

GRAMÁTICA DE GRAFOS: UM COMPARATIVO DE MÉTODOS DE ANÁLISE

**LEMOS JUNIOR, Luiz Carlos¹; MELLO, André Moura de¹;
CAVALHEIRO, Simone Andre da Costa¹, FOSS, Luciana¹**

¹Universidade Federal de Pelotas – {lclemos, ammello, simone.costa, lfoss}@inf.ufpel.edu.br

1. INTRODUÇÃO

A cada dia nos deparamos com sistemas mais complexos e com *hardwares* cada vez mais sofisticados. A tarefa de especificar um *software* ou um *hardware* não é algo natural. Diversas técnicas e metodologias são utilizadas para fazer essas especificações. O fato das técnicas difundidas no mercado permitirem gerar resultados diferentes do que foi especificado fez com que surgisse a necessidade de utilizar métodos formais para especificação e verificação de sistemas. Neste caso, o sistema de software ou de hardware é especificado formalmente através de um modelo matemático, isto é, através de uma linguagem formal que possui sintaxe e semântica bem definidas. E a verificação formal permite garantir propriedades deste sistema descrito através da linguagem formal.

Gramática de grafos (EHRIG et al., 1997) é uma linguagem formal bastante adequada para sistemas cujos estados possuem uma topologia complexa e cujo comportamento é essencialmente orientado a dados, isto é, eventos são disparados por configurações particulares do estado. De forma geral, um sistema descrito por gramática de grafos é composto por um grafo tipo, caracterizando os tipos de vértices e arcos permitidos no sistema; um grafo inicial, representando o estado inicial do sistema; e um conjunto de regras, que descreve as possíveis mudanças de estados que podem ocorrer.

A análise de sistemas especificados em gramática de grafos pode ser realizada através da técnica de verificação de modelos (CLARKE et al., 1999) e de prova de teoremas (ROBINSON et al., 2001). A técnica de verificação de modelos tem como entrada um modelo finito representando o sistema e uma propriedade a ser verificada neste modelo, e então uma busca exaustiva no espaço de estados permite decidir se a propriedade é válida no modelo dado.

Já na prova de teoremas, tanto o sistema quanto suas propriedades são expressas em alguma lógica. O processo de prova consiste em encontrar uma prova a partir dos axiomas e lemas intermediários do sistema. Este processo é semiautomático, isto é, alguns passos da prova são realizados automaticamente, porém requerem intervenção humana para ser completada.

Cada técnica tem argumentos pró e contra o seu uso, mas é possível dizer que a verificação de modelos e a prova de teoremas são complementares.

Este trabalho objetiva apresentar as principais abordagens existentes para a análise de gramática de grafos dentro de cada uma das técnicas, verificação de modelos e prova de teoremas, traçando um comparativo entre elas.

2. MATERIAL E MÉTODOS

As principais abordagens propostas para a análise de gramática de grafos, utilizando verificador de modelos ou prova de teoremas foram estudadas.

O verificador de modelos GROOVE (GRaphs for Object-Oriented Verification) (RENSINK, 2004) é uma ferramenta que permite a especificação

direta do sistema em gramática de grafos e a partir dela, gera o espaço de estados do modelo. Instanciando uma propriedade em lógica temporal (LTL ou CTL), o sistema de transição resultante é verificado de forma automática.

FOSS (2004) propõe uma tradução de uma classe específica de gramática de grafos, Gramática de Grafos Baseada em Objetos (GGBO) para π -calculus (MILNER, 1999). GGBOs incorporam conceitos como comunicação por troca de mensagens e encapsulamento, que permitem a descrição de sistemas baseados em objetos. A partir deste método, verificadores automáticos, por exemplo HAL (FERRARI et al., 1998) e Mobility Workbench (VICTOR et al., 1994) podem ser usados para a verificação automática.

Outra proposta (DOTTI et al., 2003) traduz especificações GGBO para PROMELA (*PROcess/PROtocol MEta LAnguage*), linguagem de entrada do verificador SPIN. O SPIN (HOLZMANN, 1997) é uma ferramenta bastante eficiente (em termos de tempo de processamento) e permite que propriedades sejam especificadas em lógica temporal. Contraexemplos são fornecidos quando as propriedades são violadas.

Outra abordagem (MICHELON et al., 2006) propõe uma extensão de gramática de grafos para modelar explicitamente restrições de tempo, permitindo a verificação automática de propriedades através do verificador UPPAAL. Nesta proposta, a semântica de sistemas de tempo real especificados em gramática de grafos é definida em termos de autômatos temporizados (ALUR et al., 1994), linguagem de entrada do UPPAAL.

BALDAN et. al. (2008) propõem aproximar o comportamento de sistemas de transformação de grafos (GTSs) usando uma cadeia finita de sub-aproximações ou de super-aproximações, num nível k de precisão do *unfolding* completo do sistema. AUGUR é uma ferramenta que permite a análise de GTSs usando as super-aproximações.

Até o momento, apenas uma abordagem (DA COSTA et al., 2010) foi proposta para permitir a análise de gramática de grafos através de prova de teoremas. O trabalho propõe uma abordagem lógica e relacional para gramática de grafos, provendo uma codificação de grafos e regras com relações. A abordagem relacional foi definida para o caso de grafos tipados simples e estendido para grafos com atributos e gramáticas com condições de aplicação negativas. A proposta para grafos simples foi traduzida para estruturas event-B (DEPLOY, 2011), o que permitiu o uso dos provadores disponíveis para esta linguagem, através da plataforma Rodin (ABRIAL et al., 2010).

3. RESULTADOS E DISCUSSÕES

Os métodos de análise para modelos descritos em gramática de grafos apresentados neste artigo apresentam prós e contras que devem ser levados em conta quando se deseja escolher um deles. A Tabela 1 sumariza as principais características que diferenciam as abordagens previamente descritas. Nesta tabela, são destacados os métodos de verificação disponíveis para cada abordagem: verificador de modelos (VM) ou prova de teoremas (PT); os tipos de sistemas com os quais cada abordagem pode lidar: sistemas de estados finitos (EF) ou sistemas de estados infinitos (EI); os tipos usados de gramáticas de grafos: GG tipadas (Tipos), GG com atributos (Atrib.), GG com condições de aplicação negativas (NACs) e GG temporizadas (Tempo); as linguagens usadas para especificar as propriedades a serem verificadas: lógicas CTL (*Computation Tree Logic*), TCTL (*Timed Computation Tree Logic*), ACTL (*Action Computational*

Tree Logic), LTL (*Linear Temporal Logic*), além de lógica de primeira ordem (Lóg. 1ª Ord.) e teoria dos conjuntos (Teo. Conj.); os tipos de propriedades verificadas: sobre as computações (Comp) ou sobre as estruturas dos estados (Est); as abordagens que realizam verificação diretamente sobre a gramática de grafos (Sim) e aquelas que usam algum tipo de tradução para outra linguagem (Não); e as abordagens que possuem ferramentas disponíveis para especificação (Espec), verificação (Verif) e simulação (Sim).

Tabela 1. Comparativo entre métodos de verificação para gramática de grafos

	GROOVE	AUGUR	SPIN	Mobility Workbenck	UPPAAL	Rodin
Verificação	VM	VM	VM	VM	VM	PT
Tipos de Sistema	EF	EI (aprox.)	EF	EF	EF	EI
Tipos de Gramáticas	Tipos NACs	Atrib.	Tipos Atrib.	Tipos	Tipos Atrib. Tempo	Tipos Atrib. NACs
Linguagens de especificação de propriedades	CTL, LTL	LTL, ACTL	LTL	μ -Calculus	TCLT	Lóg. 1ª Ord. Teo. Conj.
Tipos de propr.	Comp	Comp	Comp Est (limit.)	Comp	Comp	Est
Análise direta sobre GGs	Sim	Não	Não	Não	Não	Não
Suporte de ferramentas	Espec Verif Sim	Espec Verif Sim	Espec Verif	Verif	Verif	Verif

Ficou evidenciado que verificação de modelos e prova de teoremas são complementares, por exemplo, nos tipos de propriedades.

4. CONCLUSÕES

Neste artigo apresentamos um comparativo entre os métodos, verificação de modelos e prova de teoremas, para sistemas descritos usando gramática de grafos. Neste comparativo, foram analisadas as principais abordagens existentes para GGs bem como as ferramentas que dão suporte a elas. Nesta análise, ficou claro que uma abordagem que possa usufruir dos pontos fortes de cada um dos métodos pode ser bastante interessante. Como trabalhos futuros, pretendemos estudar a possibilidade de integrar de alguma forma verificação de modelos e prova de teoremas em um framework para especificação e verificação formal de gramática de grafos.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- ABRIAL, J. R.; BUTLER, M.; et al. **International Journal on Software Tools for Technology Transfer**, Springer, 2010.
- ALUR, R.; DILL, D. L. **Theoretical Computer Science**, v. 126, n. 2, p. 183 – 235, 1994.
- BALDAN, P.; CORRADINI, A.; KÖNING, B. **Information and Computation**, v. 206, p.869 – 907, 2008.
- CLARKE, E.; GRUMBERG, O.; PELED, D. **Model Checking**. Cambridge: MIT Press, 1999.
- DA COSTA, S. A.; RIBEIRO, L. **Science of Computer Programming**, In Press, Corrected Proof, 2010.
- DEPLOY. **Event-b and the Rodin Platform**. Rodin Development is Supported by European Unicon ICT Projects DEPLOY (2008 to 2012) and RODIN (2004 to 2007), Acessado em junho 2011. Online. Disponível em: <http://www.event-b.org/>
- DOTTI, F. L.; FOSS, L.; RIBEIRO, L.; SANTOS, O. M. **In 6th International Conference on Formal Methods for Open Object-Based Distributed Systems**, Springer, v. 2884, p. 261 – 275, 2003.
- EHRIG, H.; HECKEL, R.; KORFF, M.;LOWE, M.;RIBEIRO, L.; WAGNER, A.; CORRADINI, A. **Handbook of graph grammars and computing by graph transformation**, River Edge, NJ, USA, v. I, p. 247 – 312, 1997.
- FERRARI, G.; GNESI, S.; MONTANARI, U.; PISTORE, M.; RISTORI, G. **CAV '98: Proceedings of the 10th International Conference on Computer Aided Verification**, London, Springer-Verlag, p. 511 – 515, 1998.
- FOSS, L.; RIBEIRO, L. **Electronic Notes in Theoretical Computer Science**, v. 95, p. 245 – 267, 2004.
- HOLZMANN, G. J. **IEEE Transactions on Software Engineering**, Los Alamitos, v. 23, n. 5, p. 279 – 295, 1997.
- MICHELON, L.; COSTA, S. A.; RIBEIRO, L. **Brazilian Symposium on Software Engineering**, p. 97 – 112, 2006.
- MILNER, R. **Communicating and Mobile Systems: The π -Calculus**. New York: Cambridge University Press, 1999.
- RENSINK, A. **Applications of Graph Transformations with Industrial Relevance AGTIVE**, Springer-Verlag, v. 3062, p. 479 – 485, 2004.
- ROBINSON, J. A.; VORONKOV, A. **Handbook of Automated Reasoning**. Elsevier and MIT Press, 2001.
- VICTOR, B.; MOLLER, F. **Proceedings of the 6th International Conference on Computer Aided Verification**, London, Springer-Verlag, p. 428 – 440, 1994.