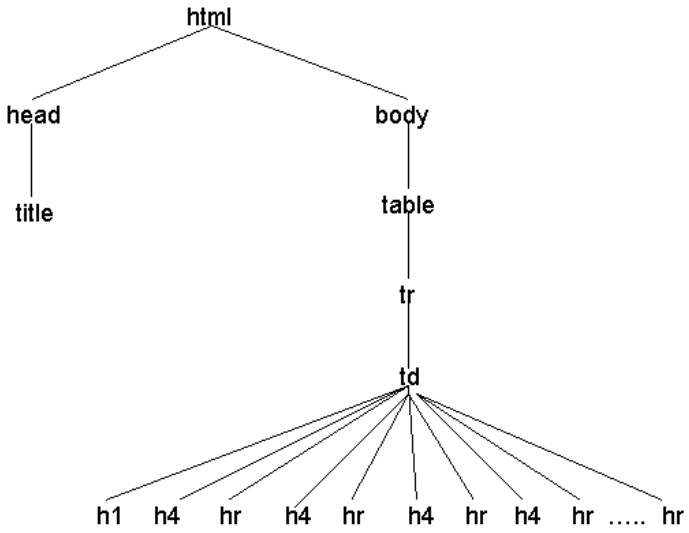

FIGURA 2.8 – Árvore de "*tags"* a partir de um documento HTML

A determinação da *"tag"* separadora dos registros é feita utilizando-se de cinco heurísticas:

- *Maior ocorrência*: relaciona as "*tags"* candidatas através do número de ocorrências. Determina que o separador aparece freqüentemente quando existem muitos registros;

- *Separador identificável*: utiliza uma lista pré-determinada de "*tags"* separadoras. A lista utilizada pelo procedimento é **hr**, **td**, **tr**, **a**, **table**, **p**, **Br**, **h4**, **h1**, **b**, **i**.;
- *Desvio padrão*: identifica os registros pelo comprimento. Normalmente os registros repetidos possuem o mesmo tamanho. É calculado o desvio padrão do comprimento e, o que tiver o menor comprimento, é considerado;
- *Padrão repetido*: considera que a divisão entre os registros é formada por *"tags"* que se repetem. Se uma ou mais *"tags"* aparecem como limites de um registro, então o registro pode ser identificado;
- *Combinação com a ontologia*: baseado no conteúdo ontológico do registro. A ontologia diz quais campos deveriam estar contidos nos registros identificados no documento. O número de registros identificados é contado, calculando-se a média de campos identificados por registro. São então relacionadas as *"tags"* candidatas pelo número de ocorrências que correspondam à média calculada.

2.3 Uso de *"Data Frames"*

Os *"Data Frames"* contém os padrões das cadeias de caracteres para constantes dos objetos léxicos, palavras-chave para descrever o contexto de palavras sinônimas e armazenam os comandos de extração de dados para os documentos do domínio de interesse. Por estas características, os *"data frames"* são uma forma de encapsular os conceitos de um conjunto de objetos com as suas propriedades. A Figura 2.9 mostra um exemplo.

```
LIVRO [0:1] has ANO [1:*];
ANO matches [4]
   constant
   {
    extract "19[5-9][0-9]";
   };
   constant
   {
    extract "200[0-2]";
   };
end;
```

FIGURA 2.9 – Exemplo de *"Data Frame"*

Ainda com base na Figura 2.9, o número entre colchetes após a palavra reservada *"matches"* indica o tamanho do objeto léxico ANO. No caso deste *"data frame"*, os padrões das constantes estão declarados através de duas expressões regulares que permitem a extração de constantes que variam desde 1950 até 2002. Este é um exemplo de expressão regular padrão, que foi elaborado manualmente.

2.4 Arquitetura do Extrator do Grupo DEG

Esta é uma abordagem baseada na criação de um modelo-conceitual que guia e estrutura a extração de dados a partir de documentos semi-estruturados na *Web*. Este modelo conceitual é a ontologia, a qual é construída manualmente através da análise do

domínio de interesse para se definir os objetos e seus relacionamentos. Além desta análise, é preciso definir o formato e conteúdo dos dados a serem extraídos, constituindo os *"data frames"* na ontologia. Ontologia e página Web servem como entrada ao programa extrator que automaticamente extrai os registros de acordo com a ontologia definida. A Figura 2.10 mostra uma visão geral do processo de extração proposto pelo grupo DEG.

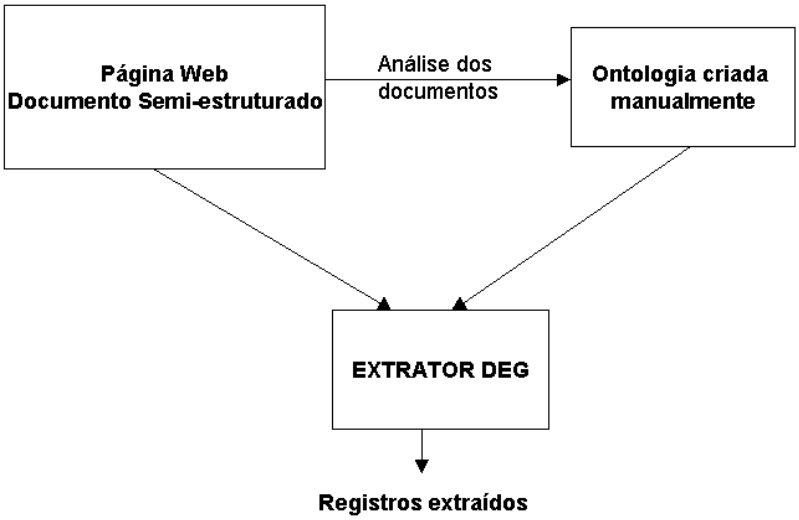

FIGURA 2.10 – Arquitetura do extrator do grupo DEG

2.5 Eficiência do Método de Extração

Os experimentos realizados pelo grupo DEG mostraram que é possível obter dados da *Web* com altos percentuais de recuperação dos dados (*"recall"*) e precisão (*"precision"*). Além dos experimentos realizados pelo grupo, dois estudos de caso são apresentados no capítulo 5 deste trabalho, mostrando os percentuais de recuperação e precisão alcançados. A principal diferença é que neste trabalho a ontologia não é criada manualmente, mas sim através do método semi-automático de construção.

As medidas de recuperação e precisão estão atreladas a uma coleção de documentos ou a um conjunto de pesquisas (*"queries"*). Neste ambiente, é possível variar a política de indexação ou metodologia de pesquisa, e verificar como elas alteram a performance em termos de recuperação e precisão. Em resumo, estas medidas refletem o sucesso de sistemas de recuperação de informação em encontrar as necessidades dos usuários [SAL83].

2.6 Resumo do Capítulo

Este capítulo mostrou as abordagens de extração de dados existentes, com maior detalhe para o extrator do grupo DEG, o qual utiliza ontologias para guiar o processo de extração de dados. A ferramenta de extração, disponível na Internet, foi descrita por um domínio de problema utilizado em um dos estudos de caso deste trabalho.

Os próximos capítulos mostram o processo proposto para criar a ontologia de forma semi-automática.

3 Algoritmos para Reconhecimento de Linguagens

Este capítulo tem como objetivo mostrar a solução adotada para geração das expressões regulares a partir de um conjunto de palavras. Para isso, um breve resumo do que existe na literatura sobre linguagens é também apresentado. As expressões regulares são utilizadas, na ontologia do grupo DEG, para reconhecimento e extração dos dados nos documentos semi-estruturados. Os estudos apresentados neste capítulo servem como base para a compreensão dos algoritmos de geração das expressões regulares, implementados neste trabalho, e apresentados na seção 4.3 do capítulo 4. Os algoritmos apresentados neste capítulo são conhecidos na literatura.

3.1 Classificação das linguagens

Uma expressão regular é um formalismo denotacional, ou gerador, pois pode representar ou inferir as palavras de uma linguagem. Uma expressão regular sobre um alfabeto é composta de expressões (cadeias de caracteres) utilizando operações de união, concatenação e iteração [HOP79]. Nesta dissertação, as cadeias de caracteres (conjuntos de palavras) são as instâncias dos dados no banco de dados a partir das quais deseja-se extrair dados de documentos semi-estruturados. A partir das instâncias dos dados no banco, serão geradas as expressões regulares que irão compor os *"data frames"* na ontologia.

O processo de geração de formalismos (expressões regulares, gramáticas, autômatos finitos) a partir de um conjunto de palavras de uma linguagem tem sido extensamente estudado na área de processamento de linguagem natural (NLP) e é conhecido como inferência gramatical [DEN2001, GON78, NAD93, SCH92]. Uma linguagem é um conjunto de palavras sobre um alfabeto, isto é, uma linguagem sobre um alfabeto é um subconjunto finito ou infinito contável de palavras sobre este alfabeto [GON78]. Entre 1956 e 1963, Noam Chomsky desenvolveu uma teoria sobre linguagens formais [HOP79, NAD93]. Chomsky definiu quatro classes de linguagens, apresentadas na Figura 3.1.

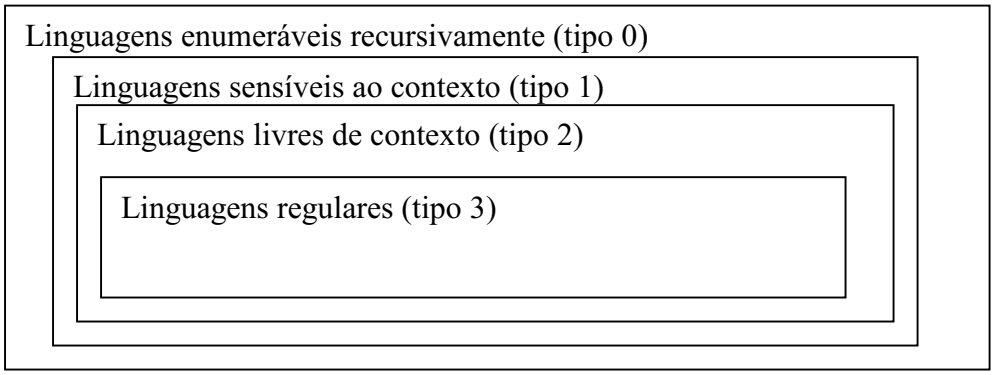

FIGURA 3.1 – Tipos de Linguagens segundo a hierarquia de Chomsky

A linguagem regular será o foco deste estudo. Uma linguagem regular (tipo 3 na hierarquia de Chomsky) é dita regular se e somente se é reconhecida por um autômato finito [GON78]. Outra definição de [HOP79, AHO74] declara que uma linguagem é regular se e somente se ela pode ser denotada por uma expressão regular.

3.1.1 Formalismos para representar ou reconhecer uma linguagem regular

As instâncias do dados no banco de dados satisfazem à definição de linguagens regulares porque formam um conjunto finito. Partindo-se do princípio que o conjunto de palavras pertence a uma linguagem regular, ela pode ser reconhecida ou representada pelos formalismos do tipo reconhecedor (Autômatos Finitos), Gerador (Gramática Regular) e Denotacional (Expressão Regular), definidos como:

- *Autômatos Finitos (AF):* Um autômato finito é um modelo matemático de máquinas de computação, para o qual, dado uma cadeia de caracteres de entrada, tem a capacidade de reconhecer se este padrão pertence ou não à linguagem especificada [GON78]. Um autômato finito pode ser determinístico, não determinístico ou com movimentos vazios e são equivalentes entre si. Ser equivalente significa dizer que dois autômatos finitos distintos reconhecem a mesma cadeia de caracteres [HOP79];
- *Gramática Regular:* Gramáticas provêm um meio de definir linguagens através de um conjunto finito de regras que descrevem como construir uma cadeia de caracteres válida. Uma gramática regular pode descrever uma linguagem regular [HOP79];
- *Expressão Regular:* Como já definido anteriormente, uma expressão regular pode descrever uma linguagem regular. É a descrição de uma linguagem regular utilizando operações de concatenação e união dos símbolos de seu alfabeto.

A Figura 3.2 mostra a tradução de uma linguagem regular para qualquer destes formalismos utilizando os teoremas já comprovados na literatura [HOP79]. Deve-se ressaltar que a construção de uma expressão regular a partir de um autômato finito não foi comprovada. Ainda na Figura 3.2, AFD significa Autômato Finito Determinístico, AFN significa Autômato Finito Não Determinístico, AFε significa Autômato Finito com Movimentos Vazios, GR significa Gramática Regular e ExR significa Expressão Regular. Os AFD, AFN e AFε são equivalentes.

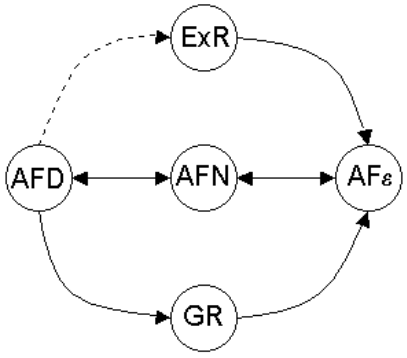

FIGURA 3.2 – Tradução dos formalismos das Linguagens Regulares

Existem algoritmos que geram uma gramática regular a partir de linguagens regulares [GON78, NAD93], e existem teoremas que provam que um autômato finito é capaz de reconhecer uma cadeia de caracteres de uma linguagem regular [HOP79]. A Figura 3.3 mostra as duas abordagens estudadas para, a partir de linguagens regulares (LR), gerar uma gramática regular (GR) ou um autômato finito (AFD) com o objetivo final de obter a expressão regular (ExR).

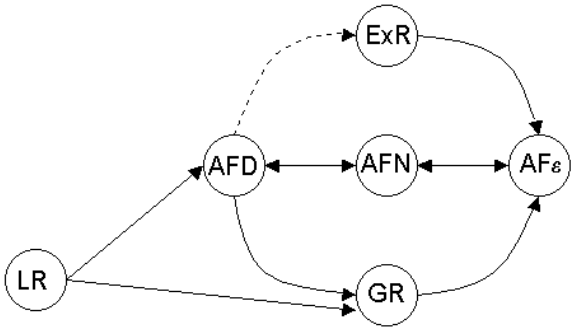

FIGURA 3.3 – Geração de gramáticas a partir de linguagens regulares

A três próximas seções mostram a abordagem adotada para gerar um autômato finito determinístico e, a partir deste, como é gerada a expressão regular.

3.2 Geração do Autômato Finito Determinístico

A abordagem utilizada neste trabalho para gerar uma expressão regular que represente exatamente um conjunto de exemplos, está baseada na representação do conjunto de instâncias através de um autômato finito determinístico não cíclico.

A idéia principal é gerar um autômato finito determinístico não cíclico com os exemplo do conjunto. Utilizando um conjunto de exemplos {JRTK, JRMP, NTXV, BCLQ, BCLX}, o autômato finito é gerado. Para cada exemplo, o estado inicial é o mesmo, e cada caractere do alfabeto leva a um novo estado. O último caractere de cada exemplo é o estado final. Todas as palavras do conjunto, submetidas a este autômato são aceitas. A Figura 3.4 mostra o grafo do autômato finito determinístico gerado.

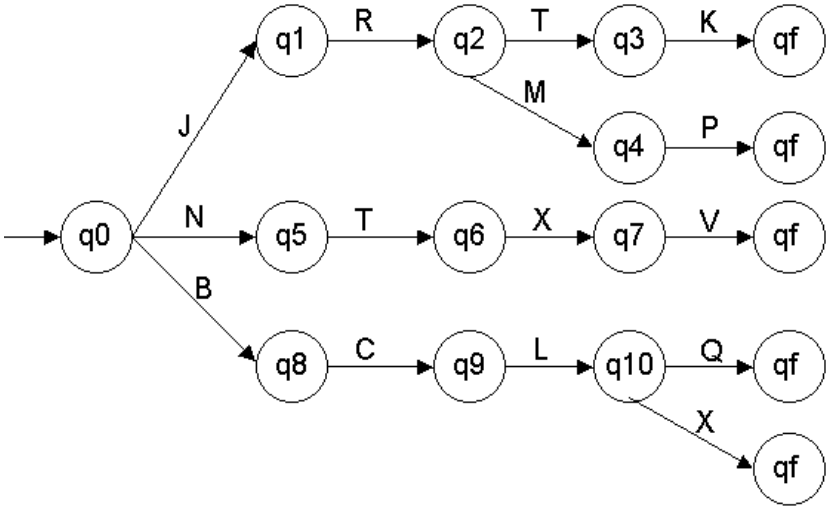

FIGURA 3.4 – Grafo do Autômato Finito Determinístico

3.3 Geração da Árvore Digital

Analisando o grafo do autômato finito determinístico gerado, ele satisfaz às propriedades de uma árvore *m-ária*. Uma árvore *m-ária* é um tipo de grafo sem ciclos e tem as seguintes propriedades:

- Existe exatamente um vértice (nó), chamado raiz, que não recebe nenhum arco como entrada;
- Cada nó, exceto a raiz, recebe exatamente um arco como entrada;
- Existe um caminho único da raiz até cada nó.

Árvores *m-árias* também são chamadas de árvores digitais. Uma árvore digital é uma estrutura de dados padrão para representação de conjuntos de cadeias de caracteres sobre um alfabeto [KNU73]. São utilizadas em algoritmos de busca de chaves de tamanho variável. A chave não é tratada como um elemento único, indivisível. Assume-se que cada chave é constituída de um conjunto de caracteres ou dígitos definidos em um alfabeto apropriado [TEN95]. A utilização da árvore digital neste trabalho não é para busca de chaves, mas sim para construir um grafo não cíclico com os exemplos de um conjunto de cadeias de caracteres (palavras) de uma linguagem regular. Tomando como exemplo o mesmo conjunto de instâncias {BCLQ, BCLX, JRMP, JRTK, NTXV}, a árvore digital resultante está representada na Figura 3.5. A principal diferença é que o conjunto deve estar ordenado.

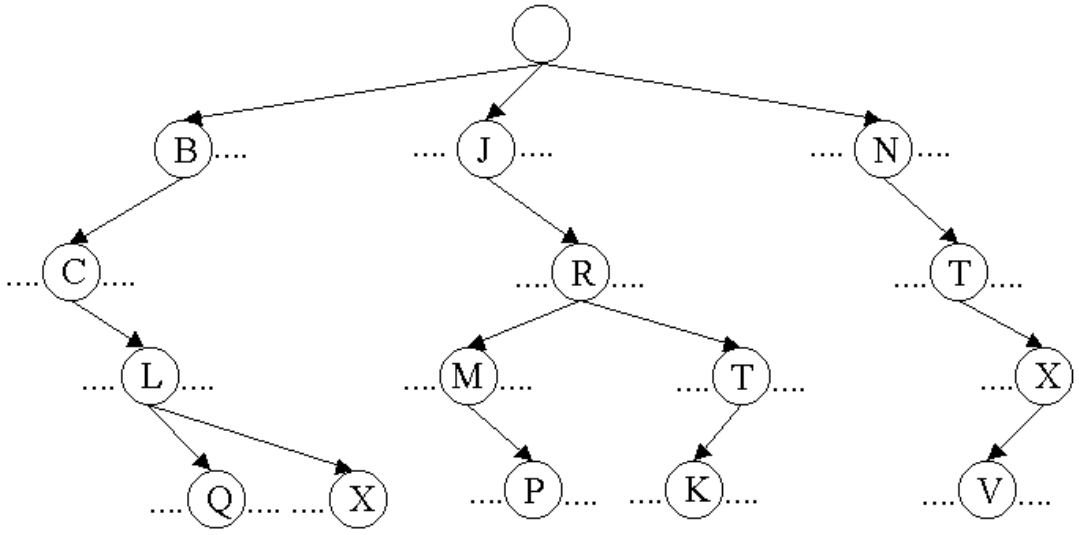

FIGURA 3.5 – Exemplo de árvore digital

Seja $S = \{s_1, s_2, ..., s_n\}$ um conjunto de n chaves em que cada s_i é formada por uma seqüência de dígitos d_j. Existe em S um total distinto de caracteres m, cujo conjunto forma o alfabeto D. Os dígitos do alfabeto admitem ordenação $d_1 < d_2 < ... < d_m$. No exemplo da Figura 3.5, S é o conjunto de exemplos R, ordenados. Se as palavras forem decompostas em letras, o valor de m é 13. Uma árvore digital para S é uma árvore *m-ária* T, não vazia, tal que:

- Se um nó v é o *j-ésimo* filho de seu pai, então v corresponde ao dígito d_j do alfabeto D, $1 < j < m$;
- Para cada nó v, a seqüência de dígitos definida pelo caminho da raiz de T até v corresponde a um prefixo de algum dos exemplos de S.

O dígito correspondente a cada nó está representado no próprio nó, depois do arco que chega de seu pai. A primeira condição é satisfeita, pois os dígitos *B, C, J, K, L, M, N, P, Q, R, T, V, X,* aparecem, respectivamente, em nós correspondentes ao

primeiro, segundo, terceiro até o décimo terceiro filho de seus pais. A segunda condição também é satisfeita, pois percorrendo o caminho desde a raiz até qualquer nó, corresponde a um prefixo de um dos exemplos do conjunto *S*.

Na implementação da árvore digital, cada nó da árvore *m-ária T*, apontado por *pt (pt ≠ nil)* possui *m* filhos ordenados, apontados por *ponteiro(pt)[1], ..., ponteiro(pt)[m]*. Se algum *i-ésimo* filho desse nó está ausente, então *ponteiro(pt)[i]* é nulo (*nil*). Se o nó for terminal de alguma chave, então *info(pt) = terminal*; caso contrário *info(pt)=não terminal*. Uma cadeia de caracteres *x* a ser pesquisada/inserida na árvore possui *k* caracteres (dígitos) denotados por *d[1], ...,d[k]*.O parâmetro *pt* indica o nó corrente da árvore, *l* indica o tamanho do maior prefixo de *x* e *a* indica se a cadeia de caracteres foi encontrada *(a=1)* ou não *(a=0)*. O alfabeto neste exemplo está representado por *D={B, C, J, K, L, M, N, P, Q, R, T, V, X}*. Os algoritmos utilizados para pesquisa e inserção na árvore digital estão demonstrados a seguir, e são conhecidos na literatura [TEN95].

algoritmo 3.1 – Pesquisa em árvore digital

```
procedimento pesqdig(x, pt, l, a)
    se l < k então   {k é o tamanho da cadeia de caracteres}
       seja j a posição de S(l + 1)     -- D =  {B,C,J,K,L,M,N,P,Q,R,T,V,X}
       se ponteiro(pt)[j] <> nil  então
          pt := ponteiro(pt)[j]; l := l + 1;
          pesqdig(x, pt, l, a)
    senão
          se info(pt) = terminal então a := 1;
```

algoritmo 3.2 – Inserção em árvore digital

```
pt := ptraiz; l:=a:=0;
pesqdig(x, pt, l, a)
se a = 0 então
   para h = l + 1,...,k faça
         seja j a posição de D(h)
         alocar (ptz);
         para i = 1,...,m faça
               ponteiro(ptz)[i] := nil;
               ponteiro(pt)[j] := ptz;
               info(ptz) := não terminal;
               pt := ptz;
         info(ptz) := terminal;
senão “inclusão inválida”   .
```

3.4 Geração da Expressão Regular a partir de um AFD

Apesar de não existir prova da geração de uma expressão regular a partir de um autômato finito, um autômato finito é equivalente a um grafo e a partir de um grafo é possível gerar a expressão regular, segundo o teorema de Kleene [HOP79]. Um grafo consiste em um conjunto de vértices (nós) V e de um conjunto de arcos (arestas) E ("*edge"* em inglês). Um arco é um par ordenado de nós (v, w), onde v é o nó origem e w é o nó destino ($v \rightarrow w$) [AHO87]. Um grafo pode conter ciclos, quando existe um caminho onde a origem e destino do arco são um único nó. Pela definição, um autômato finito pode ser representado através de um grafo. Segundo o teorema de Kleene, a geração da expressão regular, a partir de um grafo de transição, executa os seguintes passos:

- O processo tem início tomando-se qualquer grafo de transição;
- O primeiro passo é transformar o grafo inicial em um outro grafo equivalente com apenas um estado inicial e um estado final, caso haja mais que um;
- Nos próximos passos, eliminar alguns estados ou arcos transformando o grafo em um outro equivalente, através de um processo iterativo. Para isso, deve-se trocar o conjunto de caracteres do arco por uma expressão regular correspondente;
- Ao final, obtém-se um grafo apenas com o estado final e o inicial. No arco que liga os dois estados estará a expressão regular.

3.5 Resumo do Capítulo

Neste capítulo mostrou-se que as instâncias dos dados no banco de dados podem ser representadas por um autômato finito determinístico não cíclico. O grafo do autômato finito respeita as propriedades de uma árvore *m-ária*, também conhecida como árvore digital. O uso da estrutura de árvores digitais, para inclusão das instâncias dos dados no banco serve para geração da expressão regular. Isto é feito através do caminhamento em pré-ordem da árvore. O algoritmo que executa esta tarefa está descrito na subseção 4.3.1, a qual trata da geração de expressões regulares específicas.

Outra abordagem para geração de expressões regulares é através de inferência gramatical. O objetivo da inferência gramatical é encontrar algoritmos capazes de obter uma gramática para uma linguagem a partir de um conjunto de caracteres desta linguagem [GON78]. A partir da gramática regular gerada é possível obter a expressão regular correspondente. Três algoritmos são apresentados por [GON78, SCH92] para geração de gramáticas a partir de um conjunto de cadeias de caracteres. Esta abordagem não foi adotada porque não foi suficientemente pesquisada para utilização de seus algoritmos.

4 Método de Construção da Ontologia

Este capítulo apresenta um método para a construção semi-automática da ontologia utilizada pelo extrator de dados do grupo DEG, a partir de um banco de dados relacional já existente. As etapas para criação serão apresentadas utilizando como exemplo a ontologia empregada no estudo de caso do capítulo 5. Os algoritmos apresentados neste capítulo foram implementados para este trabalho.

4.1 Arquitetura Proposta para Geração da Ontologia

Com o objetivo de minimizar o processo manual de construção da ontologia, o método proposto irá auxiliar o usuário em sua construção. O processo é dividido em duas partes: engenharia reversa do modelo relacional para o modelo OSM, e geração de expressões regulares a partir das instâncias dos dados no banco de dados para uso nos *"data frames"*. O processo é semi-automático porque o usuário pode escolher as tabelas do banco e decidir qual tipo de expressão regular utilizar. A Figura 4.1 mostra a arquitetura do método proposto, com enfoque principal na área tracejada.

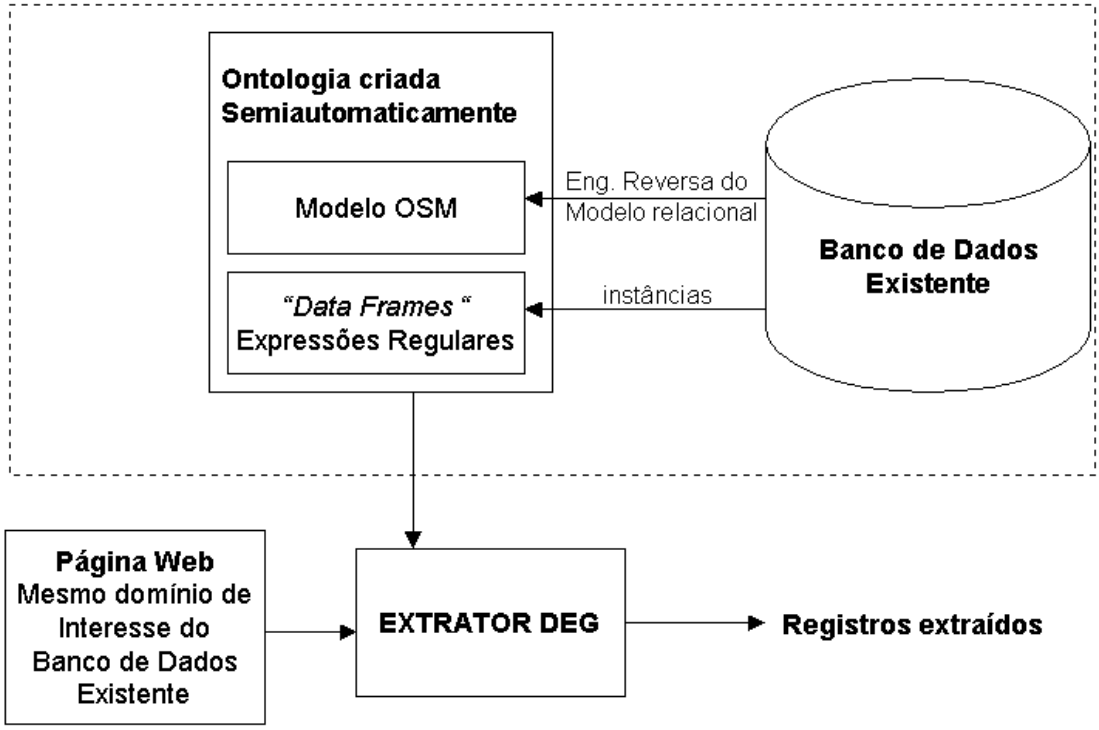

FIGURA 4.1 – Arquitetura do método proposto

O domínio de problema escolhido para demonstrar o processo de criação da ontologia é na área de livrarias virtuais. As páginas das livrarias disponibilizam informações sobre livros em blocos de registros. Existe também um banco de dados com informações sobre livros, base para construção da ontologia. Explicação mais detalhada sobre o domínio de aplicação escolhido e sobre os resultados da extração, são tratados no capítulo 5.

4.2 Construção do Modelo Conceitual OSM

A primeira etapa na construção da ontologia é o processo de engenharia reversa a partir do modelo relacional. Este processo adota as regras clássicas de engenharia reversa definidas na literatura [BAT92] para a abordagem ER e as adapta para a abordagem OSM. Somente os conceitos OSM utilizados no processo de extração de dados serão considerados. Estes conceitos estão descritos no capítulo 2.

O processo de engenharia reversa objetiva a transformação de um modelo de implementação para um modelo conceitual, descrevendo sua especificação de forma abstrata. O modelo de implementação, do qual o processo se inicia, é um banco de dados relacional Oracle. A Tabela 4.1 mostra uma comparação das terminologias entre as abordagens ER e OSM. Esta comparação permite entender as construções resultantes do modelo OSM correspondentes a cada tabela do modelo relacional. Como exemplo, se na engenharia reversa para o modelo ER, uma tabela pode corresponder a uma entidade, então, na engenharia reversa para o modelo OSM, uma tabela pode corresponder a um conjunto de objetos não léxicos, e assim sucessivamente para com as outras terminologias.

TABELA 4.1 – Comparação das terminologias básicas entre abordagem ER e OSM

Entidade Relacionamento (ER)	*Object-oriented System Model (OSM)*
Entidade	Conjunto de objetos não léxicos
Atributo	Conjunto de objetos léxicos
Relacionamento entre entidades	Conjunto de relacionamentos
Cardinalidade	Restrição de participação
Chave primária	---
Chave estrangeira	---

Na abordagem OSM, o modelo conceitual pode ser representado de forma gráfica ou textual. O primeiro passo, nesta etapa de engenharia reversa, é selecionar as tabelas e colunas do banco de dados, necessários para o processo de extração. Esta tarefa é realizada pelo usuário. A partir do domínio de problema definido, as tabelas do banco de dados que contém as informações sobre livros estão relacionadas na Figura 4.2. A tabela `LIVRO` contém informações cadastrais do livro, como o título do livro, ano de publicação, editora, etc. O campo *editora* na tabela `LIVRO` não pertence a uma chave estrangeira, sendo livre seu preenchimento. Isto faz parte da regra de negócio do sistema existente. As demais tabelas são `ASSUNTO`, que armazena a classificação dos livros por assunto, `AUTOR` que armazena as informações cadastrais do autor e a tabela associativa `AUTOR_LIVRO` que armazena a relação entre livros e autores.

```
Tabela: LIVRO
Name                           Null?    Type
------------------------------ -------- ----
NUM_LIVRO                      NOT NULL NUMBER(6)
DES_LOCAL1                              VARCHAR2(10)
DES_LOCAL2                              VARCHAR2(10)
ANO_PUBLICACAO                          NUMBER(4)
QTD_EXEMPLARES                          NUMBER(2)
DES_TITULO                              VARCHAR2(100)
NUM_EDICAO                              NUMBER(2)
DES_CIDADE                              VARCHAR2(20)
DES_EDITORA                             VARCHAR2(50)
COD_ASSUNTO                             NUMBER(3)
VAL_PRECO                               NUMBER(11,2)
Chave primária: NUM_LIVRO
Chave estrangeira: COD_ASSUNTO referencia tabela ASSUNTO

Tabela: ASSUNTO
Name                           Null?    Type
------------------------------ -------- ----
COD_ASSUNTO                    NOT NULL NUMBER(3)
DES_ASSUNTO                             VARCHAR2(40)
Chave Primária: COD_ASSUNTO

Tabela: AUTOR
Name                           Null?    Type
------------------------------ -------- ----
COD_AUTOR                      NOT NULL NUMBER(5)
NOM_AUTOR                               VARCHAR2(50)
Chave Primária: COD_AUTOR

Tabela: AUTOR_LIVRO
Name                           Null?    Type
------------------------------ -------- ----
COD_AUTOR                      NOT NULL NUMBER(5)
NUM_LIVRO                      NOT NULL NUMBER(6)
Chave Primária: COD_AUTOR e NUM_LIVRO
Chave estrangeira: COD_AUTOR referencia tabela AUTOR
Chave estrangeira: NUM_LIVRO referencia tabela LIVRO
```

FIGURA 4.2 – Tabelas Existentes no Banco de Dados

Considerando que a escolha dos dados a extrair das páginas das livrarias seja o título do livro, o ano de edição, a editora e o nome do autor ou autores, o usuário deve selecionar, a partir do esquema apresentado anteriormente, as tabelas e colunas correspondentes. As tabelas envolvidas são `LIVRO`, `AUTOR` e `AUTOR_LIVRO`. Da tabela `LIVRO`, as colunas envolvidas são `DES_TITULO`, `ANO_PUBLICACAO` e `DES_EDITORA`. Da tabela `AUTOR`, a coluna envolvida é `NOM_AUTOR`. A tabela `AUTOR_LIVRO` é necessária porque ela é o relacionamento entre as tabelas `LIVRO` e `AUTOR`. A geração do modelo conceitual OSM deve seguir algumas regras a partir da análise da composição da chave primária de cada tabela. Estas regras são descritas a seguir:

- *Chave primária composta por mais de uma chave estrangeira*. Esta situação implementa um conjunto de relacionamentos entre os conjuntos de objetos não léxicos, com restrição de participação `(0:*)` para `(0:*)`, ou seja, cada conjunto de objetos não léxicos pode participar do relacionamento nenhuma ou várias vezes. Comparando com a

abordagem ER, este seria um relacionamento n:n. A tabela AUTOR_LIVRO é um exemplo desta situação. Sua chave primária é composta pelas colunas COD_AUTOR e NUM_LIVRO, que são chaves estrangeiras para as tabelas AUTOR e LIVRO respectivamente;

- *Toda chave primária é uma chave estrangeira*. Quando uma tabela possui uma chave primária em que todas colunas são chaves estrangeiras, será implementado um conjunto de objetos não léxicos de especialização. A tabela referenciada por estas chaves estrangeiras é implementada como um conjunto de objetos não léxicos de generalização. O modelo OSM implementa esta situação da mesma maneira que o modelo ER;
- *Demais casos*. Quando uma tabela possui uma chave primária que não obedece à primeira regra (chave composta de múltiplas chaves estrangeiras) e nem à segunda regra (chave primária é toda uma chave estrangeira), a tabela será implementada como um conjunto de objetos não léxicos. As tabelas LIVRO e AUTOR são exemplos desta situação. Tanto a chave primária da tabela LIVRO quanto a chave primária da tabela AUTOR não são compostas por chaves estrangeiras.

Além da implementação das tabelas em conjuntos de objetos não léxicos, a engenharia reversa do modelo relacional precisa implementar as colunas do banco de dados no modelo conceitual OSM. Uma das principais características do modelo OSM é não possuir atributos. Não há necessidade de distinção entre atributos e objetos. Todas as "coisas" do mundo real são relacionadas como objetos, e as relações entre essas "coisas" são denominadas como relacionamentos [EMB98a]. Baseado neste conceito, a geração do modelo conceitual segue a seguinte regra para colunas:

- *Colunas que não fazem parte de chaves primárias e estrangeiras*. As colunas que não pertencem à chave primária da tabela nem às chaves estrangeiras são implementadas como conjuntos de objetos léxicos. Como exemplo, as colunas definidas neste exemplo a partir das tabelas LIVRO e AUTOR se encaixam nesta regra.

A restrição de participação, na representação gráfica do modelo OSM apresentada na Figura 4.3, é representada pelos números nas extremidades de um relacionamento (linhas ligando dois objetos). Como exemplo, considere o relacionamento entre o conjunto de objetos não léxicos LIVRO e o conjunto de objetos léxicos EDITORA. A restrição de participação (0:1), do lado do conjunto de objetos LIVRO, indica que um livro não precisa participar do relacionamento. A restrição de participação (1:*), do lado do conjunto de objetos EDITORA, indica que uma editora participa uma ou mais vezes no relacionamento. As seguintes regras foram estabelecidas para implementação dos relacionamentos entre os conjuntos de objetos léxicos (colunas no modelo relacional) e os conjuntos de objetos não léxicos:

- *Colunas obrigatórias*. Se uma coluna no modelo relacional é obrigatória (NOT NULL), então a restrição de participação do lado do conjunto de objetos não léxico será (1:1) e a restrição de participação do lado do conjunto de objetos léxicos será (1:*). Voltando ao exemplo entre os conjuntos LIVRO e EDITORA, se a coluna DES_EDITORA na tabela LIVRO fosse obrigatória, a restrição de participação do lado do conjunto de objeto LIVRO seria (1:1);

- *Colunas opcionais*. Se uma coluna no modelo relacional não é obrigatória (NULL), então a restrição de participação do lado do conjunto de objetos não léxicos será (0:1) e a restrição de participação do lado do conjunto de objetos léxicos será (1:*). Esta situação ocorre com todas as colunas selecionadas das tabelas porque nenhuma delas é obrigatória;
- *Colunas pertencentes a chaves únicas*. Se uma coluna no modelo relacional pertencer a uma chave única, então a restrição de participação do lado do conjunto de objetos léxicos será 1. Como exemplo, supondo que o título do livro fosse uma coluna pertencente a uma chave única, seu correspondente conjunto de objetos léxicos não poderia participar várias vezes no relacionamento, estando restrito a uma ocorrência.

Segundo as regras acima, pode-se observar que a restrição de participação do lado do conjunto de objetos léxicos é sempre (1:*) desde que o conjunto não seja derivado de uma coluna pertencente a uma chave única. Justificando este fato através do conjunto de objetos léxicos ANO, este pode participar no relacionamento com LIVRO no mínimo uma vez ou no máximo várias vezes, ou seja, em um determinado ano vários livros podem ter sido publicados. Do lado contrário, no conjunto LIVRO, um livro participa no máximo uma vez porque um livro tem um único ano de publicação.

Definidas as regras para geração do modelo conceitual OSM, a Figura 4.3 mostra o modelo relacional à esquerda e o modelo gráfico OSM à direita, para as três tabelas selecionadas. No modelo OSM, os retângulos com linhas contínuas representam os conjuntos de objetos não léxicos e os retângulos com linhas tracejadas representam os conjuntos de objetos léxicos, com relacionamentos entre si. O nome do conjunto de objetos léxicos no modelo OSM difere do nome do atributo no modelo ER porque o usuário precisa eliminar o caractere sublinhado.

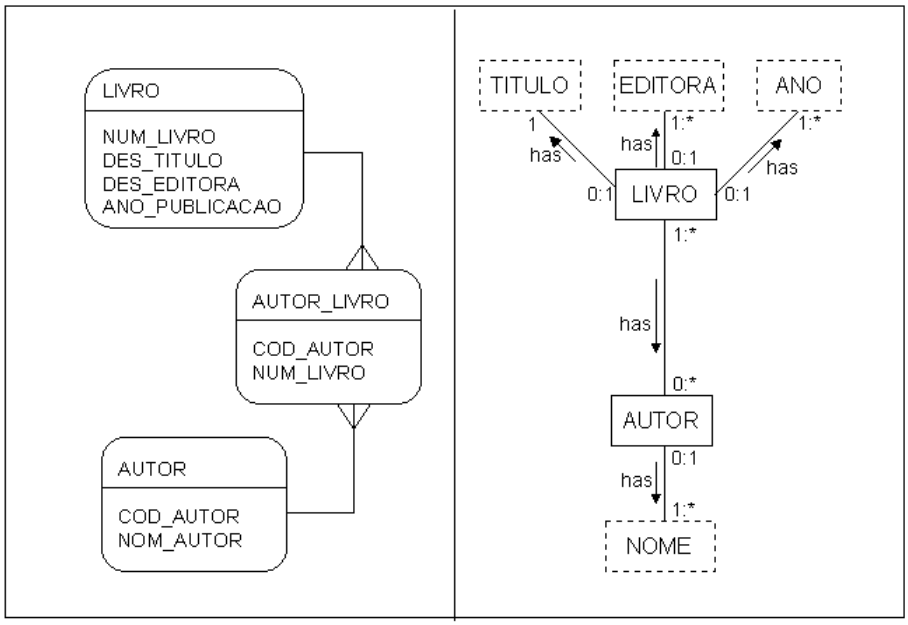

FIGURA 4.3 – Modelo Relacional e Modelo OSM correspondente

Outras definições de restrição existem na literatura sobre a abordagem OSM [EMB98a], como restrição de cardinalidade, restrição de co-ocorrência e restrição de integridade referencial, mas não serão tratadas aqui por serem irrelevantes na ontologia gerada para extração de dados.

Além da representação gráfica, a abordagem OSM possui a representação textual equivalente [LID95], a qual é utilizada como sintaxe na construção da ontologia. Esta sintaxe é um requisito do extrator do grupo DEG no processo de extração de dados das páginas desejadas. O Segundo passo nesta etapa de engenharia reversa, depois de selecionada as colunas, é analisar o esquema do banco de dados, aplicando as regras definidas para geração do modelo conceitual em sua representação textual. A partir das regras definidas, foi implementado um algoritmo que analisa as tabelas e colunas selecionadas gerando a sintaxe textual OSM equivalente. Este algoritmo foi implementado neste trabalho utilizando a linguagem Pl/Sql da Oracle [URM96].

A partir da tabela livro, implementada como um conjunto de objetos não léxicos, a representação OSM textual equivalente é:

```
[LIVRO -> OBJECT]
```

Para a tabela `AUTOR_LIVRO`, implementada como um conjunto de relacionamentos entre `LIVRO` e `AUTOR`, a representação OSM textual é a seguinte:

```
LIVRO[1:*] has AUTOR[0:*]
```

Cada coluna selecionada nas tabelas é implementada como um conjunto de objetos léxicos. É necessário, também, que um conjunto de relacionamentos seja mapeado com o conjunto de objetos não léxicos associado. A representação OSM textual para os conjuntos de objetos léxicos é a seguinte:

```
LIVRO[0:1] has EDITORA[1:*]
```

A primeira etapa de geração da ontologia é concluída com a geração da ontologia no modelo textual, conforme a Figura 4.4.

```
 1: LIVRO [-> OBJECT]
 2:
 3: LIVRO[0:1] has TITULO[1:*]
 4: TITULO matches [100]
 5: END;
 6:
 7: LIVRO[0:1] has EDITORA[1:*]
 8: EDITORA matches [20]
 9: END;
10:
11: LIVRO[0:1] has ANO[1:*]
12: ANO matches [4]
13: END;
14:
15: LIVRO[1:*] has AUTORNOME[0:*]
16: AUTORNOME matches [50]
17: END;
```

FIGURA 4.4 – Representação OSM textual da ontologia

4.3 Geração das expressões regulares

Depois de construir o modelo conceitual OSM, no formato textual, a segunda etapa é definir para cada conjunto de objetos léxicos as expressões regulares nos "*data frames"*. A idéia é gerar as expressões regulares o mais automático possível através da análise dos tipos das colunas no esquema do banco de dados combinado com a análise das instâncias dos dados neste mesmo banco de dados.

Como explicado no capítulo anterior (algoritmos para reconhecimento de linguagens), uma das características dos algoritmos de inferência gramatical é que eles constroem gramáticas ou máquinas de estado que reconhecem exatamente um conjunto de palavras fornecido como exemplo. No caso das expressões regulares destinadas à extração de dados, esta situação nem sempre será necessária.

No relacionamento entre o banco de dados e os documentos a serem processados pelo extrator de dados, duas situações podem ocorrer:

- O conjunto de valores de um objeto léxico que pode existir no documento é um subconjunto dos valores que aparecem na coluna do banco de dados. Um exemplo poderia ser unidades da federação. Os documentos contêm apenas unidades da federação que estejam cadastradas em uma tabela do banco de dados. *Neste caso, as expressões regulares recuperam exatamente os dados que estão no banco de dados*;
- Os documentos podem conter valores de objetos léxicos que não estejam armazenados no banco de dados. Este é um caso no qual os documentos possuem dados que, após o processo de extração, serão incluídos no banco de dados. Um exemplo poderia ser a extração de dados de documentos legados para posterior inclusão no banco. Outro exemplo poderia ser a extração de dados para complementar as tabelas do banco de dados, como preços de produtos concorrentes. *Neste caso, as expressões regulares são mais genéricas, recuperando não só os dados do banco de dados como também dados semelhantes*.

Considerando estas duas situações, a geração das expressões regulares, a serem inseridas nos "*data frames"* da ontologia, necessitam de pelo menos dois tipos de algoritmos. Um que gera uma expressão regular que define exatamente o conjunto de léxicos que podem aparecer no documento, e outro que gera uma expressão regular que define um conjunto maior de dados que podem aparecer no documento.

Com base nos estudos apresentados no capítulo 3, foram construídos alguns algoritmos específicos para geração das expressões regulares, utilizadas para extração de dados na ontologia. Os algoritmos para as duas situações são apresentados nas subseções 4.3.1 e 4.3.2.

4.3.1 Expressão regular específica

Este tipo de expressão regular é utilizado quando se deseja recuperar exatamente os dados que estão no banco de dados. É aplicada a colunas que possuem poucas instâncias, no entanto, depende de uma análise do usuário que está apoiando o processo de construção da ontologia para decidir a forma de criação da expressão. Um exemplo de aplicação para este tipo de expressão poderia ser com as instâncias de editora. O número de instâncias não é grande e, portanto, uma expressão regular que recupere exatamente os dados do conjunto seria interessante. Na ontologia criada pelo grupo

DEG, utilizam-se três formas de expressão regular para recuperar dados exatos: através da inclusão de dicionários na ontologia, utilização do dicionário em arquivos e uso de expressões padrões. Na abordagem de criação semi-automática da ontologia, estas três formas são utilizadas também, com a diferença de que o usuário não precisa escrever manualmente a sintaxe do *"data frame"*. A ferramenta de apoio à construção da ontologia executa esta tarefa de modo automático. Além das três formas citadas, uma quarta forma foi criada, que é a utilização de uma estrutura de árvore digital que permite a construção de uma expressão regular mais otimizada do que a pura inclusão do dicionário na ontologia. A seguir, estão as descrições das três formas de geração da expressão regular, acrescida da quarta forma proposta por este trabalho.

Inclusão de dicionários na ontologia. O dado que se deseja extrair do documento semi-estruturado é incluído no *"data frame"* do objeto léxico considerado. Na construção manual, os valores que se desejam extrair são incluídos após análise do documento. Na construção semi-automática da ontologia, o dicionário é incluído por um processo que lê as instâncias do banco de dados e as insere no *"data frame"* de acordo com a sintaxe exigida. Como exemplo, considere a coluna `EDITORA` do banco de dados. O programa lê as instâncias de editora, gerando o *"data frame"* com a seguinte sintaxe:

```
LIVRO[0:1] has EDITORA[1:*]
EDITORA matches [20]
Constant
   {extract "\bACADEMICA\b";},
   {extract "\bADVANCED\b";},
   {extract "\bBERKLEY\b";},
    :
   {extract "\bVEREDA\b";};
END;
```

Utilização do dicionário em arquivos. Em vez de incluir o dicionário na ontologia, ocasionando um incremento em seu tamanho, o dicionário pode ser gravado em um arquivo com extensão *".dict"*. Este é um arquivo no formato texto que contém as instâncias do objeto léxico a serem extraídas do documento. No processo de construção semi-automática da ontologia, este arquivo é gerado automaticamente pela ferramenta, se esta for a opção escolhida pelo usuário. Considerando o mesmo exemplo com a coluna `EDITORA`, mas desta vez utilizando um arquivo de dicionário, a sintaxe do *"data frame"* é:

```
LIVRO[0:1] has EDITORA[1:*]
EDITORA matches [20]
Constant
   {  extract EDITORA  };
   {
     EDITORA case insensitive;
     Filename "editora.dict";
   };
END;
```

Utilização de expressões regulares padrão. Para determinadas colunas do banco de dados não é necessário um processo de análise do tipo de dados e instâncias. Em colunas cujo conteúdo seja uma data, hora, preço ou ano, pode-se utilizar uma expressão regular padrão onde seja informado apenas o formato como o dado se encontra no documento semi-estruturado. São valores bastante específicos que não causam confusão com outros dados, mesmo sabendo que existem exceções. Exemplificando, considere a coluna de ano de edição do livro, com quatro dígitos numéricos. Poderia ser utilizada uma expressão regular para extrair exatamente esta informação. A inclusão da expressão no *"data frame"* é feita pelo usuário, não de forma manual, mas escolhendo a expressão a partir de uma biblioteca pronta de expressões para esse fim. A sintaxe do *"data frame"* para o objeto léxico ano de publicação é:

```
LIVRO [0:1] has ANO [1:*];
ANO matches [4]
   constant
      {  extract "19[4-9][0-9]";  };
    constant
      {  extract "200[0-2]";      };
end;
```

Além do exemplo com ano de publicação, podem existir outros tipos de dados que podem ser utilizados na ontologia com a utilização de expressões regulares pré-definidas.A Tabela 4.2 mostra alguns destes exemplos.

TABELA 4.2 – Exemplos de Expressões Regulares Padrão

Regular Expression Template	Format
"(0[1-9])|(1[0-2])/(0?[1-9]|[12]\d)|3[01]"	MM/DD {month/day}
"(\d\d|(1[89]|2[01])\d\d)"	YYYY {year)
([1-9]|[01]\d|2[0-4]):([0-5]\d)	HH:MM {hour:min}
\bpage\s*(no\.?\s*)?\d+\b	"page no. 9.." or "page no 9.."
"(jan|feb|mar|apr|may|....|dec)"	{strmonth}

Utilização de estrutura de Árvore Digital. Esta forma de geração da expressão regular é baseada em um algoritmo bastante conhecido para construção de árvores digitais apresentado na seção 3.3 do capítulo anterior [TEN95]. Utilizando como exemplo a coluna `EDITORA` da tabela `LIVRO`, suas diferentes instâncias serão inseridas na estrutura de árvore digital para posterior geração da expressão regular. Para inclusão na árvore digital, considere que as diferentes instâncias de editora sejam formadas pelo conjunto `S = {ACADEMICA, ADVANCED, ARTENOSSA, BERKLEY, BERTRAND}`. O alfabeto para este conjunto de valores é `D={A, B, C, D, E, I, K, L, M, N, O, R, S, T, V, Y}`. O valor de *m* (total distinto de caracteres no conjunto *S*) é igual a 16, obtido pela decomposição das palavras do conjunto *S* em letras. O algoritmo de pesquisa/inclusão em árvores digitais foi implementado utilizando-se a linguagem Pl/Sql da Oracle [URM96]. A árvore digital resultante do processo de inclusão é mostrada na Figura 4.5.

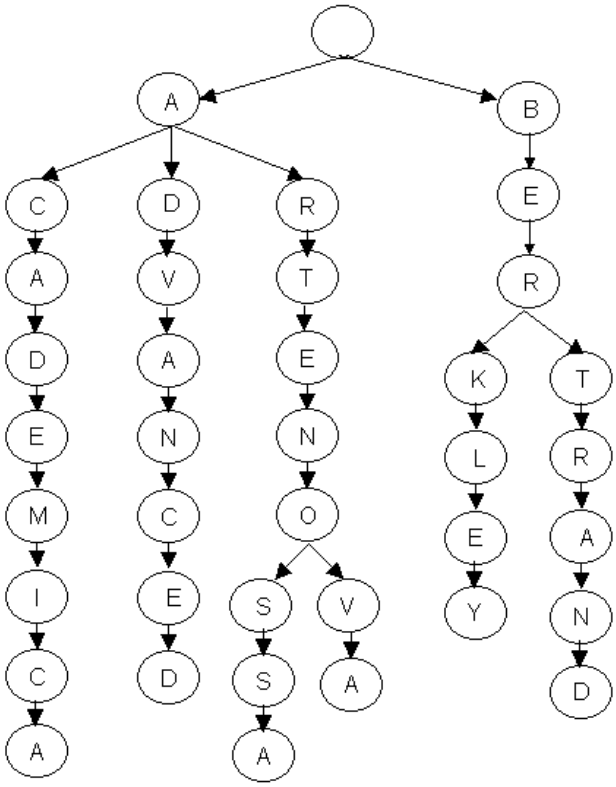

FIGURA 4.5 – Árvore Digital a partir de um conjunto de nomes de editoras

Depois das instâncias terem sido incluídas na árvore digital, o próximo passo será percorrer a árvore para geração da expressão regular. Entende-se por percorrer a árvore a tarefa de visitar (ler) todos os nós da árvore, com o objetivo de consultar a informação nele contida. Existem três principais ordens de caminhamento em árvores: pré-ordem, pós-ordem e central. Neste trabalho, o algoritmo utilizado para percorrer a árvore é chamado de caminhamento em pré-ordem ("*preorder traversal"*). O caminhamento em pré-ordem executa, recursivamente, os seguintes passos [AHO74]:

- Visita-se a raiz;
- Visita-se a sub-árvore da esquerda;
- Visita-se a sub-árvore da direita.

O resultado do caminhamento em pré-ordem na árvore digital apresentada na Figura 4.5 é o seguinte:

A-CADEMICA-DVANCED-RTENO-SSA-VA-BER-KLEY-TRAND

Com base no caminhamento em pré-ordem, foi criado um novo algoritmo que gera a expressão regular na sintaxe exigida para a ontologia de extração. A este algoritmo foi dado o nome de "*digi-tree"*. Este algoritmo foi implementado neste trabalho, através da linguagem Pl/Sql [URM96], e gera como saída a expressão regular que denota as cadeias de caracteres que foram inseridas na árvore digital. O algoritmo está descrito a seguir:

algoritmo 4.1 – Algoritmo para geração da expressão regular

```
procedimento GeraExpReg
   expressao := ´´;
   PercorreArvore (ptraiz);

procedimento PercorreArvore (v:nó)
cont:=0;
inicio
   visita nó(v)
   se ContaFilhos(v) = 1 então
      pt := ponteiro(v)[id filho];  visita nó(pt);
      expressao := expressao + info(pt);
      aux := aux + info(pt);
      se terminal(pt) = ´S´ e ContaFilhos(pt) <> 0 então
         expressao := expressao + ´|´ + aux;   aux := nil;
      PercorreArvore(id(pt));
   senão se ContaFilhos(v) > 1 então
      expressao := expressao + ´(´;
      para h = 1,..., m faça  {m é o total de filhos de um nó}
         pt := ponteiro(v)[h];  visita nó(pt);
         cont := cont + 1;
         expressao := expressao + info(pt);
         aux := aux + info(pt);
         se terminal(pt) = ´S´ e ContaFilhos(pt) <> 0 então
            expressao := expressao + ´|´ + aux;
            aux := nil;
         PercorreArvore(id(pt));
         se cont = m então  expressao := expressao + ´)´;
                     senao  expressao := expressao + ´|´;
fim;

Procedimento ContaFilhos (v1)
num_filhos := 0;
inicio
   para h = 1,..., m faça  {m é o total de filhos de um nó}
        se ponteiro(v)[h] <> nil então
           num_filhos := num_filhos + 1;
   retorna (num_filhos);
fim;
```

Considerando, então, o conjunto S = {ACADEMICA, ADVANCED, ARTENOSSA, BERKLEY, BERTRAND}, que foi inserido na estrutura de árvore digital, a expressão regular gerada pelo algoritmo *"digi-tree"* está inserida na sintaxe do *"data-frame"* para o conjunto léxico EDITORA, a seguir:

```
LIVRO[0:1] has EDITORA[1:*]
EDITORA matches [20]
Constant
   {extract
    "(A(CADEMICA|DVANCE|RTENO(SSA|VA))|BER(KLEY|TRAND))";
   };
END;
```

Para este conjunto de instâncias, a expressão regular é pequena, mas se o número de instâncias a ser reconhecida aumentar, a expressão irá aumentar também. O extrator do grupo DEG impõe limites de implementação para o tamanho da expressão regular gerada.

4.3.2 Expressão Regular Genérica

Este tipo de expressão regular é utilizado quando se deseja recuperar não somente os dados que estão no banco de dados, mas também outros dados que se assemelham com os existentes. É aplicada a colunas que possuem muitas instâncias, não sendo possível gerar uma expressão regular que reconheça todo o conjunto de dados a extrair, caso contrário a expressão regular seria muito grande. Um exemplo de uso deste tipo de expressão regular poderia ser com o nome do autor ou título do livro. Um exemplo é mostrado logo abaixo. A expressão regular é muito genérica e pode recuperar dados que não sejam do mesmo escopo que o dados desejados. Na ontologia do grupo DEG, as expressões genéricas são criadas manualmente, mas são complementadas com palavras-chave que guiam o processo de extração. As cláusulas para guiar o processo de extração serão tratadas na próxima subseção.

Na abordagem de criação da ontologia de forma semi-automática, as instâncias do banco de dados são utilizadas para construir uma expressão regular bastante simples. Ela especifica os caracteres que podem aparecer em cada posição das instâncias do banco de dados; ou então especifica os caracteres que podem aparecer em cada palavra das instâncias do banco de dados. A este algoritmo foi dado o nome de *"position-based"*, e não é baseado em nenhum algoritmo clássico da literatura.

Exemplificando, considere a coluna NOME_DO_AUTOR. Esta coluna possui muitas instâncias diferentes e é inviável incluir todos os léxicos na ontologia ou mesmo gerar a expressão regular através da árvore digital. A expressão regular ficaria muito grande e infringiria as limitações técnicas da ferramenta de extração. Supondo que o conjunto com os nomes de autores sejam os seguintes:

```
CANTU, MARCO                    SETZER, VALDEMAR W.
PRESSMAN, ROGER S.              SHIMIZU, TAMIO
RAMALHO, JOSE ANTONIO           SILBERSCHATZ, ABRAHAM
RUMBAUGH, JAMES                 YOURDON, EDWARD
SEBESTA, ROBERT W.
```

O algoritmo lê todas as instâncias de nome do autor verificando quais os diferentes caracteres que ocorrem em cada posição. Cada conjunto de caracteres que aparece entre colchetes representa os possíveis caracteres que podem aparecer nesta posição. O ponto de interrogação segue os colchetes quando o caractere na posição é opcional. O caractere espaço é representado pela expressão "\s" e "\." representa o caractere ponto. A quantidade de conjuntos entre colchetes é igual ao tamanho do maior nome, 21 neste caso. O décimo terceiro conjunto de caracteres passa a ser opcional, pois o menor nome tem 12 caracteres. Esta expressão pode recuperar outros nomes além dos que estão representados no conjunto de nomes apresentado anteriormente. A expressão regular gerada é:

```
[CPRSY][AREHIOU][BEILMNTU][ABEMRSTZ][ADEILSU][,HMORTUZ][\s,AGNOSU][\s,
CHMN][\s,AHV][\sAEJRT][ACDJLORT][ABDMOSWZ][,AEGIM]?[\sEMOR]?[ADRST]?[\
sBNR]?[\sRSTW]?[\.AOW]?[\.HN]?[AI]?[MO]?
```

Em outra alternativa, o algoritmo lê todas as instâncias de nome do autor verificando os diferentes caracteres que ocorrem em cada palavra. As palavras foram definidas como toda cadeia de caracteres entre espaços em branco. Cada conjunto de caracteres entre colchetes representa os possíveis caracteres que ocorrem em cada palavra do nome do autor. Como o menor nome é composto por duas palavras, o conjunto que representa as letras a partir da terceira palavra é opcional, representado pelo ponto de interrogação (?). O caractere (*) após cada conjunto de caracteres indica que os caracteres podem ocorrer mais de uma vez em cada palavra. Da mesma forma, esta expressão regular, apesar de menor, pode recuperar outros dados que não tem relação com nome do autor. Por exemplo, a palavra *"AAA"* estaria correta segundo esta expressão regular. A expressão regular gerada para o mesmo conjunto de nomes de autores é:

```
[ABCDEGHILMNOPRSTUYZ,]*\s[ABCDEGHIJLMORSTVW]*\s[\.AINOSTW]*?
```

Cada uma destas expressões regulares pode ser otimizada, observando-se a seqüência de caracteres que ocorrem. Por exemplo, se a expressão gerada contém todos os caracteres de A até Z, então a expressão regular pode ser otimizada com a expressão [A-Z]. A expressão regular otimizada para este último caso seria:

```
[(A-E)GHI(L-P)(R-U)YZ,]*\s[(A-E)(G-J)LMORSTVW]*\s[\.AINOSTW]*?
```

Na ontologia criada manualmente pela abordagem do grupo DEG, as expressões regulares genéricas são escritas já otimizadas, com a expressão *[A-Za-z]** para reconhecer palavras [EMB99]. Utilizando as expressões regulares propostas pela abordagem de geração semi-automática da ontologia, a sintaxe do *"data frame"* para o objeto léxico `NOME_DO_AUTOR`, com análise por caractere, seria:

```
LIVRO[1:*] has AUTOR[1:*]
AUTOR matches [30]
Constant
   {extract
"[CPRSY][AREHIOU][BEILMNTU][ABEMRSTZ][ADEILSU][,HMORTUZ][\s,AGNOSU][\s
,CHMN][\s,AHV][\sAEJRT][ACDJLORT][ABDMOSWZ][,AEGIM]?[\sEMOR]?[ADRST]?[
\sBNR]?[\sRSTW]?[\.AOW]?[\.HN]?[AI]?[MO]?";
   };
END;
```

Como segunda opção, com análise por palavra, a sintaxe do mesmo *"data frame"* para o `NOME_DO_AUTOR` seria:

```
LIVRO[1:*] has AUTOR[1:*]
AUTOR matches [30]
Constant
   {extract
"[ABCDEGHILMNOPRSTUYZ,]*\s[ABCDEGHIJLMORSTVW]*\s[\.AINOSTW]*?";
   };
END;
```

Este tipo de expressão regular pode ser utilizado não somente para colunas alfanuméricas, mas também para colunas numéricas.

4.3.3 Alterações na Ontologia para Guiar o Processo de Extração

O uso de expressões regulares genéricas na ontologia para extração de dados pode ocasionar imprecisão nos dados recuperados, ou seja, as expressões permitem que outros dados não pertencentes ao escopo desejado sejam recuperados. Para evitar este problema, a abordagem de extração de dados proposta pelo grupo DEG [EMB98, EMB99] utiliza cláusulas que ajudam o processo de extração a localizar o dado no documento semi-estruturado. Além destas cláusulas, o usuário pode alterar as expressões regulares geradas, enriquecendo-as através da análise de como o dado se encontra na página.

Duas das cláusulas que o usuário pode utilizar para enriquecer a ontologia são: *"KEYWORD"* e *"CONTEXT"*.

- *KEYWORD* – Esta cláusula pode ser utilizada para indicar ao extrator a presença de um dado no documento. Utiliza-se uma cadeia de caracteres constante. Por exemplo, para localizar o nome do autor no documento poderia ser utilizada a cadeia "nome do autor". O extrator do grupo DEG considera o valor que está imediatamente antes ou depois desta constante como o provável dado a ser extraído;
- *CONTEXT* – Esta cláusula pode ser utilizada para definir o contexto do dado que se deseja extrair. Para facilitar a localização do dado no documento, pode ser indicada alguma característica específica sobre o conteúdo do dado.

Para ilustrar o uso destas duas cláusulas, considere o *"data frame"* para extração do nome do autor. A cláusula *"keyword"* poderia ser utilizada para facilitar a localização do nome. No documento semi-estruturado, antes ou depois do dado a ser extraído, se existir uma das duas cadeias de caracteres ("autor:", "nome do autor"), o dado será extraído com maior grau de certeza de que pertence ao conjunto de objetos léxico. A alteração no *"data frame"* está destacada na sintaxe a seguir:

```
LIVRO[1:*] has AUTOR[1:*]
AUTOR matches [30]
Constant
   {extract
"[ABCDEGHILMNOPRSTUYZ,]*\s[ABCDEGHIJLMORSTVW]*\s[\.AINOSTW]*?";
   };
    keyword "\bautor:\b",
            "\bnome\s+do\s+autor";
END;
```

Da mesma forma, a cláusula *"context"* poderia ser utilizada para localizar o dado, mas agora a cadeia de caracteres não precisa estar antes ou depois do dado a ser extraído, mas sim fazer parte do dado a ser extraído. Um exemplo seria extrair os nome dos autores que contenham o nome James. A alteração no *"data frame"* está destacada a seguir:

```
LIVRO[1:*] has AUTOR[1:*]
AUTOR matches [30]
Constant
   {extract
"[ABCDEGHILMNOPRSTUYZ,]*\s[ABCDEGHIJLMORSTVW]*\s[\.AINOSTW]*?";
    context "\bJames\b"; };
END;
```

Outras situações podem exigir conhecimento adicional a respeito de como o dado está representado no documento semi-estruturado, sendo então necessário melhorar a ontologia gerada inicialmente de forma semi-automática. Além da utilização das cláusulas mencionadas anteriormente, alguns dados podem estar representados com máscaras específicas, contendo símbolos especiais. Exemplos poderiam ser códigos de produtos. Como a ontologia gerada semiautomaticamente reflete somente a informação que está armazenada no banco de dados, fica a cargo do usuário enriquecer a ontologia com os detalhes de formatação necessários para obter melhores taxas de recuperação e precisão no processo de extração.

4.3.4 Heurísticas para Sugestões de Geração das Expressões Regulares

Após a escolha das tabelas e colunas a partir das quais se deseja construir a ontologia, o processo de geração semi-automática irá sugerir o tipo de expressão regular adequado para cada tipo de coluna, conforme os tipos de expressões regulares descritos nas subseções 4.3.1 e 4.3.2. Cabe ao usuário decidir o tipo de expressão regular que ele deseja adotar, podendo ou não seguir a sugestão do processo.

As regras para determinar os tipos de expressões regulares para cada tipo de coluna foram definidas de forma heurística, de acordo com os seguintes critérios:

- Se a coluna for do tipo `DATA` ou `HORA`, sugerir o uso de expressões regulares padrão a partir da biblioteca de padrões;

- Se a coluna for do tipo `NUMÉRICA` sem casas decimais e se o número de instâncias diferentes da coluna for maior que 200, sugerir o uso do algoritmo *"position-based"* com análise por caractere, ou então uma expressão regular padrão de acordo com o tamanho da coluna. Exemplo: se a coluna for numérica de 5 posições, a expressão padrão é *[0-9]{5}*;
- Se a coluna for do tipo `ALFANUMÉRICA`, o número de instâncias diferentes da coluna for maior que 200 e se o conteúdo corresponder à descrição de algum produto ou nome, sugerir o uso do algoritmo *"position-based"*, com análise por caractere ou por palavra;
- Se a coluna for do tipo `NUMÉRICA` com casas decimais, independente do número de instâncias diferentes, provavelmente corresponde a um valor monetário ou algum dado específico. Neste caso, sugerir que seja utilizada uma expressão regular padrão para valor ou que seja criada outra expressão específica;
- Independente do tipo da coluna, se o número de instâncias diferentes da coluna for menor que 200, sugerir o uso do algoritmo "digi-tree" para geração de expressão regular que reconheça todas as instâncias. Ainda neste caso, a sugestão ao usuário será a possibilidade de inserir as instâncias na ontologia ou criar um arquivo contendo as mesmas.

4.4 Resumo do Capítulo

Neste capítulo foi apresentado e descrito o método de construção da ontologia, composto pelo processo de engenharia reversa do modelo relacional para o modelo OSM e pela geração das expressões regulares a partir das instâncias dos dados no banco de dados. Como resultado destes dois processos, a ontologia é gerada no formato textual e na sintaxe OSM, para que possa ser utilizada pelo extrator do grupo DEG.

Para geração das expressões regulares, foram implementados dois tipos de algoritmos. Um para gerar uma expressão regular que define exatamente um conjunto de dados, e outro que gera uma expressão mais genérica. O primeiro algoritmo é utilizado para construir expressões regulares que permitem recuperar exatamente os dados do banco e o segundo permite recuperar não somente os dados do banco. As expressões regulares resultantes destes algoritmos completam a ontologia gerada semiautomaticamente.

5 Estudo de Caso

Este capítulo apresenta dois estudos de caso para extrair dados de páginas de livrarias virtuais na Internet. Este domínio de problema foi escolhido porque existe um banco de dados com informações sobre livros e, em complemento, páginas na Internet com o mesmo conteúdo. O objetivo do estudo de caso é avaliar a ontologia que é gerada semiautomaticamente através do processo proposto, bem como analisar que alterações são necessárias para obter melhores resultados no processo de extração de dados.

As páginas das livrarias virtuais disponibilizam informações sobre livros em blocos de registros, seguindo os requisitos da abordagem de extração de dados do grupo DEG. As livrarias virtuais pesquisadas são a Livraria Cultura, Livraria Siciliano, Livrarias Curitiba e a Loja Virtual Submarino. O ponto de partida para geração da ontologia de extração dos livros é um banco de dados relacional Oracle, versão 7.3.4. Esta base de dados pertence à Biblioteca da Faculdade Paranaense – FACCAR, localizada no município de Rolândia – Paraná. O acervo da Biblioteca possui 17.000 títulos cadastrados, e atende aos cursos de Administração de Empresas, Contabilidade, Direito, Letras e Tecnologia em Processamento de Dados.

O estudo de caso compreende os seguintes passos:

- Inicialmente, a ontologia é gerada através do método proposto e descrito no capítulo anterior. No Anexo, está descrito um protótipo para guiar o usuário na construção da ontologia, desde a escolha das tabelas do banco de dados, até a geração das expressões regulares nos *"data fram*es", a partir das instâncias dos dados nesse mesmo banco de dados;
- Em seguida, é feita uma busca em páginas da Internet para obter o código HTML de onde os dados são extraídos. Este é um processo manual, no qual o usuário armazena a página HTML para servir como entrada ao extrator DEG;
- A ontologia e as páginas HTML são submetidas ao extrator DEG, para extração dos dados, observando-se as taxas de recuperação e precisão. Estas taxas são os parâmetros de avaliação definidos nesta abordagem de extração;
- Depois de avaliado os resultados, com a ontologia gerada semiautomaticamente, o usuário tem a opção de enriquecer a ontologia com expressões regulares para dados genéricos, como datas, horas, valores monetários. As alterações são feitas para melhorar o processo de extração e são baseadas na análise do conteúdo das páginas. A inclusão de uma palavra-chave ou outra característica que possa guiar o extrator de registros na localização do dado na página, é um exemplo de alteração.

Os dados sobre os livros que estão armazenados no banco de dados são: título do livro, assunto, ano de publicação, edição, editora, assunto, quantidade de exemplares, localização na biblioteca e também os autores do livro. O modelo conceitual do banco de dados está na Figura 5.1.

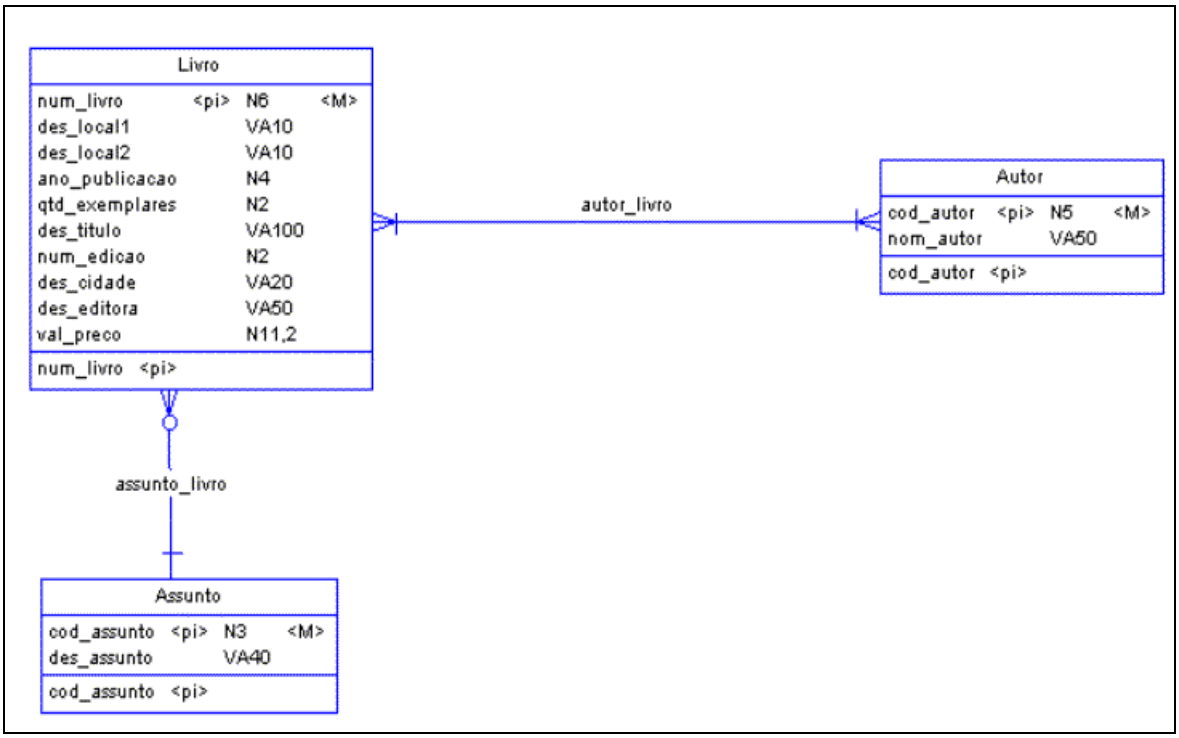

FIGURA 5.1 – Modelo conceitual do banco de dados

Foram realizados dois estudos de casos, com aplicações diferentes. O primeiro extrai, das páginas das livrarias virtuais, os títulos de livros de um determinado autor que estejam cadastrados no banco de dados, a fim de comparar preços. O segundo extrai títulos de livros que não estejam cadastrados no banco de dados. As informações sobre os livros disponibilizadas nestas livrarias virtuais são praticamente as mesmas, variando o contexto e a forma em que são apresentadas. A ontologia gerada através da abordagem do grupo DEG não depende do contexto ou forma de apresentação das páginas. Uma vez criada a ontologia, ela pode ser utilizada em diversas páginas, desde que seja do mesmo domínio de problema.

5.1 Estudo de Caso sobre Livros de um Autor

O primeiro estudo de caso é um exemplo de situação em que se têm dados sobre livros em uma base de dados e se deseja encontrá-los em páginas Web. Os dados sobre livros armazenados no banco de dados e que também existem nas páginas das livrarias são o *título do livro, autor, editora e ano de publicação*. Além destes dados, as páginas possuem dados complementares, como o preço do livro. O objetivo do primeiro estudo é a extração do preço dos livros que estão cadastrados no banco.

O processo de extração irá recuperar, das páginas das livrarias, os títulos de livros que coincidam com os títulos existentes no banco de dados e também seu preço. Esta situação pode ser aplicada a bibliotecas que queiram fazer uma pesquisa de preços dos títulos de seu acervo. O mesmo exemplo pode ser aplicado às livrarias para obter o preço dos livros de seus concorrentes. Para outro domínio de problema, este exemplo também pode ser aplicado, desde que se tenha um banco de dados com um determinado produto, seu preço, e páginas da Internet onde dados semelhantes estejam publicados.

Neste estudo de caso, foi necessário definir um conjunto de livros a ser pesquisado, pois a utilização da base de livros completa seria inviável. O critério

utilizado foi o de escolher livros de um determinado autor, que tivesse uma quantidade de livros razoável e que fosse conhecido do público em geral. Com essas características, e para não se prender a uma área muito técnica, o autor escolhido foi Jorge Amado. O processo de extração foi executado com páginas de quatro diferentes livrarias virtuais e o resultado da extração será avaliado, verificando-se a taxa de recuperação e precisão. Os livros existentes no banco de dados, de autoria de Jorge Amado, estão relacionados na Figura 5.2, e são apenas vinte e oito livros. A partir dos dados destes livros a ontologia foi construída.

TITULO	ANO	EDITORA
A MORTE E A MORTE DE QUINCAS BERRO DAGUA	1983	Record
A. B. C. DE CASTRO ALVES	1983	Record
BAHIA DE TODOS OS SANTOS	1983	Record
CACAU	1983	Record
CAPITAES DA AREIA	1983	Record
DONA FLOR E SEUS DOIS MARIDOS	1966	Martins
DONA FLOR E SEUS DOIS MARIDOS	1983	Record
FARDA, FARDAO, CAMISOLA DE DORMIR	1983	Record
GABRIELA, CRAVO E CANELA		Record
JUBIABA	1983	Record
MAR MORTO	1982	Record
MAR MORTO	1983	Record
O AMOR DO SOLDADO	1983	Record
O AMOR DO SOLDADO	1986	Editora Record
O CAVALEIRO DA ESPERANCA	1983	Record
O MENINO GRAPIUNA	1912	Record
O PAIS DO CARNAVAL	1983	Record
O SUMICO DA SANTA	1988	Record
OS PASTORES DA NOITE	1983	Record
OS SUBTERRANEOS DA LIBERDADE	1983	Record
OS VELHOS MARINHEIROS	1983	Record
SAO JORGE DOS ILHEUS	1983	Record
SEARA VERMELHA	1983	Record
SUOR	1983	Record
TEREZA BATISTA CANSADA DE GUERRA	1983	Record
TERRAS DO SEM FIM	1983	Record
TIETA DO AGRESTE	1983	Record
TOCAIA GRANDE	1984	Record

28 rows selected.

FIGURA 5.2 – Títulos de Jorge Amado existentes no banco de dados

5.1.1 Ontologia Gerada Semiautomaticamente

Seguindo o processo descrito no capítulo anterior, na construção desta ontologia o usuário previamente selecionou a tabela livro e as colunas TITULO, EDITORA e ANO_PUBLICACAO, conforme modelo apresentado na Figura 5.1. Na representação textual da ontologia, a tabela foi mapeada para um conjunto de objetos não léxicos, linha 1 da ontologia na Figura 5.3, e as colunas foram mapeadas para conjuntos de objetos léxicos, linhas 3, 17 e 25 da ontologia na Figura 5.3.

```
 1: LIVRO [-> object];
 2:
 3: LIVRO [0:1] has TITULO [1:*];
 4: TITULO matches [100] case insensitive
 5: constant
 6:    { extract "\bO\s*Amor\s*do\s*soldado\b"; },
 7:    { extract "\bOs\s*Velhos\s*marinheiros\b"; },
 8:    { extract "\bO\s*Amor\s*do\s*soldado\b"; },
 9:    { extract "\bOs\s*Subterraneos\s*da\s*liberdade\b"; },
10:    { extract "\bSeara\s*vermelha\b"; },
11:    { extract "\bSao\s*Jorge\s*dos\s*ilheus\b"; },
12:    { extract "\bBahia\s*de\s*todos\s*os\s*Santos\b"; },
13:    { extract "\bTereza\s*Batista\s*cansada\s*de\s*guerra\b"; },
14:    :
15: end;
16:
17: LIVRO [0:1] has ANO [1:*];
18: ANO matches [4]
19: constant
20:    {
21:    extract "19(12|66|8(2|3|4|6|8))";
22:    };
23: end;
24:
25: LIVRO [0:1] has EDITORA [1:*];
26: EDITORA matches [50]
27: constant
28:    {
29:    extract "(Editora\s+Record|Martins|Record)";
30:    };
31: end;
32:
33: LIVRO [0:1] has PRECO [1:*];
34: PRECO matches [15] case insensitive
35:         constant
36:         {
37:                   context "R\$\s*?([0-9]{1,3},?)+";
38:                   extract "([0-9]{1,3},?)+";
39:         };
40: end;
```

FIGURA 5.3 – Representação textual da ontologia

Seguindo as regras propostas no capítulo anterior, a geração das expressões regulares nos *"data frames"* para cada coluna seguiu alguns critérios de acordo com o tipo e volume dos dados.

Para a coluna `TÍTULO`, como são apenas vinte e oito os livros existentes no banco de dados, as instâncias poderiam ser incluídas no *"data frame"*. Nesta primeira ontologia, optou-se por incluir as instâncias no *"data frame"*. Desta forma, os dados extraídos foram exatamente os indicados na ontologia. Na Figura 5.3, o *"data frame"* contendo a expressão regular para instâncias de título do livro está parcialmente representada nas linhas 4 a 15.

Para a coluna `ANO_PUBLICACAO`, as instâncias dos livros de Jorge Amado, cadastrados no banco de dados, não são muitas. Isso permite a geração de uma expressão regular que represente estes valores. O algoritmo selecionado foi o de árvores

digitais – *"Digitree"*. Da mesma forma, os dados extraídos foram exatamente os indicados pela expressão regular na ontologia. O *"data frame"* contendo a expressão regular dos valores desta coluna está representado nas linhas 18 a 23 da Figura 5.3.

Para a coluna EDITORA, assim como no caso de ano de publicação, são poucas as que publicaram os livros de Jorge Amado existentes na Biblioteca considerada. Da mesma forma, foi escolhido o processo de geração da expressão regular a partir do algoritmo de árvores digitais – *"Digitree"*. O *"data frame"* contendo a expressão regular das instâncias desta coluna está representado nas linhas 26 a 31 da ontologia na Figura 5.3.

Além da geração dos *"data frames"* a partir das colunas selecionadas do banco de dados, o usuário deve incluir um *"data frame"* com o propósito de extrair o preço dos livros. O usuário escolhe, a partir da biblioteca de expressões regulares, a mais adequada ao formato que o preço do livro aparece nas páginas. Neste caso, nas quatro páginas definidas para o experimento, o preço é precedido pelo símbolo "R$". Ao final da ontologia apresentada na Figura 5.3, linha 33, consta a sintaxe OSM representando o conjunto de objetos léxicos PRECO, assim como as linhas 34 a 40 contém o *"data frame"* da expressão regular extraída da biblioteca de expressões regulares padrão.

Até este ponto, a primeira versão da ontologia foi gerada com a intervenção do usuário através da seleção das tabelas no banco de dados, escolha dos algoritmos de geração da expressão regular a partir das instâncias e inclusão de *"data frames"* para extração de dados adicionais. O próximo passo é buscar na *Web* as páginas HTML de onde se deseja extrair os dados sobre os livros. O processo de busca consiste em navegar até a página de cada livraria virtual e fazer a seleção dos livros pelo critério definido. Neste experimento, o critério de busca é o nome do autor Jorge Amado.

A busca dos dados sobre livros foi realizada nas quatro livrarias virtuais mencionadas anteriormente, com 327 títulos existentes, muitos dos quais repetidos, com ano de publicação ou idioma diferentes. Cada livraria disponibiliza o resultado da busca de uma maneira diferente, desde a quantidade de livros por página até sua ordem. As páginas resultantes da busca nas livrarias foram gravadas como um arquivo HTML separado e identificado. Posteriormente, estes arquivos foram carregados para página de demonstração do extrator do grupo DEG (www.deg.byu.edu).

A ontologia gerada foi carregada para a página de demonstração do extrator, assim como os arquivos HTML. O processo de extração recupera os dados das páginas a partir da ontologia definida e de um dos arquivos HTML selecionados. O processo de extração foi repetido para cada arquivo HTML resultante das páginas das livrarias pesquisadas, observando-se os resultados para cada execução. A primeira livraria virtual a ser pesquisada foi a Livraria Siciliano. A busca dos livros de Jorge Amado retornou 66 títulos, como mostra a Figura 5.4.

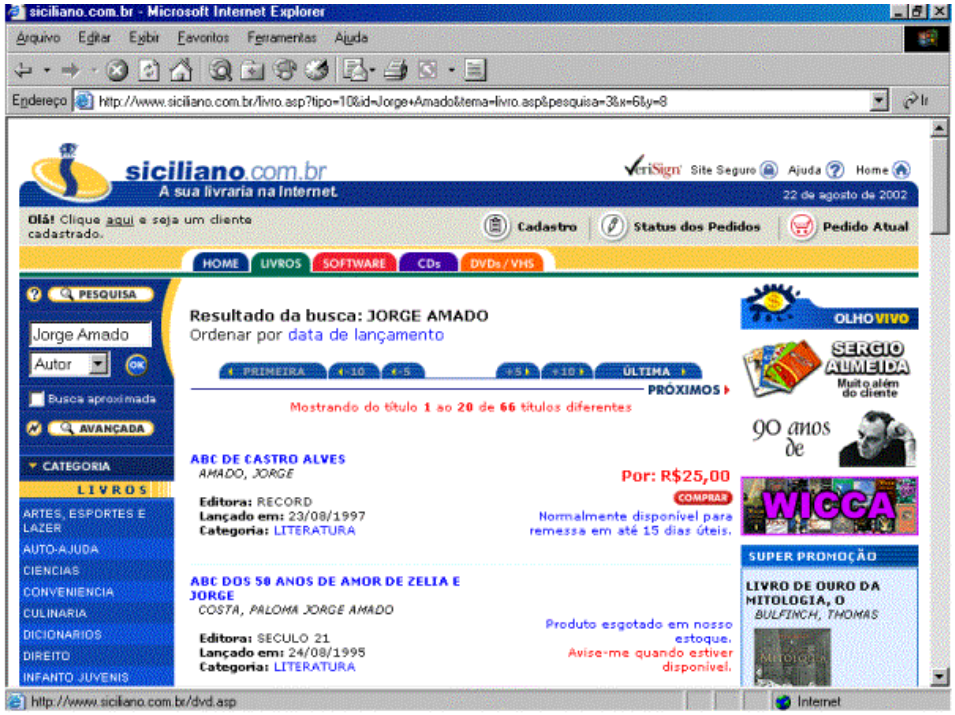

FIGURA 5.4 – Resultado da busca dos livros de Jorge Amado na Livraria Siciliano

Os 66 livros foram distribuídos em quatro páginas, com no máximo 20 títulos. Cada uma das páginas foi gravada como um arquivo HTML, e utilizado no processo de extração. Do total de títulos existentes nesta livraria, trinta títulos coincidem com os existentes no banco de dados. A verificação da quantidade de títulos coincidentes com o banco foi realizada visualmente, contando-se as ocorrências. Verificado este número, o próximo passo foi executar o processo de extração observando-se as taxas de recuperação e precisão. O resultado está demonstrado na Tabela 5.1, onde as linhas representam os conjuntos de léxicos(atributos) que deverão ser extraídos e as colunas representam os dados avaliados na extração.

TABELA 5.1 – Resultados da extração na página da Livraria Siciliano

	Número de ocorrências (N)	Número de ocorrências corretas + parcialmente (C)	Número de ocorrências extraídas incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	30	13	0	0,43	1,00
Editora	30	29	0	0,97	1,00
Ano publicação	30	1	0	0,03	1,00
Preço	30	29	0	0,97	1,00
Total	120	72	0	0,60	1,00

Nesta tabela, como nas demais que demonstram o resultado da extração de outras livrarias, as colunas têm o seguinte significado:

- O *número de ocorrências* está vinculado à quantidade de títulos dos livros de Jorge Amado existentes na página da livraria, e que também existam no banco de dados. Para as demais ocorrências (atributos) como *editora, ano de publicação e preço*, foi considerado o mesmo número de ocorrências de título do livro, pois não faz sentido sua extração sem o título;

- O *número de ocorrências correta + parcialmente* é a quantidade de léxicos que foram recuperados de forma correta ou parcialmente correta. Foram considerados parcialmente corretos os títulos extraídos nos quais faltava apenas um artigo ou uma preposição;
- O *número de ocorrências extraídas incorretamente* é a quantidade de ocorrências que foram extraídas, mas que não correspondem aos atributos associados, ou que estavam incompletas;
- A *taxa de recuperação* é calculada pela razão entre o número de ocorrências extraídas corretamente e parcialmente corretas(C) e o número de ocorrências existentes na página (N);
- A *taxa de precisão* é calculada pela razão entre o número de ocorrências extraídas corretamente e parcialmente corretas (C) e o número total de ocorrências extraídas, corretas e incorretas (C + I).

Analisando-se os resultados apresentados no processo de extração desta primeira livraria, a taxa de recuperação dos títulos de livros foi bastante baixa. O atributo *título* aparece nas páginas sem delimitadores, isto é, sem alguma palavra-chave que identifique que aquele conjunto de caracteres se refere a um título de livro.O mesmo acontece com o *nome do autor*. Para resolver esta questão, optou-se por incluir as instâncias de *título* na ontologia. Desta forma, o extrator recuperou exatamente as constantes definidas no *"data frame"* de *título do livro*.

Na abordagem de extração do grupo DEG, é muito difícil a construção de uma expressão regular para dados que não possuam delimitadores nas páginas. Por outro lado, dados como preço são mais fáceis de recuperar porque possuem delimitadores como o símbolo da moeda real (R$), presente nas páginas das quatro livrarias pesquisadas. Pode-se construir expressões regulares com um alto grau de precisão, assim como foi o resultado da extração do preço dos livros. A taxa de recuperação para o preço do livro foi de 97%, deixando de recuperar apenas uma ocorrência porque não casou com expressão regular definida.

Com o atributo *nome da editora*, embora houvesse na página um delimitador que poderia ter sido usado como palavra-chave, sua expressão regular foi gerada para identificar exatamente as editoras que publicaram os livros de Jorge Amado existentes no banco de dados. Esta expressão foi gerada através do algoritmo de árvores digitais (*"digitree"*) obtendo uma taxa de recuperação de 97%. A única ocorrência não recuperada foi porque a editora era diferente das editoras existentes no banco.

Na extração dos dados do atributo *ano de publicação*, apesar de ter sido usado uma expressão regular bastante precisa, a taxa de recuperação foi muito pequena (3%) porque foram utilizadas as instâncias existentes no banco de dados. Os anos de publicação dos livros cadastrados na base de dados da biblioteca são diferentes dos anos de publicação de cada livro na página da livraria.

A taxa de precisão para os quatro atributos foi de 100%. Isto ocorreu porque no processo de geração da ontologia optou-se por expressões regulares mais precisas. Com isso, não ocorreram extrações incorretas nem parcialmente corretas. O dado é extraído apenas se coincidir com as expressões regulares definidas na da ontologia.

A mesma ontologia foi utilizada no processo de extração das páginas das demais livrarias virtuais. A Figura 5.5 mostra a seleção dos livros na página da Livraria Cultura.

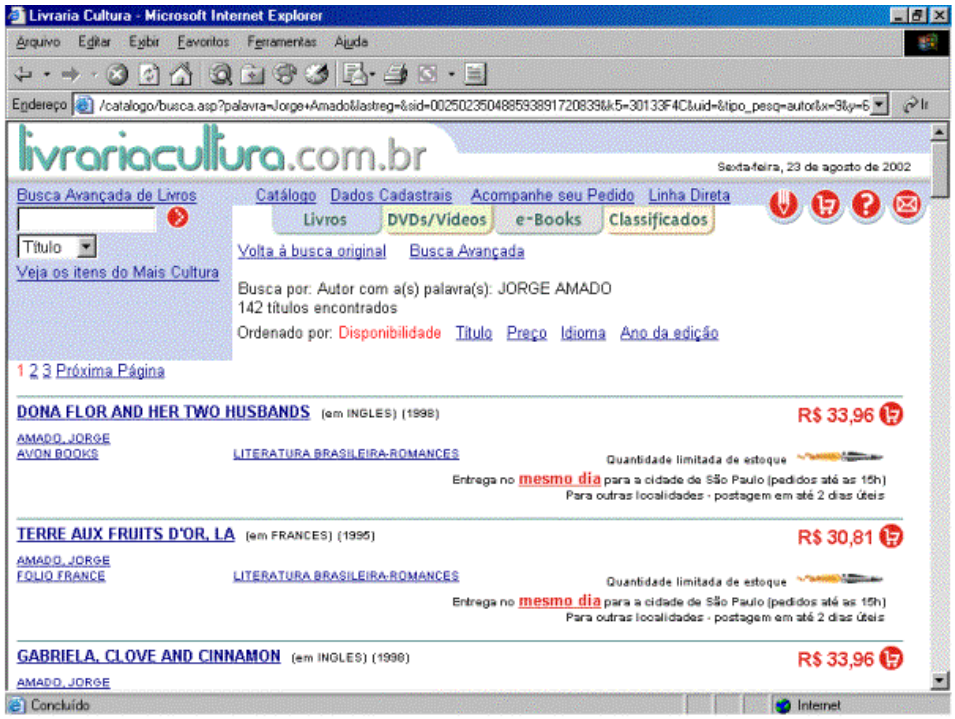

FIGURA 5.5 – Resultado da busca dos livros de Jorge Amado na Livraria Cultura

Nesta livraria, o número de títulos encontrados foi maior que na livraria anterior, 142 títulos distribuídos em três páginas com até 50 títulos. Os resultados da extração na página da Livraria Cultura estão demonstrados na Tabela 5.2.

TABELA 5.2 – Resultado da extração na página da Livraria Cultura

	No. Ocorrências na página (N)	No. Ocorrências corretas + parcialmente (C)	No. Ocorrências extraídas Incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	72	36	0	0,50	1,00
Editora	72	34	0	0,47	1,00
Ano publicação	72	6	0	0,08	1,00
Preço	72	71	1	0,99	0,99
Total	288	147	1	0,51	0,99

A contagem dos livros que coincidiram com os livros cadastrados no banco de dados resultou em 72 títulos. Dentre eles, alguns são repetidos, com editora ou ano de publicação diferentes. Esta página apresenta os livros de maneira bastante semelhante à página anterior, com exceção para o nome da editora que não vem precedido por alguma palavra-chave. Além disso, a página apresenta o idioma no qual o livro foi publicado.

A taxa de recuperação dos títulos também foi baixa, com a mesma situação que na página anterior, onde o extrator irá recuperar exatamente o que foi definido nos *"data frames"* da ontologia e a taxa de precisão continuou em 100%.

Já a taxa de recuperação para o atributo *editora* caiu bastante nesta página, pois os títulos foram publicados por livrarias diferentes das que estão cadastradas no banco de dados. A taxa de precisão também ficou em 100% porque as editoras recuperadas correspondiam às definidas na ontologia.

A taxa de recuperação do atributo *ano de publicação* manteve-se baixa pelo mesmo motivo da extração na livraria anterior, com precisão alta para os que foram extraídos. Finalmente, com o atributo *preço* houve uma única extração incorreta, pois a expressão regular não previa livros com preço na casa dos milhares. Com isso, a taxa de precisão foi de 99%.

A próxima página utilizada no processo de extração foi a da Loja Virtual Submarino, com 52 títulos recuperados pelo critério de busca estabelecido, distribuídos em duas páginas. A Figura 5.6 mostra o resultado da busca nesta Loja.

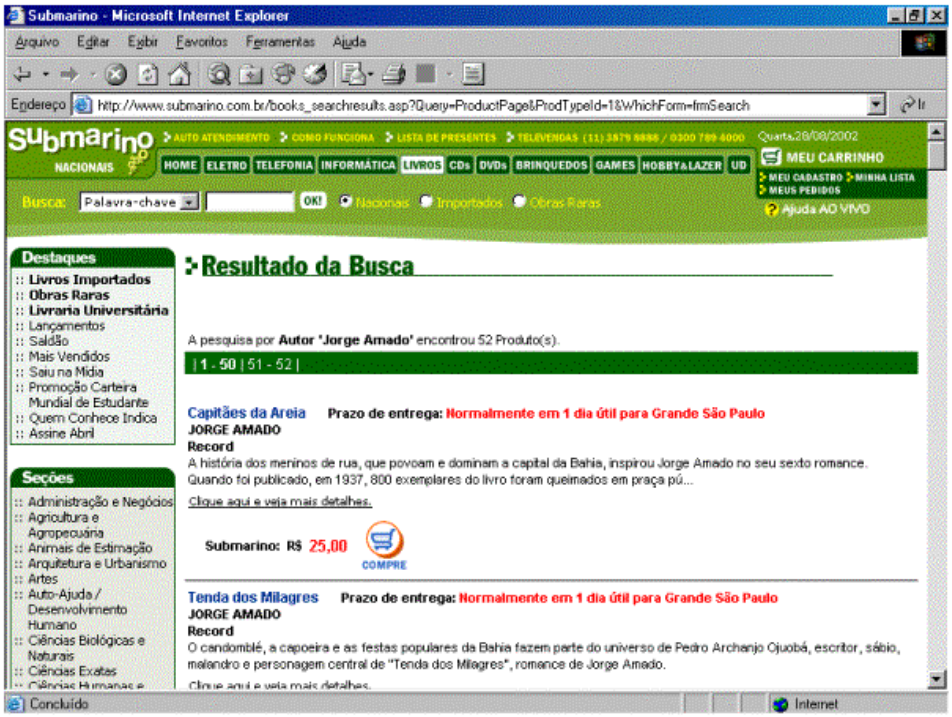

FIGURA 5.6 – Resultado da busca dos livros de Jorge Amado na Loja Submarino

Esta página é a que mais se diferenciou das demais. O ano de publicação do livro não é mostrado, existe uma pequena sinopse e alguns aparecem com preço de lista, preço promocional e o valor economizado. Mesmo com estas diferenças, a ontologia pôde ser aplicada a esta página no processo de extração. Dos 52 títulos recuperados pelo critério estabelecido, 22 coincidiram com os títulos existentes no banco de dados. A Tabela 5.3 mostra os resultados da extração nesta página.

TABELA 5.3 – Resultado da extração na página da Loja Submarino

	No. Ocorrências na página (N)	No. Ocorrências corretas + parcialmente (C)	No. Ocorrências extraídas Incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	22	18	0	0,82	1,00
Editora	22	19	0	0,86	1,00
Ano publicação	22	0	0	0,00	0,00
Preço	22	0	22	0,00	0,00
Total	88	37	22	0,42	0,63

Apesar das diferenças existentes com relação à extração do *título do livro*, esta foi a página que proporcionou a maior taxa de recuperação, pois os títulos disponíveis

coincidiram com os títulos cadastrados no banco. Para *editora*, o processo de extração se comportou de maneira semelhante ao da página da Livraria Siciliano, com uma boa taxa de recuperação. O maior problema foi com o *preço*. Como existiam três valores monetários para cada livro, o processo extraiu os três, impossibilitando a identificação de qual valor correspondia a seu preço. Na página, existem delimitadores que permitiam identificar o preço, no entanto, a ontologia foi criada independente de análise de contexto das páginas. Como resultado, a taxa de recuperação e precisão foi zero porque todos os valores extraídos foram considerados incorretos. Para o ano de publicação não houve análise dos resultados porque na página não existe este dado.

A quarta página utilizada no processo de extração foi a da Livraria Curitiba. O resultado da busca apresentou 67 títulos do autor Jorge Amado. A Figura 5.7 mostra o resultado da busca.

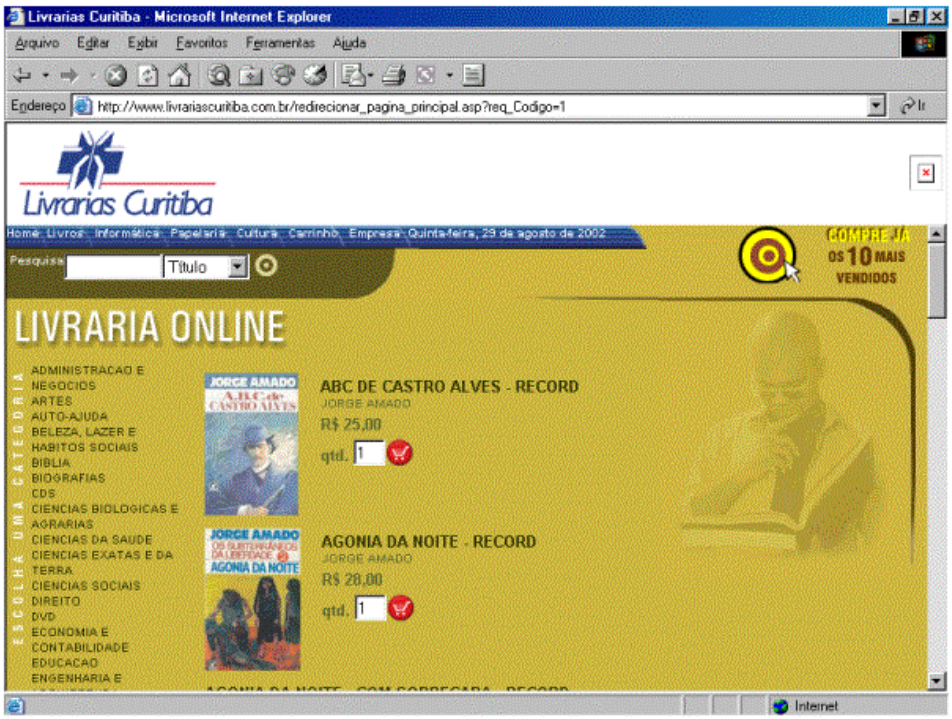

FIGURA 5.7 - Resultado da busca dos livros de Jorge Amado nas Livrarias Curitiba

Esta página contém poucas informações sobre o livro. Assim como na Loja Submarino, o ano de publicação do livro não está disponível. Dos 67 livros recuperados pelo critério estabelecido, 34 coincidiram com os títulos existentes no banco de dados. A Tabela 5.4 mostra os resultados da extração nesta página.

TABELA 5.4 - Resultado da extração na página da Livraria Curitiba

	No. Ocorrências na página (N)	No. Ocorrências corretas + parcialmente (C)	No. Ocorrências extraídas Incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	34	20	0	0,59	1,00
Editora	34	34	0	1,00	1,00
Ano publicação	34	0	0	0,00	0,00
Preço	34	34	0	1,00	1,00
Total	136	88	0	0,65	1,00

A análise dos resultados apresentados na extração desta página é o mesmo que nas páginas anteriores, com maior atenção ao atributo *ano de publicação*, que não teve nenhum valor extraído em função da página não apresentar este dado.

5.1.2 Alterações para Melhoria da Ontologia

Depois da extração utilizando a primeira ontologia gerada, o próximo passo foi verificar o que poderia ser alterado manualmente para que o processo de extração obtivesse melhores taxas de recuperação. Os atributos que contribuíram para que as taxas de recuperação fossem baixas, foram o título do livro e o ano de publicação. Com relação ao título do livro, a principal causa da não extração foram os artigos e preposições componentes do título. Na maioria das páginas o artigo inicial do título é colocado ao final, separado por uma vírgula. Por exemplo, enquanto no banco de dados existe o título "O AMOR DO SOLDADO", algumas livrarias apresentam este título como "AMOR DO SOLDADO, O".

Para melhorar a extração do título do livro, o *"data frame"* do `TITULO` foi alterado de forma que nas expressões regulares os artigos e preposições fossem considerados opcionais no momento da extração. Em Perl-5, sintaxe em que foram escritas as expressões regulares, o símbolo para tornar um ou mais caracteres opcionais é o ponto de interrogação (?). As alterações na ontologia estão destacadas na Figura 5.8.

```
 1: LIVRO [-> object];
 2:
 3: LIVRO [0:1] has TITULO [1:*];
 4: TITULO matches [100] case insensitive
 5: constant
 6:   { extract "\bO?\s*?Amor\s*(do)?\s*?soldado\b"; },
 7:   { extract "\b(Os)?\s*?Velhos\s*marinheiros\b"; },
 8:   { extract "\bO?\s*?Amor\s(do)?\s*?soldado\b"; },
 9:   :
10: end;
11:
12: LIVRO [0:1] has ANO [1:*];
13: ANO matches [4]
14:    constant
15:    {    extract "19[5-9][0-9]";    };
16:    constant
17:    {    extract "200[0-2]"; };
18: end;
19:
20: LIVRO [0:1] has EDITORA [1:*];
21: EDITORA matches [50]
22: constant
23:   {   extract "(Editora\s+Record|Martins|Record)";    };
24: end;
25:
26: LIVRO [0:1] has PRECO [1:*];
27: PRECO matches [15] case insensitive
28:         constant
29:         {
30:                   context "R\$\s*?([0-9]{1,3},?)+";
31:                   extract "([0-9]{1,3},?)+";
32:         };
33: end;
```

FIGURA 5.8 - Representação textual da ontologia alterada manualmente

As alterações no *"data frame"* para extração do título do livro estão destacadas nas linhas 6 a 8 da Figura 5.8, onde os artigos e preposições serão considerados opcionais com a inclusão do caractere de controle *"?"*. Com esta alteração espera-se extrair títulos que tenham pequenas diferenças cadastrais. A outra alteração está destacada nas linhas 14 a 17 da mesma figura. O propósito desta alteração foi melhorar a extração do ano de publicação do título, já que o ano de publicação cadastrado no banco de dado limitou sua extração. A expressão regular para extração do ano de publicação foi trocada por uma expressão regular existente na biblioteca de expressões, deixando de usar a que foi gerada automaticamente a partir das instâncias. Depois de alterada a ontologia, cada página das livrarias virtuais pesquisadas foi novamente submetida ao extrator.

Com a alteração da ontologia, a taxa de recuperação do atributo título na Livraria Siciliano subiu de 43% para 83% e a taxa de recuperação do atributo ano de publicação subiu de 3% para 100%. No caso de ano de publicação esperava-se o crescimento elevado, pois o extrator passou a recuperar os anos de publicação independente de coincidir com o dado existente no banco. Esta opção fica a critério do usuário no momento da construção da ontologia. Os resultados da extração dos dados das páginas da Livraria Siciliano, após alteração da ontologia, estão demonstrados na Tabela 5.5.

TABELA 5.5 – Resultado da extração após alteração da ontologia (Siciliano)

	No. Ocorrências nas páginas (N)	No. Ocorrências corretas + parcialmente (C)	No. Ocorrências extraídas Incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	30	25	0	0,83	1,00
Editora	30	29	0	0,97	1,00
Ano publicação	30	31	0	1,00	1,00
Preço	30	29	0	0,97	1,00
Total	120	114	0	0,95	1,00

Na Livraria Cultura, com a ontologia alterada, a taxa de recuperação do atributo título subiu de 50% para 72%, enquanto que a taxa de recuperação do atributo ano de publicação subiu de 8% para 71%. Os resultados da extração das páginas da Livraria Cultura estão demonstrados na Tabela 5.6.

TABELA 5.6 – Resultado da extração após alteração da ontologia (Cultura)

	No. Ocorrências na página (N)	No. Ocorrências corretas + parcialmente (C)	No. Ocorrências extraídas Incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	72	52	0	0,72	1,00
Editora	72	34	1	0,47	0,97
Ano publicação	72	51	0	0,71	1,00
Preço	72	71	1	0,97	0,99
Total	288	208	2	0,72	0,99

Na Loja Virtual Submarino, com a ontologia alterada, a taxa de recuperação do atributo título, o qual tinha obtido a melhor taxa de recuperação na primeira ontologia, foi ainda melhor, subindo de 82% para 91%. Como a página desta livraria não contém o ano de publicação, nenhum dado foi recuperado. No entanto, outros dados referentes ao ano foram extraídos das sinopses dos livros, gerando 5 ocorrências incorretas. Os

resultados da extração das páginas da Loja Virtual Submarino estão demonstrados na Tabela 5.7.

TABELA 5.7 – Resultado da extração após alteração da ontologia (Submarino)

	No. Ocorrências na página (N)	No. Ocorrências corretas + parcialmente (C)	No. Ocorrências extraídas Incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	22	20	0	0,91	1,00
Editora	22	19	0	0,86	1,00
Ano publicação	22	0	5	0,00	0,00
Preço	22	22	0	1,00	1,00
Total	88	61	0	0,69	1,00

Por último, a taxa de recuperação do atributo título na Livraria Curitiba, após alterações na ontologia, foi a melhor das quatro, subindo de 59% para 97%. Como não existe ano de publicação na página, nenhum dado foi recuperado. Os resultados da extração das páginas da Livraria Curitiba estão demonstrados na Tabela 5.8.

TABELA 5.8 – Resultado da extração após alteração da ontologia (Curitiba)

	No. Ocorrências na página (N)	No. Ocorrências corretas + parcialmente (C)	No. Ocorrências extraídas Incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	34	33	2	0,97	0,94
Editora	34	34	0	1,00	1,00
Ano publicação	34	0	0	0,00	0,00
Preço	34	34	0	1,00	1,00
Total	136	101	2	0,74	0,98

5.1.3 Variações na Ontologia

Neste primeiro experimento, foram inseridas as instâncias de *título do livro* na ontologia, pois não existem muitas ocorrências no banco de dados, apenas 28. Caso o número de títulos fosse muito grande, seria inviável incluí-los na ontologia. Um dos recursos utilizados pelo grupo DEG é a geração de um arquivo contendo as instâncias, denominado dicionário. Este arquivo é referenciado no *"data frame"* e o processo de extração faz a busca do dado neste arquivo. A sintaxe no *"data frame"* para este recurso é a seguinte:

```
01: LIVRO [0:1] has TITULO [1:*];
02: TITULO matches [100] case insensitive
03: constant
04: {  extract TITULO; };
05: lexicon
06: {  TITULO case insensitive;
07:    filename "titulos.dict";   };
08: end;
```

A linha 07 está referenciando o arquivo "titulos.dict" contendo as instâncias que devem ser extraídas da página Web no momento da extração. O algoritmo de pesquisa neste arquivo não é mencionado nas páginas do grupo DEG.

Além da alternativa de inclusão das instâncias em um arquivo que tem a função de dicionário, pode-se utilizar uma expressão regular mais genérica. Este recurso permite extrair um conjunto maior de dados. Um exemplo de expressão regular para o título do livro poderia ser `[A-Za-z]*\s+[A-Za-z]*`. Esta expressão regular permite a extração de um conjunto de caracteres que possuam uma combinação de letras de "A" a "Z" ou de "a" a "z", com duas palavras e separadas por no mínimo um espaço em branco (\s+). Esta é uma expressão regular gerada manualmente e baseada na análise visual das páginas.

5.2 Estudo de Caso sobre Livros por Assunto

O segundo estudo de caso é uma situação onde se deseja extrair livros de páginas da Web que não necessariamente existam no banco de dados, pertencentes a um determinado assunto. Neste caso, será necessário extrair também os nomes dos autores dos livros. O objetivo do segundo estudo é extrair títulos de livros que não somente estejam cadastrados no banco de dados

Embora se deseje extrair livros que não necessariamente estejam cadastrados, a geração da ontologia de extração será baseada nas instâncias existentes no banco, tanto para os atributos de livro quanto para o nome do autor. A principal diferença deste estudo para o primeiro é que a expressão regular será genérica, permitindo extrair dados além dos existentes no banco de dados. Por outro lado, o processo de extração pode não ser muito bom utilizando somente a ontologia gerada semiautomaticamente.

Esta situação pode ser aplicada quando se deseja incluir novos títulos em uma base de dados. A inclusão pode ser feita por título de um autor, por editora, por assunto. Os registros resultantes do processo de extração podem ser inseridos no banco segundo critérios definido pelo usuário. Este caso também poderia ser aplicado a outros domínios de problema, desde que se tenha um banco de dados com informações que permitam guiar o processo de extração de páginas da Web, e que estas informações estejam disponíveis nestas páginas.

Os dados cadastrados no banco de dados servirão como ponto de partida para geração da primeira ontologia e, a partir desta e da análise das páginas por parte do usuário, a ontologia poderá ser enriquecida permitindo melhores taxas de recuperação e precisão. Também diferente do primeiro estudo, o volume de dados analisado para cada atributo na construção da expressão regular foi maior. Todas as instâncias foram analisadas.

5.2.1 Ontologia Gerada Semiautomaticamente

Seguindo o processo descrito no capítulo 4, a ontologia foi gerada a partir da escolha de três atributos da tabela livro (título, editora e ano de publicação) e um atributo da tabela autor (nome do autor). Assim como no primeiro estudo de caso, para cada atributo foi escolhido um algoritmo de geração da expressão regular. Para o atributo *título do livro*, optou-se por gerar uma expressão regular genérica que relaciona os possíveis caracteres de cada palavra. O conjunto de caracteres fica entre colchetes e o caractere asterisco (*) indica que podem existir várias ocorrências dos caracteres nele representados. A quantidade de palavras no título do livro, após análise de todas instâncias do banco, pode chegar a onze. Isto pode ser verificado pela quantidade de conjuntos de caracteres, entre colchetes, existentes na expressão regular. O algoritmo

que gera esta expressão regular é o *"position-based"*. O *"data frame"* gerado para as instâncias do atributo *título* foi gerado com a seguinte sintaxe:

```
LIVRO [0:1] has TITULO [1];
TITULO matches [50] case
constant
{extract
  "[ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz]*\s*
   [ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz]*\s*
   [ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz]*\s*
   [ABCDEFGHIJKLMNOPQRSTUVWXYabcdefghijklmnopqrstuvwxyz]*\s*
   [ABCDEFGHIJKLMNOPQRSTUVWXZabcdefghijklmnopqrstuvwxyz]*\s*
   [ABCDEFGHIJKLMNOPQRSTUVWXZabcdefghijklmnopqrstuvwxyz]*\s*
   [ABCDEFGHIJKLMNOPQRSTVWXZabcdefghijklmnopqrstuvwxyz]*\s*
   [BCEFGHIJKLNPRSTVXabcdefghijklmnopqrstuvwxyz]*\s*
   [ABCDEGILMNPQSTUXabcdefghilmnopqrstuvwxz]*\s*
   [BDJOSacdefghilmnoprstu]*\s+[ademnors]*";
};
end;
```

Para o atributo *ano de publicação*, optou-se por uma expressão regular mais específica, uma vez que os diferentes valores cadastrados no banco de dados não são muitos. O algoritmo utilizado para geração da expressão regular foi o *"Digi-Tree"*. O *"data frame"* para as instâncias deste atributo foi gerado com a seguinte sintaxe:

```
LIVRO [0:1] has ANO [1:*];
ANO matches [50] case
constant
{extract
   "(1(8(62|7(3|6))|9(1(2|4)|2(2|3|4|8)|3(3|4|5|6|7|8|9)|
    4(0|1|2|3|4|5|6|7|8|9)|5(0|1|2|3|4|5|6|7|8|9)|
    6(0|1|2|3|4|5|6|7|8|9)|7(0|1|2|3|4|5|6|7|8|9)|
    8(0|1|2|3|4|5|6|7|8|9)|9(0|1|2|3|4|5|6|7|8|9)))|200(0|1))";
};
end;
```

Para o atributo *nome da editora*, optou-se também por uma expressão regular mais específica, uma vez que os nomes de editoras existentes estão bem definidos e também não são muitos. Encontram-se cadastradas no banco de dados 208 editoras. Utilizando também o algoritmo *"Digi-Tree"*, o *"data frame"* para as instâncias deste atributo foi gerado com a seguinte sintaxe:

```
LIVRO [0:1] has EDITORA [1:*];
EDITORA matches [20]
constant
{extract
"(A(C(ADEMICA|H(EVE|IAME))|DVANCED|GUALARGA|LIANZA|
  N(DIMA|GELOTTI|NABLUME)|QUARELA|RTE(NO(SA|VA)|SC)|
  T(ENIENSE|HENAEUM)|UR(IVERDE|ORA))|B(ER(K(ANA|ELEY)|TRAND)|
  IBLOS|LUMENAU|O(MLIVRO|OK(MAN|OKMANN|STORE))|RAS(IL|PO(RT|T))|
  USHATSKY)|C(A(BICIERI|LLIS|M(BRIDGE|PUS)|NALES|
  R(DEAL|T(GRAF|HAGO)))|E(BLAB|NPHA|P(LAB|PEV|RESG)|TESB)|
  LASSICA|O(DPOE|MPUDATA|NVIVIO|RTEZ)|RCRGS|U(LT(RI(X|Z)|
  URA|URAL)|POLO))|D(E(STAQUE|USTO)|I(ALETICA|FUSION|SCUBRA)|
  URBAN)|E(CONOMIA|D(ANEE|ELSA|I(BOLSO|C(EL|ON)|GON|MAX|
  OURO|PRO|T(AU|OR)|UPF)|UCATOR)|GERIA|L(DORADO|ETROBAS|
  LACURIA)|NSAIO|ST(ADAO|RUTURA)|XECUTIVE)|F(A(BRIS|CCAR
  |DESP)|E(NAME|PLAM)|IPECAFI|OR(ENSE|MAR)|U(N(A(DESP|RTE)|
  REI)|TURA))|G(AUCHO|IBRAN|LOBAL|R(A(DIVA|F(IPAR|MARK))|
  ECCO|IJALBO|OUND|YPHUS)|UA(NABARA|ZZELLI))|
  H(A(GAESSE|MBURG|RBRA)|E(INEMANN|RDER)|UCITEC)|
  I(B(ERICO|RA(CON|SA))|M(AGEM|P(ETUS|RES))|NFOBOOK|
  PA(NEMA|RDES)|RERICO|SPECO|TATIAIA)|J(ALOVI|URIDICA)|
  L(APPONI|E(ITURA|RFIXA)|I(DADOR|NARTH)|O(N(DRINA|GMAN)|
  YOLA)|UTECIA)|M(A(CCHI|DRAS|KRON|L(HEIROS|TESE)|
  NDARI(M|NO)|R(CURYO|TINS))|CMLXIV|ERCURYO|IDIOGRAF|
  O(DERNA|NTANHA|RAES))|N(ACIONAL|EGOCIO|O(RDICA|VATEC))|
  OBJETIVA|P(A(IDOS|LL(AS|OTTI)|PIRUS|RA(LELO|NINFO)|
  U(LUS|MAPE))|E(NINSULA|RITAS)|IONEIRA|LAQUETTE|
  O(LI(EDIT|GONO)|RTI(CO|NHO))|R(ESENCA|O(GRAFE|JETO|TEO)))|
  R(E(CORD|DHAT|NOVAR|SENHA|VISAO)|IMAGO)|S(A(MPEDRO|RAIVA)|
  C(HWARCZ|IPIONE)|E(NETE|RGASA)|I(CILIANO|N(GULAR|TESE))|
  U(CESU|LINA|MMUS))|T(EXTONOVO|OPBOOKS|R(IBUNAIS|OQUEL))|
  U(N(AFISCO|I(C(AMP|SUL)|DAS|FICADO|VERSO))|RANIA)|VEREDA|
  WORDWARE|ZRINYI)";
};
end;
```

O último atributo utilizado para construção da ontologia é resultado de um relacionamento *n:n* entre livro e autor. A geração do *"data frame"* para as instâncias do nome do autor pode ser através de um arquivo de dicionário que contém todos possíveis nomes de pessoas ou também através de uma expressão regular genérica. Para geração semiautomática da primeira ontologia, optou-se por utilizar o algoritmo *"position-based"* que analisa os possíveis caracteres por palavra no nome do autor. A expressão

regular, depois da análise de todos os nomes de autores cadastrados, foi gerada com a seguinte sintaxe:

```
LIVRO [1:*] has AUTOR [1:*];
AUTOR matches [30]
constant
 {extract
 "[ABCDEFGHIJKLMNOPQRSTUVWXYZaceimnorstx,]*\s*
  [ABCDEFGHIJKLMNOPQRSTUVWXYZabcdeghilmnorstuvz]*\s*
  [ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefgilmnoprstuz]*\s*
  [ABCDEFGHIJKLMNOPQRSTUVWXYZacdegilmnoprstuvxz]*\s*
  [ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefgilmnoprstuvz]*\s*
  [ABCDEFGHIJKLMNOPQRSTUVXZacdegilmnoqrstuvz]*\s*
  [ABCDEFGHIJLMNOPRSTUVWabcdegilnoprtuz]*\s*
  [ABCDEFGILMNOPRSTUV]*\s*[DE]*";
 };
end;
```

Cada conjunto de caracteres entre colchetes representa os caracteres que ocorrem em cada palavra do nome. Os nomes dos autores cadastrados no banco têm no máximo nove palavras, separadas por um ou mais espaços (*s**).Com este último atributo, a ontologia está montada e preparada para ser utilizada como entrada no processo de extração de dados. Vale lembrar que a denominação atributo é referente às colunas das tabelas `LIVRO` e `AUTORLIVRO`. Na abordagem OSM, cada coluna do banco é mapeada para um conjunto de objetos léxicos e as tabelas são mapeadas para conjuntos de objetos não léxicos.

Neste estudo, também foi necessário definir algum critério de busca nas páginas das livrarias virtuais para que se tenha um conjunto de livros disponível para extração. O critério de busca utilizado nas páginas foi por título do livro, utilizando *"banco de dados"* como chave de busca.

O resultado da busca nas páginas das quatro livrarias virtuais recuperou 67 títulos na Livraria Siciliano, 57 títulos na Livraria Cultura, 74 na Loja Virtual Submarino e 65 na Livraria Curitiba. Os dados foram apresentados nestas páginas no mesmo formato que no primeiro estudo de caso. A diferença agora é que os livros apresentados pertencem a diferentes autores. As várias páginas de cada livraria foram salvas e identificadas separadamente como arquivos HTML, que serviram posteriormente como entrada para o processo de extração. A Figura 5.9 mostra o resultado da busca na Livraria Siciliano sobre livros de bancos de dados. Por uma questão de espaço, as páginas das outras três livrarias não serão mostradas aqui.

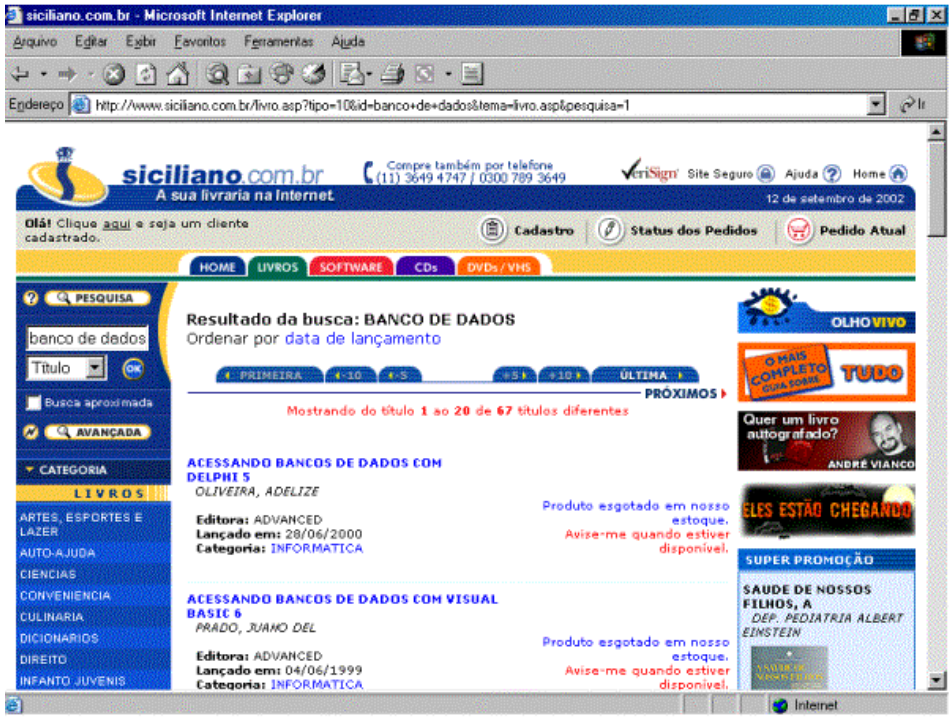

FIGURA 5.9 – Resultado da busca de títulos sobre banco de dados

Utilizando a ontologia gerada semiautomaticamente, o processo de extração foi iniciado com cada uma das páginas das livrarias pesquisadas. No entanto, o processo não recuperou nenhum dado, pois excedeu o tempo de extração da ferramenta na Internet. A causa identificada para o problema foi que as expressões regulares para o título do livro e nome do autor eram muito genéricas e o processo de extração dos dados tornava-se longo. Exceto para os *"data frames"* de título do livro e nome do autor, o processo teve sucesso. Para resolver o problema, decidiu-se por alterar manualmente as expressões regulares de título do livro e nome do autor na tentativa de incluir alguma expressão para facilitar o processo de extração.

5.2.2 Alterações para Melhoria da Ontologia

Como a chave de busca utilizada foi *"Banco de Dados"*, a expressão regular foi enriquecida para identificar esta seqüência de caracteres como contexto para o dado que seria extraído. A palavra reservada *"context"* foi utilizada no *"data frame"* do *título do livro* a fim de identificar esta seqüência de caracteres. O *"data frame"* foi então alterado para:

```
LIVRO [0:1] has TITULO [1];
TITULO matches [50] case
constant
{ context "[A-Z\s+]*BANCO\s+DE\s+DADOS\s+[A-Z\s+]*";
  extract
  "[ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz]*\s*
   :    :
};
end;
```

A apresentação dos nomes dos autores nas páginas das livrarias inicia, na maioria das vezes, pelo seu sobrenome seguido de uma vírgula e, posteriormente, o restante do nome. Esta é a forma como o nome está cadastrado no banco de dados. A expressão regular gerada semiautomaticamente, incluiu a vírgula como parte da primeira palavra do nome. Para facilitar a extração, colocou-se a vírgula como caractere fixo após a expressão regular que representa a primeira palavra. O *"data frame"* do *nome do autor* foi alterado para:

```
LIVRO [1:*] has AUTOR [1:*];
AUTOR matches [30]
constant
 {extract
 "[ABCDEFGHIJKLMNOPQRSTUVWXYZaceimnorstx]*,\s*
  :    :
 };
end;
```

Depois destas alterações, o processo de extração apresentou resultados muito significativos, pois era esperado que fossem necessárias mais alterações na ontologia. Ao contrário, apesar dos pequenos ajustes o processo de extração executou sem erros. Executado com cada uma das páginas das livrarias, os resultados estão demonstrados na Tabela 5.9, Tabela 5.10, Tabela 5.11 e Tabela 5.12, respectivamente para Livraria Sicialino, Livraria Cultura, Loja Virtual Submarino e Livrarias Curitiba.

TABELA 5.9 – Resultado da extração de livros sobre banco de dados na Livraria Siciliano

	Número de ocorrências nas páginas (N)	Número de ocorrências corretas + parcialmente (C)	Número de ocorrências extraídas incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	67	4 + 45	0	0,73	1,00
Editora	67	0	67	0,00	0,00
Ano publicação	67	61	0	0,91	1,00
Nome do autor	68	9 + 25	3	0,50	0,92
Total	269	13 + 131	70	0,54	0,67

Os resultados da extração na Livraria Siciliano foram razoáveis. A taxa de recuperação para *título do livro* e *nome do autor*, utilizando expressões regulares genéricas, foi de 73% e 50%. No entanto, os dados extraídos estavam parcialmente corretos porque em ambos o nome não ficou completo. No caso da extração do *nome da editora*, apesar do uso de uma expressão regular específica, nenhum dado foi recuperado, pois na expressão regular existe um nome de editora que coincide com uma seqüência fixa de caracteres nesta página, a qual não corresponde ao nome da editora. Para o *ano de publicação*, o resultado foi muito bom, obtendo uma taxa de recuperação próxima a 100%.

TABELA 5.10 – Resultado da extração de livros sobre banco de dados na Livraria Cultura

	Número de ocorrências nas páginas (N)	Número de ocorrências corretas + parcialmente (C)	Número de ocorrências extraídas incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	57	28 + 26	0	0,95	1,00
Editora	57	19	0	0,33	1,00
Ano publicação	57	45	0	0,79	1,00
Nome do autor	66	62 + 1	1	0,95	0,98
Total	237	154 + 27	1	0,76	0,99

Os resultados do processo de extração a partir da página da Livraria Cultura obteve os melhores resultados. A quantidade de ocorrências correta para *título do livro* superou as expectativas. Somando este resultado com o total de ocorrências parcialmente corretas, obteve-se uma taxa de recuperação de 95% e precisão de 100%, sem valores incorretos. Para o atributo *nome da editora*, apesar da expressão regular mais precisa, a taxa de recuperação ficou em 33%, conseqüência de algumas editoras serem diferentes das cadastradas no banco de dados. Para o atributo *ano de publicação*, o resultado foi semelhante ao da página da livraria anterior. Como na extração do título, a extração do nome do autor atingiu a taxa de recuperação de 95%, com destaque para o número de ocorrências extraídas corretamente.

TABELA 5.11 – Resultado da extração de livros sobre banco de dados na Loja Virtual Submarino

	Número de ocorrências nas páginas (N)	Número de ocorrências corretas + parcialmente (C)	Número de ocorrências extraídas incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	74	1 + 24	22	0,34	0,53
Editora	74	29	6	0,39	0,83
Ano publicação	74	0	1	0,00	0,00
Nome do autor	96	0	64	0,00	0,00
Total	318	54	93	0,17	0,36

Os resultados da Loja Virtual Submarino foram muito fracos. Principalmente pela disposição dos dados na página, o extrator de registros misturou os dados do *título do livro* com o *nome do autor*. Analisando o arquivo HTML, isto ocorreu porque o nome do autor estava localizado logo a seguir ao título do livro, na mesma célula da tabela. Em alguns casos o extrator recuperou alguns nomes de autores como título do livro. Para o *nome da editora*, a taxa de recuperação ficou semelhante. No caso de *ano de publicação*, a taxa de recuperação foi 0% porque nesta página não existe o ano do livro publicado. No caso do *nome do autor*, a taxa de recuperação foi 0% porque o nome do autor, na página HTML, está na ordem normal. No banco de dados, assim como nas duas primeiras páginas, o nome do autor começa com seu último sobrenome, seguido de uma vírgula e em seguida o resto do nome. Com isso, a expressão regular precisaria ser enriquecida para permitir extrair o nome também no formato normal.

TABELA 5.12 – Resultado da extração de livros sobre banco de dados na Livrarias Curitiba

	Número de ocorrências nas páginas (N)	Número de ocorrências corretas + parcialmente (C)	Número de ocorrências extraídas incorretamente (I)	Taxa de recuperação (C/N)	Taxa de precisão (C / (C + I))
Título	65	30 + 24	0	0,83	1,00
Editora	65	35	11	0,54	0,76
Ano publicação	65	0	0	0,00	0,00
Nome do autor	65	0	26	0,00	0,00
Total	260	65 + 24	37	0,34	0,71

Os resultados da extração nesta livraria são muito semelhantes ao resultado da extração na Loja Virtual Submarino. A principal diferença é que não houve recuperação incorreta para o *título do livro*. O *ano de publicação* também não estava disponível nesta página. O *nome do autor*, da mesma forma, estava disposto no formato normal, resultando em recuperações incorretas.

As alterações feitas na ontologia foram realizadas para atender às quatro páginas das livrarias, sem atenção especial para uma página específica. A ontologia ainda poderia sofrer manutenções de forma a obter uma taxa de recuperação melhor. A taxa de precisão precisaria ser melhorada em alguns casos somente. Este estudo mostrou que a ontologia pode ser gerada com expressões regulares genéricas a partir do banco de dados. No entanto, para obter melhores taxas de recuperação, é necessário incluir nas expressões regulares palavras-chave que permitam ao extrator de registros localizar o dado na página. Nesta ontologia, foi utilizada a palavra reservada *"context"*, a qual permitiu ao extrator localizar as palavras *"banco de dados"* no título do livro e extrair a seqüência de caracteres determinada pela cláusula *"extract"* no *"data frame"*.

5.3 Limitações da Ferramenta de Extração

Os estudos de caso foram realizados através de ferramentas que estão disponíveis no Web Site do grupo de extração de dados da Brigham Young University, www.deg.byu.edu. Para que o processo de extração seja executado, é necessário carregar o arquivo da ontologia, no formato texto, e também os arquivos HTML resultantes das páginas das livrarias utilizadas no estudo. As ferramentas disponíveis na página do grupo DEG foram implementadas utilizando linguagens como Perl, Java e C++. Seu código fonte não está disponível, e os experimentos deste estudo de caso foram executados somente através da Internet. A velocidade de carga da ontologia e das páginas HTML foi muito boa e processo de extração teve uma performance razoável.

A principal dificuldade para a execução dos experimentos foi com a utilização das páginas HTML. As páginas resultantes das buscas não podiam ser utilizadas diretamente no extrator. O processo de extração não consegue identificar os registros nos fontes HTML muito complexos. A complexidade está na quantidade de tabelas definidas para cada página, fazendo com que o extrator não consiga identificar algum conjunto de caracteres que separe os registros.

Na etapa de busca nas páginas das livrarias, o resultado pode ser um conjunto de livros que fisicamente não é mostrado em uma única página HTML. Em todas as livrarias, o resultado da busca ocupava de duas a cinco páginas, dependendo da

quantidade de livros. Cada uma dessas páginas teve que ser salva separadamente para ser utilizada no processo de extração.

Para resolver estes problemas, não foi possível a utilização dos arquivos HTML originais, e algumas alterações foram necessárias:

- Diminuir o tamanho da página HTML, minimizando sua complexidade. Foram retirados os comandos de Java Script e as tabelas desnecessárias da página;
- Em alguns casos, foi necessário dividir a página HTML em duas. Segundo os testes realizados, o processo de extração não funcionava com códigos HTML maiores que 50 Kbytes;
- Inserir um separador de registros necessário para o processo *"Record Separator"*. O programa de separação de registros procura por uma *"tag"* que ocorra entre cada registro da página. Um exemplo de *"tags"* para ser utilizada como separador de registros é **
<hr>
**.

Estas alterações foram feitas com o auxílio de ferramentas de edição de HTML. No primeiro estudo de caso, sobre extração de livros de um determinado autor, foram utilizadas 16 páginas HTML. No segundo estudo, foram utilizadas 17 páginas. Em ambos os estudos, as páginas tiveram que ser editadas para reduzir seu tamanho, mas sem prejuízo ou alteração de seu conteúdo.

No processo de engenharia reversa do modelo relacional para o modelo OSM é necessário que o usuário altere o nome dos conjuntos de objetos léxicos caso estes contenham o caractere sublinhado. Como apresentado na Figura 4.3, os nomes dos atributos tiveram que ser alterados pelo usuário. O extrator de registros não permite que nomes de conjuntos de objetos léxicos com tal caractere.

6 Conclusões

Esta dissertação apresenta um método semi-automático para construção de ontologias utilizadas para extração de dados a partir de documentos semi-estruturados (páginas da *Web*). Estas ontologias são especificamente construídas para uso no extrator de registros do grupo DEG, e pressupõe que exista um banco de dados relacional com mesmo domínio de problema que as páginas de onde os dados são extraídos. O método proposto permite a criação semi-automática da ontologia pela execução de duas tarefas: engenharia reversa do modelo relacional para o modelo OSM e a geração das expressões regulares a serem inseridas nos *"data frames"* da ontologia. O processo é semi-automático porque o usuário toma decisões no momento da geração da ontologia.

Um estudo de caso foi aplicado em uma organização que possui documentos legados (contratos com fornecedores). Tais documentos contêm dados que precisam ser extraídos e armazenados no banco de dados. O banco de dados existente sobre contratos foi o ponto inicial para o processo de construção da ontologia. A partir deste banco (esquema e instâncias dos dados) a primeira ontologia foi construída e aplicada no extrator de dados do grupo DEG juntamente com uma base de 50 contratos. Após algumas alterações manuais na ontologia gerada automaticamente, as taxas de recuperação e precisão ficaram entre 77% e 81% respectivamente. Os documentos utilizados estão escritos em linguagem natural sem muitas marcações ou palavras-chave para guiar o processo de extração. Este estudo não está descrito neste trabalho e foi realizado especialmente para apresentação do método no Simpósio Brasileiro de Banco de Dados [VIV2002].

Com o objetivo de melhorar o estudo de caso realizado, dois outros estudos foram realizados em documentos mais adequados à ferramenta de extração do grupo DEG, como páginas HTML que apresentem uma estrutura mínima, onde as informações estejam disponíveis em blocos de registros. Estes dois estudos estão descritos nesta dissertação.

O estudo de caso apresentado no artigo [VIV2002] e os dois estudos apresentados neste trabalho mostraram que a abordagem de extração de dados semântica, proposta pelo grupo DEG, pode ser auxiliada pela utilização do esquema e instâncias de um banco de dados relacional no momento da construção da ontologia. É necessário, entretanto, intervenção e conhecimento por parte do usuário no processo de extração. A intervenção ocorre no momento de selecionar as tabelas e colunas do banco a participarem da ontologia. O conhecimento é exigido para entender a sintaxe do modelo conceitual OSM e das expressões regulares, alterando-os se necessário. Por outro lado, diminui o tempo de construção da ontologia como um todo.

O primeiro estudo de caso, apresentado neste trabalho, foi realizado com expressões regulares mais precisas. É uma situação onde se deseja recuperar dados que coincidam com os armazenados em um banco de dados. Com a primeira ontologia gerada semiautomaticamente, o processo de extraiu os dados sem, entretanto, apresentar resultados satisfatórios. A taxa de recuperação média foi de 54,5% e a taxa de precisão foi de 90,5%. A taxa de precisão é alta porque os dados extraídos são um subconjunto dos dados que estão no banco de dados. Depois de efetuada as alterações na ontologia, a taxa de recuperação melhorou em 24,5%, subindo para 79%. A taxa de precisão manteve-se a mesma. Isto mostrou que, com pequenas alterações na ontologia, é

possível obter melhores resultados. As alterações são feitas através de análise das páginas foco da extração.

Algumas variações foram também aplicadas à ontologia deste primeiro estudo de caso. A expressão regular para o título do livro foi substituída por uma expressão mais genérica. No entanto, os resultados foram piores, pois apesar da taxa de recuperação não ter alterado, a taxa de precisão caiu muito em virtude da quantidade de ocorrências extraídas incorretamente.

O segundo estudo de caso foi realizado com expressões regulares mais genéricas. Seu objetivo era recuperar ocorrências que não necessariamente existam no banco de dados. Esta pode ser uma situação onde se deseja complementar os dados do banco de dados. O experimento realizado extraiu livros sobre o assunto banco de dados. Para isso, a ontologia gerada foi baseada nas instâncias de livros existentes no banco, com expressões regulares genéricas. A primeira ontologia gerada semiautomaticamente não teve sucesso. O programa extrator de registros da ferramenta extrapolava o tempo de execução na Internet. Verificou-se que a expressão regular estava muito genérica e alterações seriam necessárias.

Depois da primeira alteração, com inclusão das cláusulas *"keyword"* e *"context"*, o processo recuperou os dados. As taxas de recuperação e precisão foram baixas se analisado a média de todos atributos, 45,25% e 68,25% respectivamente. Isto ocorreu porque duas das quatro livrarias não disponibiliza o ano de publicação em suas páginas. Nestas mesmas duas livrarias, o nome do autor está no formato normal e, no banco de dados, está no formato invertido. Se analisadas separadamente, as médias das taxas de recuperação e precisão para título do livro e nome do autor das Livrarias Siciliano e Livraria Cultura, foram respectivamente 78,25% e 97,5%. Para expressões regulares genéricas, esperava-se baixas taxas de recuperação. Apesar disso, os resultados obtidos foram satisfatórios porque houve recuperação com expressões mais genéricas.

Para aumentar as taxas de recuperação e precisão para todos atributos, é possível melhorar ainda mais a ontologia. As expressões regulares precisariam ser enriquecidas com características específicas de cada uma das páginas das livrarias. A contribuição do método proposto neste trabalho é construir a ontologia de forma semi-automática para uso no extrator DEG. Cabe observar que, provavelmente, a construção manual das ontologias não melhoraria a recuperação obtida com a construção semi-automática.

O método proposto pode ser complementado com melhorias nos algoritmos implementados para geração das expressões regulares, criando, por exemplo, procedimentos para otimização das expressões. Os algoritmos citados neste trabalho estão implementados e são utilizados no protótipo criado para auxiliar o usuário na construção da ontologia.

Como trabalhos futuros, os algoritmos existentes na literatura sobre inferência gramatical poderiam ser explorados com maior profundidade, como outra opção de geração das expressões regulares. Apesar do protótipo ter sido implementado e testado com um banco de dados, melhorias poderiam ser feitas para que o processo considere outros bancos de dados como fonte de informação na construção da ontologia. O protótipo implementado considera apenas o banco de dados Oracle.

Anexo Protótipo da Ferramenta para Construção da Ontologia

Com o objetivo de facilitar o processo de geração da ontologia, foi implementado um protótipo que auxilia o usuário na construção das expressões regulares. Este protótipo foi implementado utilizando o banco de dados Oracle, versão 7.3.4, como base de dados e o produto Oracle Forms, versão 5.0.6, como interface para o usuário. A seguir, serão apresentadas as telas utilizadas para construção da ontologia desde a escolha das tabelas e colunas do banco de dados até a geração da ontologia final.

O processo de geração da ontologia se inicia com a seleção das tabelas e colunas do banco de dados a partir das quais a ontologia será construída. Para exemplificar o processo, será considerada a criação de uma ontologia para extração de livros sobre banco de dados. A Figura 1 representa a tela onde devem ser selecionadas as tabelas do banco de dados. Na coluna da esquerda a tabela `AUTOR_LIVRO` foi selecionada. Pressionando-se o botão "relacionamentos", as tabelas com relacionamento com `AUTOR_LIVRO` serão apresentadas na coluna da direita, com suas respectivas colunas. Neste processo, deseja-se extrair o título do livro, ano de edição, editora e autor(es) do livro. Para tal, as respectivas colunas de cada tabela estão selecionadas na coluna direita da tela. A partir das instâncias destas colunas as expressões regulares serão construídas.

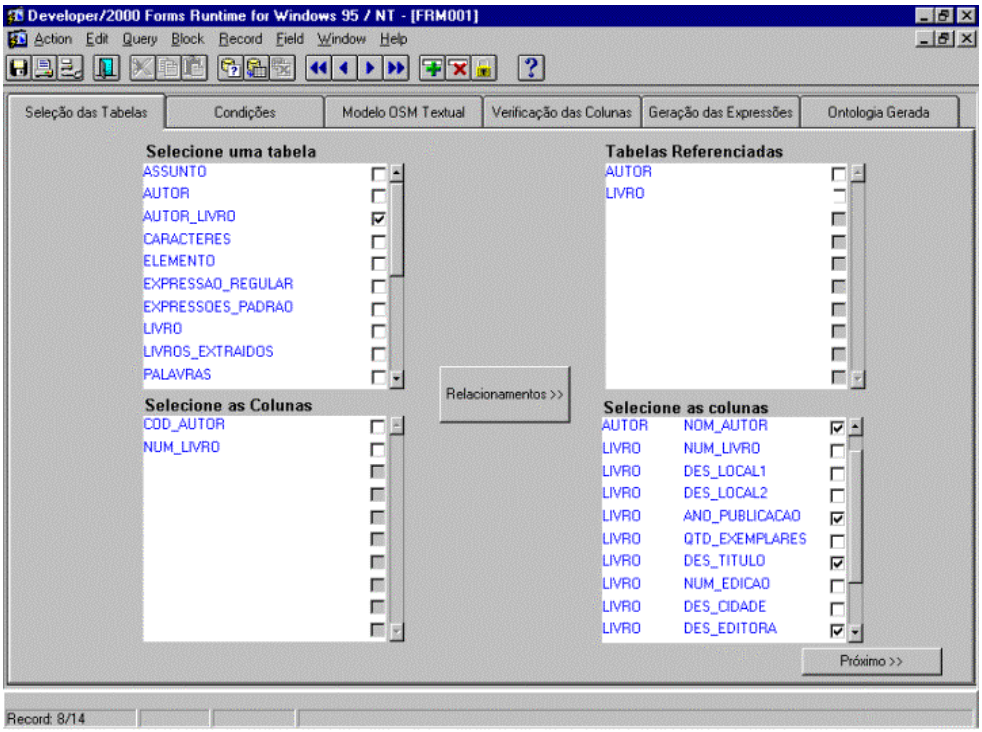

Figura 1 – Tela para seleção das tabelas e colunas do banco

Depois de selecionadas as tabelas e colunas que fornecerão os dados para construção da ontologia, o próximo passo é informar uma ou mais condições para restringir a quantidade de instâncias para geração da expressão regular. Esta informação é opcional. Se nada for informado, serão consideradas todas as instâncias de cada coluna. O comando SQL mostrado na tela mostra a consulta a ser executada para

recuperação das instâncias. A Figura 2 mostra a condição na qual o processo será restrito aos livros cujo título contenha a cadeia “BANCO DE DADOS”.

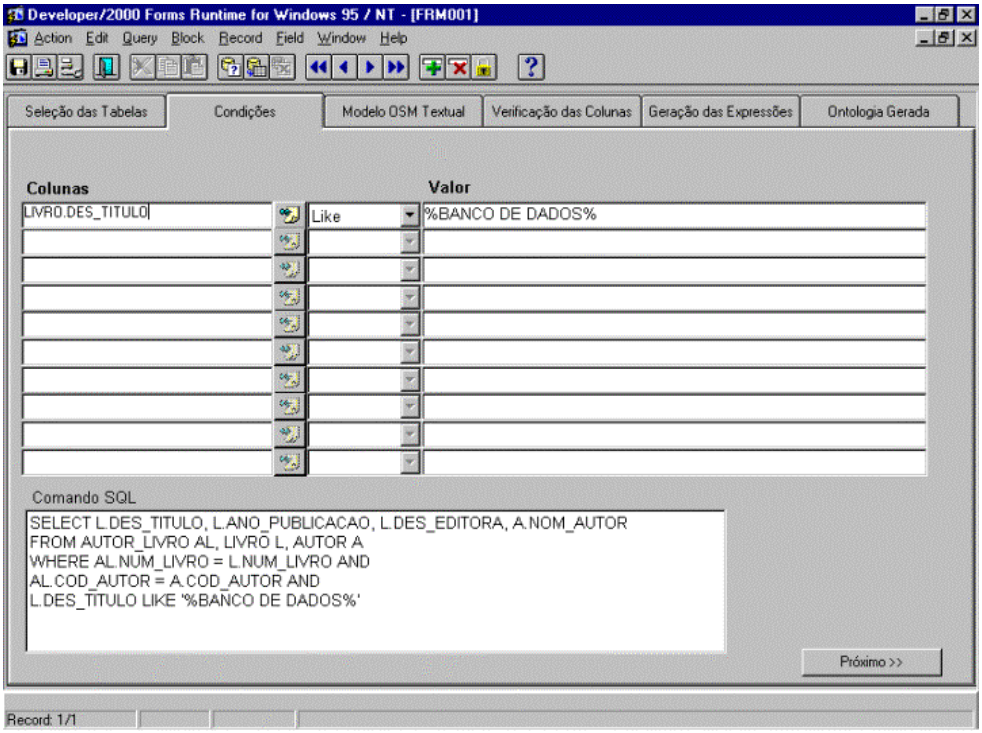

Figura 2 – Condição para selecionar as instâncias do banco de dados

De acordo com as tabelas/colunas selecionadas, o modelo textual OSM é gerado com o objeto não léxico LIVRO, incluindo os relacionamentos com os objetos léxicos (colunas selecionadas da tabela). A Figura Anexo1.3 mostra a tela com o modelo OSM gerado.

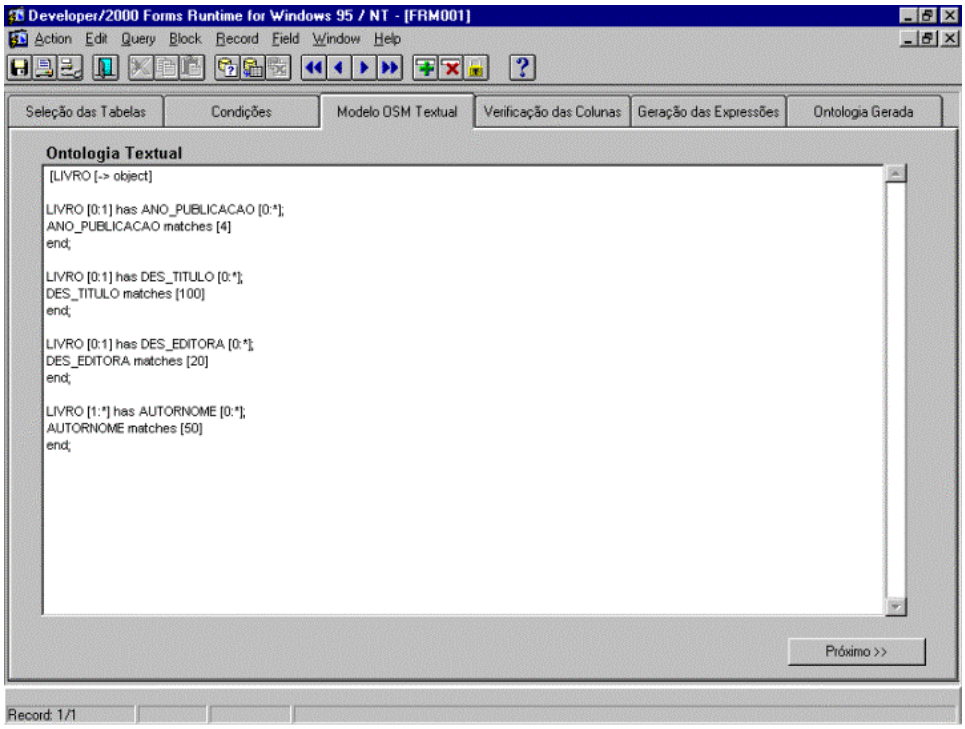

Figura 3 – Modelo OSM gerado com objetos e relacionamentos

De acordo com as regras heurísticas definidas na subseção 4.3.4 do capítulo 4, a ferramenta analisa a quantidade de instâncias existente no banco para cada coluna selecionada. O resultado desta análise está apresentado na Figura 4. À esquerda, é apresentado o total de instâncias para coluna e o número de ocorrências distintas. A partir desta contagem, à direita da tela são apresentadas as sugestões para cada coluna.

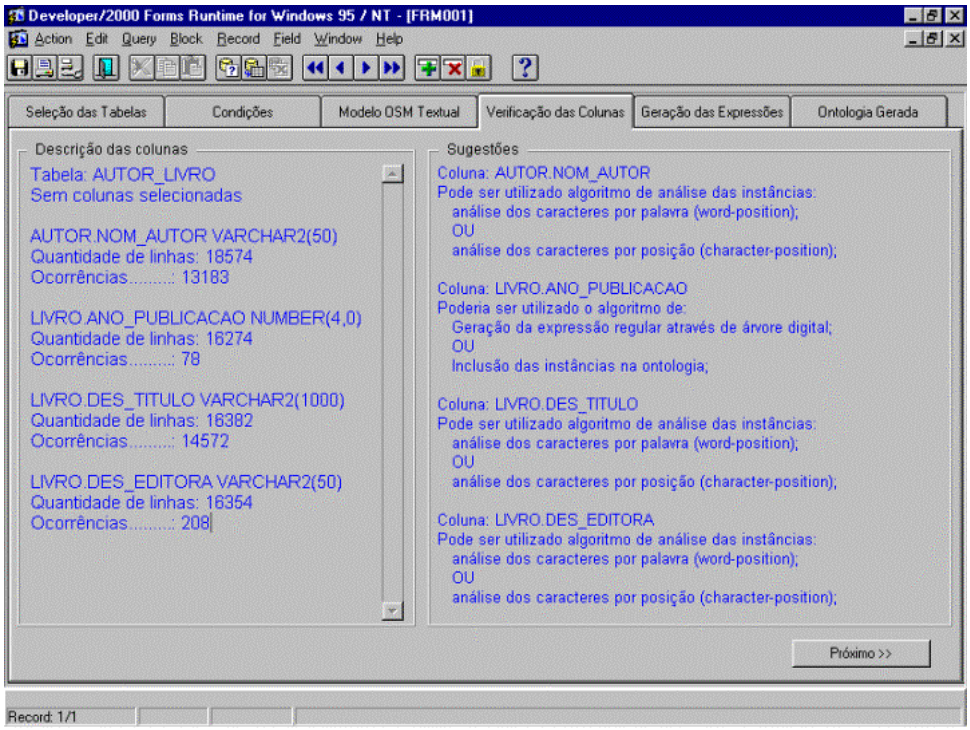

Figura 4 – Tela com resultado da análise das instâncias das colunas

O usuário decide se vai seguir ou não as sugestões feitas pelo processo de análise das colunas. A partir da próxima tela, ele pode selecionar o tipo de expressão regular que deseja utilizar para cada coluna selecionada. Para cada coluna selecionada do banco de dados, o usuário pode selecionar uma das seguintes opções de geração da expressão regular:

- Character Position;
- Word Position;
- Árvore Digital;
- Dicionário;
- Expressão Padrão.

Cada um destes tipos de geração da expressão regular está explicado no capítulo 4 nas subseções 4.3.1 e 4.3.2. A Figura 5 mostra a tela na qual são apresentadas as colunas selecionadas do banco de dados e as opções de geração da expressão regular.

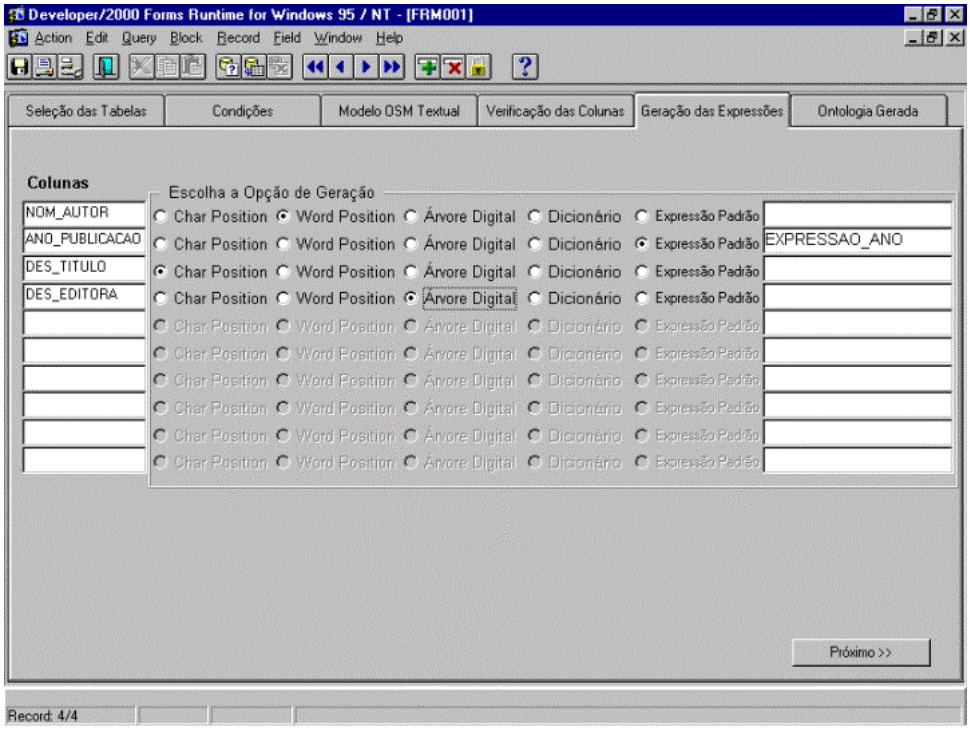

Figura 5 – Tela para escolha de geração da expressão regular

Como exemplo, a escolha de geração das expressões regulares para cada uma das colunas poderia ter seguido os seguintes critérios:

- Coluna NOM_AUTOR: O sistema sugere a geração da expressão pela análise por caractere ou por palavra. Como o número de ocorrências distintas desta coluna é muito alto, uma expressão regular genérica é recomendada para este caso. Seguindo a sugestão, a opção de geração da expressão por análise das palavras foi selecionada.
- Coluna ANO_PUBLICACAO: O sistema sugere a geração da expressão através de árvore digital ou pela inclusão do dicionário na ontologia. Como o conteúdo de ano é um valor numérico de 4 dígitos, optou-se pela utilização de uma expressão regular padrão para ano.
- Coluna DES_TITULO: Assim como para o nome do autor, o número de ocorrências distintas é alto e o sistema sugere a geração da expressão pela análise por caractere ou por palavra. Seguindo a sugestão, a opção de geração da expressão por análise dos caracteres foi selecionada.
- Coluna DES_EDITORA. O sistema sugere geração da expressão por análise de caracteres ou palavras. O número de ocorrências distintas é 208. Apesar do número ser alto, optou-se por gerar a expressão regular através da árvore digital. A expressão regular gerada é grande, e pode ser verificada na subseção 5.2.1. Apesar do tamanho, a expressão não causou erros no experimento realizado.

Definidas os tipos de expressões regulares para cada tipo de coluna, o sistema apresenta uma última tela com as expressões geradas para cada coluna, gerando a ontologia de forma semi-automática. Como a ontologia gerada é grande, não está apresentada neste anexo através de uma tela.

Bibliografia

[ABA99] ABASCAL, R.; SÁNCHEZ, J. A. X-tract: Structure Extraction from Botanical Textual Descriptions. In: STRING PROCESSING & INFORMATION RETRIEVAL SYMPOSIUM AND INTERNATIONAL WORKSHOP ON GROUPWARE, SPIRE/CRIWG, 1999. **Proceedings…** Cancún : [s.n.], 1999. p. 2-7.

[ABI2000] ABITEBOUL, Serge; BUNEMAN, Peter; SUCIU, Dan. **Gerenciando Dados na Web**. Rio de Janeiro: Campus, 2000.

[AHO74] AHO, Alfred V.; HOPCROFT, John E.; ULLMAN, Jeffrey D. **The Designs and Analysis of Computer Algorithms**. Reading, MA: Addison-Wesley, 1974.

[AHO87] AHO, Alfred V.; HOPCROFT, John E.; ULLMAN, Jeffrey D. **Data Structures and Algorithms**. Reading, MA: Addison-Wesley, 1987.

[BAT92] BATINI, C.; CERI, S.; NAVATHE, S.B**. Conceptual Database Design:** An Entity-Relationship Approach. Redwood City, The Benjamin / Cummings, 1992.

[BEN98] BENJAMINS, Richard et al. Knowledge Management through Ontologies. In: INTERNATIONAL CONFERENCE ON PRACTICAL ASPECTS OF KNOWLEDGE MANAGEMENT, PAKM, 2., 1998. **Proceedings…** Basel : [s.n.], 1998. p. 5.1 – 5.12.

[CHR2001] CHRISTIANSEN, Tom. **Perl.com – The Central Web Site for the Perl Community**. Disponível em: <http://www.perl.com>. Acesso em 01 dez. 2001.

[CRE2001] CRESCENZI, V.; MECCA, G.; MERIALDO, P. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATABASE SYSTEMS, VLDB, 26., 2001. **Proceedings…** Rome : [s.n.], 2001. p. 109-118.

[COW96] COWIE, J; LEHNERT, W. Information Extraction. **Communications of the ACM**, New York, v.39, n.1, p.80-91, Jan. 1996.

[DEN2001] DENIS, François. Learning Regular Languages from Simple Positive Examples. **Machine Learning**, Dordrecht, v. 44, n. 1/2, p.37-66, July 2001.

[DOR2000] DORNELES, Carina F. **Extração de Dados Semi-Estruturados com Base em uma Ontologia**. 2000. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

[EMB98] EMBLEY, David W. et al. Ontology-Based Extraction and Structuring of Information from Data-Rich Unstructured Documents. In: INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, CIKM, 7., 1998, Bethesda. **Proceedings…** New York : ACM, 1998. p. 52-59.

[EMB98a] EMBLEY, David W. **Object Database Development:** Concepts & Principles. Reading, MA: Addison-Wesley, 1998.

[EMB99] EMBLEY, David W. et al. Conceptual Model-Based Data Extraction of Information from Multi-Record Web Documents. **Data & Knowledge Engineering**, [S.l.], v.31, n.3, p.227-251, Nov. 1999.

[FAR96] FARQUHAR, Adam et al. The Ontolingua Server: a Tool for Collaborative Ontology Construction. In: KNOWLEDGE ACQUISITION WORKSHOP, KAW, 10., 1996. **Proceedings…** Banff : [s.n.], 1996. p. 44.1 – 44.19.

[FIK96] FIKES, Richard. **Ontologies:** What are they, and where´s the research? 1996. Disponível em: <http://www-ksl.stanford.edu/KR96/FikesPositionStatement.html>. Acesso em: 10 set. 2001.

[FLO98] FLORESCU, Daniela; LEVY, Alon; MENDELZON, Alberto. Database Techniques for the World Wide Web: A Survey. **SIGMOD Record**, New York, v.27, n.3, p. 59-74, Sept. 1998.

[GON78] GONZALES, Rafael C.; THOMANSON, Michael G. **Syntactic Pattern Recognition – An Introduction**. Reading, MA: Addison-Wesley, 1978.

[GRI97] GRISHMAN, Ralph. Information Extraction. Techniques and Challenges. In: SUMMER SCHOOL ON INFORMATION EXTRACTION, SCIE, 1997, Rome. **Proceedings…** New York : Springer-Verlag, 1997. p. 1026 – 1044.

[GRU93] GRUBBER, Tom R. A translation approach to portable ontology specifications. **Knowledge Acquisitions**, [S.l.], v.5, n.2, p. 199-220, Apr. 1993.

[GUA97] GUARINO, Nicola. **Understanding, Building, and Using Ontologies**. A comentary to "Using Explicit Ontologies in KBS Development", by Heijst, Schreiber, and Wielinga. 1997. Disponível em: <http://citeseer.nj.nec.com/guarino97understanding.html>. Acesso em: 11 jul. 2001.

[HAM97] HAMMER, Joachim; MCHUGH, Jason; GARCIA-MOLINA, Hector. Semistrutured Data: The TSIMMIS Experience. In: EAST EUROPEAN SYMPHOSIUM ON ADVANCES IN DATABASES AND INFORMATION SYSTEMS, ADBIS, 1., 1997, St. Petersburg. **Proceedings…** [S.l. : s.n.], 1997. p. 1-8.

[HOP79] HOPCROFT, John E.; ULLMAN, Jeffrey D. **Introduction to Automata Theory, Languages and Computation**. Reading, MA : Addison-Wesley, 1979.

[HUC98] HUCK, G. et al. JEDI: Extracting and Synthesizinng Information from the Web. In: CONFERENCE ON COOPERATIVE INFORMATION SYSTEM, CoopIS, 3., 1998, New York. **Proceedings…** New York : IEEE-CS, 1998. p. 32 – 43.

[KNU73] KNUTH, Donald E. **The Art of Computer Programming:** Sorting and Searching. 2nd Ed. Reading, Massashusetts: Addison-Wesley, 1973.

[LAE2002] LAENDER, Alberto; SILVA, Altigran; RIBEIRO-NETO, Berthier A.; TEIXEIRA, Juliana. A Brief Survey of Web Data Extraction Tools. **SIGMOD Record**, New York, v. 31, n. 2, June 2002.

[LAE2002a] LAENDER, Alberto; SILVA, Elaine; SILVA, Altigran. DEByE – Uma Ferramenta para Extração de Dados Semi-Estruturados. **Data Knowledge Engineering Journal**, [S.l.], v. 40, n. 2, p. 121-154, Feb. 2002.

[LIA99] LIAO, Minghong et al. Ontologies for Knowledge Retrieval in Organizations Memories. In: LEARNING SOFTWARE ORGANIZATIONS, LSO WORKSHOP, 1999, Kaiserslauten. **Proceedings…** [S.l. : s.n.], 1999. p. 19 – 26.

[LID95] LIDDLE, S; EMBLEY, D.; WOODFIELD, S. Unifying Modeling and programming through an active, object-oriented, model-equivalent programming language. In: INTERNATIONAL CONFERENCE ON OBJECT-ORIENTED AND ENTITY-RELATIONSHIP MODELING, OOER, 14., 1995, Berlin. **Proceedings…** [S.l. : s.n.], 1995. p. 55-64.

[MAE2002] MAEDCHE, A; STAAB, S. Ontologies: Representation, Engineering, Learning and Application. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, ECAI, 15., 2002, Lyon. **Tutorial.** [S.l. : s.n.], 2002.

[MEL2000] MELLO, Ronaldo S. **Aplicação de Ontologias a Bancos de dados Semi-Estruturados**. 2000. Exame de Qualificação (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

[MOU2002] MOURA, Ana Maria de Carvalho. The Semantic Web: Fundamentals, Technologies, Trends. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, SBBD, 17., 2002, Gramado. **Tutorial.** Porto Alegre : Instituto de Informática, 2002.

[NAD93] NADLER, Morton; SMITH, Eric. **Pattern Recognition Engineering**. New York: John Wiley & Sons, 1993.

[NAR2001] NARDON, Fabiane B. **Uso de XML na estruturação de informações dos históricos de pacientes**. Informação verbal obtida através visita ao Instituto do Coração – INCOR, São Paulo, jun. 2001. E-mail do responsável pelo projeto: Julio@incor.usp.br. Fabiane Nardon na época era consultora do Datasus.

[SAH99] SAHUGUET, A.; AZAVANT, F. Building lightweight wrappers for legacy Web data-sources using W4F. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES, VLDB, 25., 1999, Edinburgh. **Proceedings…** [S.l. : s.n.], 1999.

[SAL83] SALTON, Gerard; MCGILL, M. J. **Introduction to Modern Information Retrieval**. New York: McGraw Hill, 1983.

[SCH92] SCHALKOFF, Robert. **Pattern recognition:** statistical, structural, and neural approaches. New York: John Wiley & Sons, 1992.

[SNO2001] SNOUSSI, Hicham; Magnin, Laurent; Nie, Hian-Yun. Heterogeneous Web Data Extraction using Ontology. In: INTERNATIONAL BI-CONFERENCE WORKSHOP ON AGENT-ORIENTED INFORMATION SYSTEMS, AOIS, 3., 2001, Montreal. **Proceedings…** [S.l. : s.n.], 2001. p. 99-110.

[SOW2000] SOWA, John F. **Knowledge Representation:** Logical, Philosophical, and Computational Foundations. Pacific Grove, CA: Brooks Cole Publishing, 2000. Disponível em: <http://www.jfsowa.com/ontology/index.htm>. Acesso em: 20 out. 2001.

[TEI2000] TEIXEIRA, Juliana S.; LAENDER, Alberto; SILVA Altigran. **Análise Comparativa de Diferentes Abordagens para Extração de Dados Semi-Etruturados.** Belo Horizonte: UFMG, 2000. Disponível em: <http://www.dcc.ufmg.br/pos/html/spg2000/anais/juliana/juliana.htm>. Acesso em: 15 abr. 2002.

[TEN95] TENENBAUM, Aaron M.; LANGSAM, Yedidyag; AUGENSTEIN, Moshe J. **Estrutura de Dados usando C**. São Paulo: Makron Books, 1995.

[URM96] URMAN, Scott. **Oracle Pl/sql Programming**. Osborne: Oracle Press; Berkley, CA: McGraw-Hill, 1996.

[VIV2002] VIVAN, Orlando M.; HEUSER, Carlos A. Semiautomatic Generation of Data-Extraction Ontologies from Relational Databases. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, SBBD, 17., 2002, Gramado. **Anais…** Porto Alegre: Instituto de Informática da UFRGS, 2002.