

## Introdução

*Gramáticas de grafos*[3] são modelos visuais com os quais pode-se descrever sistemas, onde os estados são modelados através de grafos e o comportamento como regras de reescrita de grafos, 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. Dentre as análises que podem ser feitas sobre gramáticas de grafos encontra-se a análise de par crítico. Essa é uma análise estática que visa determinar todas as possíveis interações entre pares de regras, identificando situações como conflitos ou dependências. O uso de gramática de grafos pode ser empregado no estudo de diversas áreas, tais como modelagem de sistemas concorrentes e distribuídos, design de banco de dados, construção de compiladores e muitos outros.

## Desenvolvimento

O projeto *Verites*[2], está desenvolvendo uma nova ferramenta para edição, execução e verificação de modelos utilizando gramáticas de grafos, denominada *Verigraph*[1]. Essa ferramenta 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). Atualmente a ferramenta *Verigraph* já possui funcionalidades importantes implementadas, como análise de conflitos e dependências entre regras e cálculo de regra concorrente a partir de uma derivação. Estão sendo implementadas atualmente funcionalidades adicionais como Verificação de Modelos (utilizando a lógica *CTL*) e suporte a gramáticas de grafos de 2º ordem.

Durante o desenvolvimento deste trabalho minhas contribuições para a ferramenta se concentraram na implementação da verificação de *matches* (aplicabilidade das regras), na análise e comparação de desempenho com ferramentas já existentes e também no auxílio à implementação dos módulos de Análise de Conflitos e Dependências.

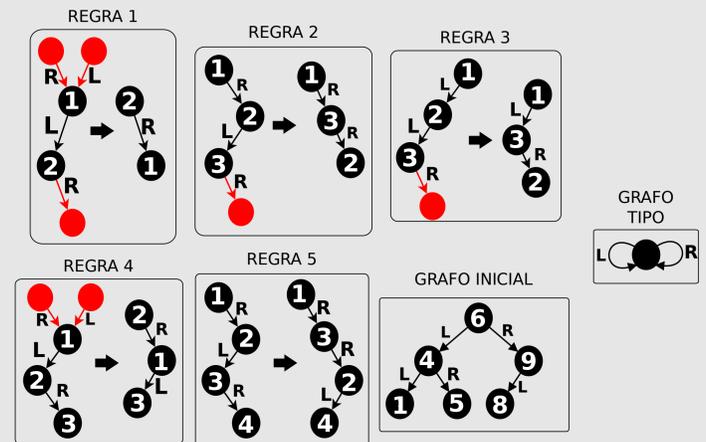
## Gramática de Grafos

Uma *Gramática de Grafos Tipados*[3] é uma tupla  $\mathcal{G} = (T, G_0, P, \pi)$ , onde  $T$  é um grafo (Grafo Tipo),  $G_0$  é um grafo tipado em  $T$  (Grafo Inicial),  $P$  é um conjunto de nomes de regras de transformação e  $\pi$  é uma função que associa cada nome pertencente a  $P$  a uma regra de transformação.

Uma regra de transformação[3] é uma tupla  $\mathcal{P} = (N, L \xleftarrow{l} K \xrightarrow{r} R)$ , sendo  $l : K \rightarrow L$  e  $r : K \rightarrow R$  homomorfismos de grafos injetores e  $N = \{n_i : L \rightarrow X_i\}$  uma coleção de morfismos com origem em  $L$ . O grafo tipado  $L$  representa o padrão de aplicação de uma transformação,  $R$  representa o padrão resultante de uma transformação e  $N$  representa um conjunto restrições na aplicação da transformação.

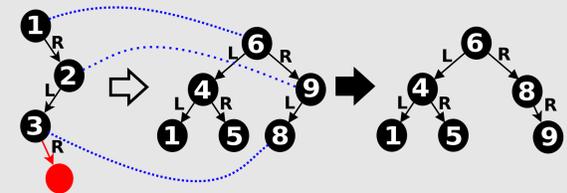
## Estudo de Caso

Para ilustrar o funcionamento de uma modelagem com Gramática de Grafos, abaixo é listado um exemplo modelando a transformação de uma árvore binária qualquer em uma lista.

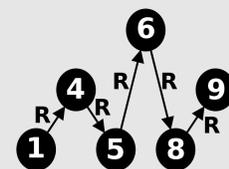


\*Símbolos vermelhos representam restrições de aplicações

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 Regra 2 no Grafo Inicial:



A execução da gramática a seguir, aplicando-se a sequência de regras formada pelas regras 2, 4, 1 e 2 ao grafo inicial resulta no seguinte grafo final:



Dentre as possíveis análises que podem ser feitas sobre gramáticas de grafos encontra-se a análise de par crítico, que visa determinar as interações entre pares de regras, identificando situações de conflitos ou de dependências. Essa análise é útil pois permite uma visão geral sobre as interações entre todas as regras, apontando principalmente onde a execução concorrente das regras é inviável. Abaixo, estão os resultados das Análises de Conflitos e Dependências realizadas na gramática apresentada:

	R1	R2	R3	R4	R5
R1	3	1	2	1	0
R2	5	10	9	7	5
R3	5	7	12	7	4
R4	4	5	5	10	5
R5	5	13	12	17	42

Resultado da Análise de Conflitos

	R1	R2	R3	R4	R5
R1	1	1	0	2	2
R2	3	5	2	7	12
R3	2	3	3	7	8
R4	4	5	4	7	8
R5	5	18	13	18	37

Resultado da Análise de Dependências

## Referências

- [1] Repositório público do verigraph. <https://github.com/Verites/verigraph>. Acessado em: 29-07-2016.
- [2] Website do grupo verites. [https://wiki.inf.ufrgs.br/Grupo\\_de\\_Verifica%C3%A7%C3%A3o,\\_Valida%C3%A7%C3%A3o\\_e\\_Testes\\_de\\_Sistemas\\_Computacionais](https://wiki.inf.ufrgs.br/Grupo_de_Verifica%C3%A7%C3%A3o,_Valida%C3%A7%C3%A3o_e_Testes_de_Sistemas_Computacionais). Acessado em: 01-08-2016.
- [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.