

Comparação de algoritmos para cálculo de Independência Paralela entre regras de Transformações de Grafos.

Introdução

Gramáticas de grafos[3] são modelos visuais para descrição e modelagem de sistemas. Numa gramática de grafos, os estados de um sistema são modelados através de grafos, enquanto as ações que alteram os estados são modeladas como regras de reescrita de grafos. Tais regras são chamadas de *regras de transformação*. A utilização desses modelos para a aplicação de métodos de verificação formal permite aliar uma apresentação visual e intuitiva com uma semântica de execução precisa. O uso de gramática de grafos pode ser empregado em diversas áreas, tais como modelagem de sistemas concorrentes e distribuídos, entre outros. Uma das análises mais importantes em gramáticas de grafos é a análise de par crítico, pois possibilita uma visão global das interações entre regras, geralmente através de tabelas de conflitos e de dependências. Como parte fundamental desta análise, encontra-se o teste de Independência Paralela, que consiste em verificar se duas regras podem ser aplicadas ao mesmo tempo num determinado estado do sistema, obtendo o mesmo resultado que qualquer outra aplicação sequencial de ambas.

Desenvolvimento

No âmbito do projeto *Verites*[1], está sendo desenvolvido um sistema para edição, execução e verificação de modelos utilizando gramáticas de grafos, denominado *Verigraph*[2]. Atualmente o sistema possui funcionalidades como análise de par crítico, cálculo de regra concorrente, análises em gramáticas de segunda ordem e geração de casos de testes a partir de uma gramática. Esse sistema visa integrar todas as funcionalidades necessárias para análises em gramática de grafos, o que é uma carência das ferramentas disponíveis (especializadas em somente um tipo de análise). O sistema já atingiu alguma maturidade, e busca-se um refinamento sobre a sua performance e arquitetura interna. Neste sentido, foram feitas diversas comparações entre diferentes algoritmos para cálculo de Independência Paralela das regras de transformações, onde meus esforços se deram na parte de implementação e avaliação de desempenho dos algoritmos. Durante o desenvolvimento deste trabalho possíveis melhorias foram identificadas, algumas delas já foram implementadas no sistema.

Gramática de Grafos, Teoria das Categorias e Implementação no Verigraph

A abordagem de Gramáticas de Grafos usada para guiar o desenvolvimento do Sistema *Verigraph*[2] é definida usando conceitos de Teoria das Categorias. Esta abordagem é especialmente importante pois permite certa abstração no momento de realizar provas, resultando em conceitos que podem ser usados em diferentes estruturas matemáticas.

Dentre as operações de *Teoria das Categorias* que são importantes para o estudo de Independência Paralela, estão os seguintes:

- Pullback (PB) - Complexidade Quadrática
- Final Pullback Complement (FBPC) - Complexidade Quadrática
- Pushout Complement (POC) - Complexidade Quadrática
- Verificação de Isomorfismos (ISO) - Complexidade Linear
- Verificação de Fatorização (FAT) - Complexidade Linear

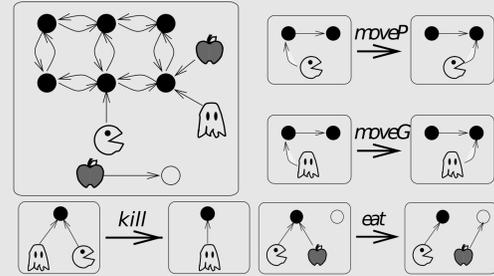
Como o sistema *Verigraph*[2] implementa todas essas operações citadas os algoritmos de Independência Paralela foram classificados de acordo com a quantidade de operações que usam:

	Alg 1	Alg 2	Alg 3	Alg 4	Alg 5
PB	1	2	2	0	1
FBPC	0	0	0	0	1
POC	0	0	0	1	0
ISO	0	1	1	0	1
FAT	1	0	0	1	0

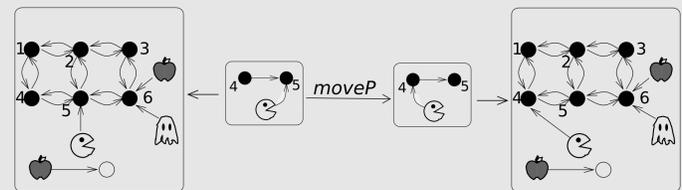
Até o desenvolvimento deste trabalho, a operação de verificação não era implementada no sistema *Verigraph* de maneira otimizada, sua implementação era baseada em duas outras operações básicas: Busca de Morfismos (Complexidade Exponencial) e Filtragem por Composições (Complexidade Linear). Durante o desenvolvimento deste trabalho, desenvolveu-se uma otimização para esta operação específica, assim conseguiu-se uma implementação de complexidade linear. É importante destacar ainda, que os algoritmos **1** e **4** são menos genéricos que os algoritmos **2**, **3** e **5**, porém os casos onde se aplicam são bastante comuns.

Estudo de Caso

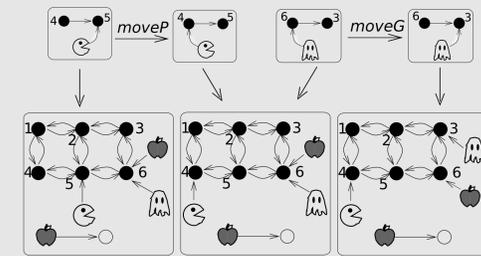
Para ilustrar o funcionamento de uma modelagem com Gramática de Grafos, abaixo é apresentado um exemplo modelando o jogo *Pacman*:



Para simular o comportamento do sistema, as regras são aplicadas sucessivamente sobre o grafo inicial, até que nenhuma das regras seja aplicável. Abaixo está um exemplo da aplicação da *moveP* no Grafo Inicial:



Para ilustrar o conceito de Independência Paralela, abaixo está um exemplo de aplicação sequencial de duas regras que possuem Independência Paralela entre si, nota-se que a aplicação das regras na ordem inversa teria o mesmo grafo resultante:



Abaixo, a tabela de Independência Paralela entre as regras da gramática, Verdadeiro (V) significa que as duas regras possuem Independência Paralela entre si:

	MoveP	MoveG	Eat	Kill
MoveP	F	V	F	F
MoveG	V	F	V	F
Eat	F	V	F	F
Kill	F	F	F	F

Resultados

Os 5 algoritmos a serem testados foram aplicados a todas as combinações de pares entre as regras de uma gramática de grafos que modela um sistema de elevadores. Foram medidos os tempos de execução de cada algoritmo por par de regras da gramática:

Alg 1	Alg 2	Alg 3	Alg 4	Alg 5
99	114	146	211	265

Tempo médio por par (microsegundos)

A partir destes resultados, verificamos que o Algoritmo 1 possui o melhor desempenho entre os 5, porém, como ele possui algumas restrições, o Algoritmo 2 é uma boa opção nos casos em que o Algoritmo 1 não pode ser aplicado. Na execução deste trabalho, percebeu-se que uma das operações mais custosas dentre as utilizadas nesse algoritmos é a operação de Pullback, que será foco de otimizações em trabalhos futuros. Os resultados detalhados deste trabalho estão disponíveis em [2].

Referências

- [1] Website do grupo verites. https://wiki.inf.ufrgs.br/Grupo_de_Verifica%C3%A7%C3%A3o,_Valida%C3%A7%C3%A3o_e_Teste_de_Sistemas_Computacionais. Acessado em: 12-09-2017.
- [2] Jonas Santos Bezerra, Andrei Costa, Guilherme Azzi, Leonardo Marques Rodrigues, Rodrigo Machado, and Leila Ribeiro. *Verites/verigraph: Parallel Independence Benchmarks*, Jun 2017.
- [3] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation (Monographs in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.