

# Desenvolvimento da ferramenta VeriGraph: algoritmos de "matching" e reconhecimento de condições negativas de aplicação

HERDT, R. G<sup>1</sup>

## Introdução

O projeto Verites dedica-se ao estudo de técnicas e ao desenvolvimento de ferramentas para verificação, validação e teste de sistemas computacionais. Uma de suas linhas de pesquisa envolve gramáticas de grafos, as quais consistem de modelos computacionais onde um grafo descreve o estado do sistema e um conjunto de regras de reescrita define seu comportamento. Gramáticas de grafos têm diversas aplicações em sistemas computacionais, sendo úteis, por exemplo, na modelagem de sua evolução estrutural. Para tanto, foi desenvolvido o modelo de gramática de grafos de segunda ordem (GGSO), onde uma gramática de grafos é o estado de sistema e um conjunto de regras de segunda ordem definem suas transformações (MACHADO, 2011).

## Discussão

**VeriGraph:** Existem diversas ferramentas para visualização, escrita e análise de gramáticas de grafos, cada qual com recursos voltados para abordagens distintas do problema. Algumas delas oferecem recursos de análise estática do sistema, outras restringem-se à análise dinâmica. Neste contexto, iniciou-se o desenvolvimento do VeriGraph, uma ferramenta que permitirá o emprego de ambas técnicas de análise, além de oferecer - através da natureza genérica das estruturas e algoritmos escolhidos e da flexibilidade da linguagem utilizada no desenvolvimento (Haskell) - um ambiente de fácil prototipação de novos modelos teóricos (como GGSO's).

**Implementação:** O projeto do VeriGraph almeja o desenvolvimento de uma ferramenta genérica e flexível, facilmente adaptável a novas necessidades e modelos teóricos. Para tanto, optou-se pela utilização da linguagem de programação funcional Haskell, a qual oferece diversos recursos para abstração de problemas e otimização de desempenho. Além disso, a organização do software busca a criação de estruturas de dados que aproximem-se ao máximo possível do referencial teórico, simplificando a evolução do programa com o amparo em conceitos já estabelecidos.

**Algoritmos de Matching e NAC's:** Parte importante do desenvolvimento do VeriGraph envolve a escolha e implementação de algoritmos de "matching" de grafos, neste caso considerando-se o problema de homomorfismo de subgrafos, provado ser NP-Completo. A estratégia adotada foi a modificação de um algoritmo de CSP (Constraint Satisfaction Problem) existente (RUDOLF, 1998), de modo a tratar características específicas do problema. A flexibilidade desta abordagem permitiu, por exemplo, que a busca eliminasse "matches" geradores de conflitos de reescrita, reduzindo a complexidade das etapas seguintes do programa. Outro recurso implementado nesta primeira etapa do projeto é o reconhecimento de NAC's (condições negativas de aplicação), necessárias para a definição de regras de reescrita mais complexas.

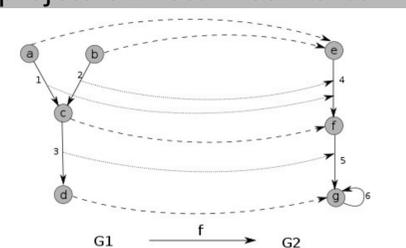


Figura 1: Morfismo de Grafos (MACHADO, 2011)

## Resultados Parciais

No momento, a ferramenta é capaz de realizar a reescrita e gerar o espaço de estados de sucessivas aplicações. Além disso, os algoritmos implementados permitiram a obtenção de um desempenho satisfatório, uma vez que o algoritmo baseado em CSP permite a restrição do domínio do problema a cada iteração. Os próximos passos envolvem: ler e gravar arquivos em formato padrão (GXL); reescrita em segunda ordem; análise de par crítico; desenvolvimento da interface gráfica.

## Considerações

O desenvolvimento por completo de uma nova ferramenta propiciou ao grupo a oportunidade de adquirir experiência e acumular conhecimento sobre diversos problemas relacionados direta e indiretamente a sistemas de reescrita de grafos. Espera-se poder, a partir de sua conclusão, implementar o modelo de gramática de grafos de segunda ordem e explorar as possibilidades oferecidas para a análise da evolução de sistemas computacionais.

## REFERÊNCIAS

RUDOLF, Michael. **Utilizing constraint satisfaction techniques for efficient graph pattern matching.** In *6th International Workshop on Theory and Application of Graph Transformations (TAGT)*, page 238-251. Springer, 1998.

MACHADO, Rodrigo. **Higher-Order Graph Rewriting Systems.** Porto Alegre: PPGC da UFRGS, 2011.

<sup>1</sup> Graduando em Ciência da Computação UFRGS; Bolsista de IC (PIBIC CNPq).