UMA TRADUÇÃO DE GRAMÁTICA DE GRAFOS CONTROLADA PARA O EVENT-B

ALEX BERTEI[1]; SIMONE ANDRÉ DA COSTA CAVALHEIRO[2]; LUCIANA FOSS[3]

[1]*Universidade Federal de Pelotas – abertei@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas* – simone.costa@inf.ufpel.edu.br
[3]*Universidade Federal de Pelotas* – lfoss@inf.ufpel.edu.br

1. INTRODUÇÃO

O desenvolvimento de sistemas complexos têm sido uma tarefa árdua, tornando necessária uma especificação de forma completa e sem erros. Diante disso, as técnicas para auxiliar o desenvolvimento de sistemas confiáveis e corretos estão se tornando mais necessárias (DWYER et al., 2007). Uma das maneiras de alcançar este objetivo é através do uso de métodos formais, que são técnicas baseadas em formalismos matemáticos que podem oferecer formas rigorosas e eficazes para modelar, projetar e analisar sistemas de computador (CRAIGEN;GERHART; RALSTON, 1993).

Gramática de grafos (GG) (EHRIG et al., 1997) é uma linguagem visual que permite especificar e verificar formalmente sistemas com características complexas como paralelismo, concorrência e distribuição. Além disso, existem diferentes técnicas e ferramentas que permitem o uso de model checking (RENSINK, 2008), e provadores de teoremas (CAVALHEIRO, 2010), para verificação de propriedades de sistemas descritos nesta linguagem. Em uma gramática de grafos, os estados do sistema são representados através de grafos e o comportamento ou as mudanças de estado são definidas por regras de transformação de grafos. Nas abordagens usuais de gramática de grafos, a ordem de aplicação das regras não é definida por uma estrutura de controle e sim pelos recursos (dados) disponíveis no estado corrente.

Gramáticas de grafos controladas (GGC), por outro lado, permitem definir uma ordem na aplicação das regras, sem levar em conta o estado, e sim uma estrutura de controle auxiliar, como por exemplo, expressões regulares (ER) e máquinas de estados. Porém, para este tipo de gramática não existem ferramentas que permitam a verificação formal de propriedades.

Em CAVALHEIRO (2010) foi proposta uma abordagem relacional para gramática de grafos que permite o uso dos provadores de teoremas da ferramenta Rodin. A ferramenta faz uma verificação estática (sintática) e uma verificação dinâmica, usando a linguagem Event-B (DEPLOY, 2014). Assim, a proposta deste trabalho é integrar gramática de grafos controladas na técnica de prova de teoremas para GGs, para permitir a analise de propriedades de sistemas especificados por GG com estruturas de controle para a aplicação das regras.

2. METODOLOGIA

Para atingir o objetivo deste trabalho, iniciou-se com o estudo da linguagem Event-B para se entender como as restrições impostas pelas ERs poderiam ser traduzidas para essa linguagem. Porém chegou-se a conclusão de que não é possível, uma vez que o Event-B não dispõe de nenhum tipo de estrutura de controle para a execução de seus eventos. Um evento está habilitado sempre que suas guardas são satisfeitas. Assim, as estruturas de controle impostas pelas ERs são transferidas para os dados. Para isso foi definida uma tradução de GGCs

para GGs contendo novos vértices para restringir a ordem de aplicação das regras através de conflitos e dependências entre estas.

Gramáticas de grafos controladas são definidas por um par $GGC = (G, exp)$, onde G é uma gramática de grafos e exp um conjunto de expressões regulares sobre o conjunto de regras da gramática G. Para uma ER E, sobre um conjunto de nomes de regras, o conjunto de regras iniciais de E contém todos os prefixos de comprimento 1 das palavras denotadas por E. De forma análoga, o conjunto de regras finais de E contém todos os sufixos de comprimento 1 das palavras denotadas por E.

A tradução de GGCs para GGs é feita através da adição de dependências e conflitos nas regras da gramática. Estas dependências e conflitos são necessários para manter a ordem de aplicação das regras que são definidas pela expressão regular. Duas regras estão em conflito se a primeira exclui algum item (vértice/aresta) que seja necessário para a aplicação da segunda, isso implicará que se a primeira acontece a segunda não poderá acontecer. Dependência é quando uma regra cria algum item que é necessário (preservado ou deletado) para outra regra ser aplicada, então a segunda só poderá acontecer depois que a primeira acontecer. Para introduzir as dependências e conflitos, são alterados os lados direito e esquerdo das regras. Para adicionar tais dependências, as expressões regulares são decompostas em sub-expressões e para cada uma delas são realizadas alterações nas regras sempre que necessário, visto que nem todas as operações das ERs requerem a adição de dependências e conflitos. Somente as concatenações requerem essas alterações que são realizadas como segue. Para cada regra final da primeira sub-expressão, é adicionado um vértice novo do lado direito, já para cada regra inicial da segunda sub-expressão, é adicionado este mesmo vértice do lado esquerdo. Após a criação das dependências entre as regras, é feita uma nova alteração para adicionar um conflito entre as regras iniciais da ER. As alterações são feitas da seguinte maneira: adiciona-se um vértice novo no grafo inicial e, para todas as regras iniciais da ER, é alterado o seu lado esquerdo, adicionando este mesmo vértice. Esse novo vértice adicionado vai criar um conflito entre essas regras, fazendo com que só uma delas possa ser aplicada no grafo inicial. O conjunto de regras é sempre incrementado com as novas instâncias das regras, então deste conjunto é retirada somente as instâncias das regras que de fato estão sendo usadas na ER, ou seja, as regras modificadas.

A adição de um vértice novo sempre acontece para manter o conflito e a dependência desejada, caso um mesmo vértice seja adicionado a outra regra, este vai criar conflito e dependências indesejadas. A ordem da aplicação das regras vai ser dada pela dependência entre elas, onde a aplicação de uma habilita a aplicação da seguinte. Através destas condições, o conflito e as dependências nas regras conseguem representar o mesmo comportamento das ER.

3. RESULTADOS E DISCUSSÃO

Realizou-se um estudo de caso sobre gramática de grafos controlada (GGC), onde foi possível fazer a transformação de uma gramática de grafos controlada para uma gramática de grafo (GG), a qual possui a mesma semântica sequencial da GGCs.

Essa tradução seguiu os passos descritos na metodologia, onde, para cada regra foram adicionados vértices novos. Como consequência tem-se a adição de conflitos e dependências entre as regras e, com isso, o comportamento foi mantido o mesmo da ER.

A Figura 1 abaixo representa o conjunto de regras da gramática de grafos controlada G.

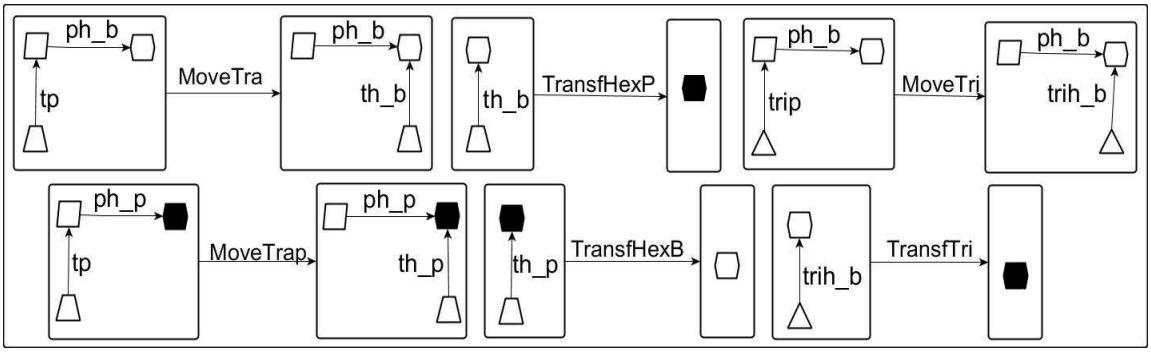

Figura 1: Regras da Gramática de grafos G

A expressão regular que define o comportamento desta GGC é a ER e = *((MoveTri + MoveTrap) + ((MoveTra + MoveTrap); MoveTri; (TransfHexP + TransfTri + TransfHexB)))*. A aplicação das regras acontece da seguinte maneira: primeiro se aplica uma escolha não determinística entre as sub-expressões *(MoveTri + MoveTrap)* ou *((MoveTra + MoveTrap); MoveTri; (TransfHexP + TransfTri + TransfHexB))*, caso seja escolhida a primeira sub-expressão, então será feita uma nova escolha não determinística entre as regras *MoveTri* e *MoveTrap*. Caso a sub-expressão escolhida seja *((MoveTra + MoveTrap); MoveTri; (TransfHexP + TransfTri + TransfHexB))* então a primeira regra a ser aplicada será escolhida não deterministicamente entre as regras *MoveTra + MoveTrap*, em seguida a regra aplicada será a regra *MoveTri*, e após será feita outra escolha não determinística para aplicar uma das regras *(TransfHexP + TransfTri + TransfHexB)*.

A tradução da GGC apresentada na Figura 1, usando a ER e é feita através da adição de vértices novos para criar as dependências e conflitos entre as regras, para manter a ordem de aplicação definida pela ER. A Figura 2 mostra as regras da GG' resultante, com suas devidas dependências e conflitos.

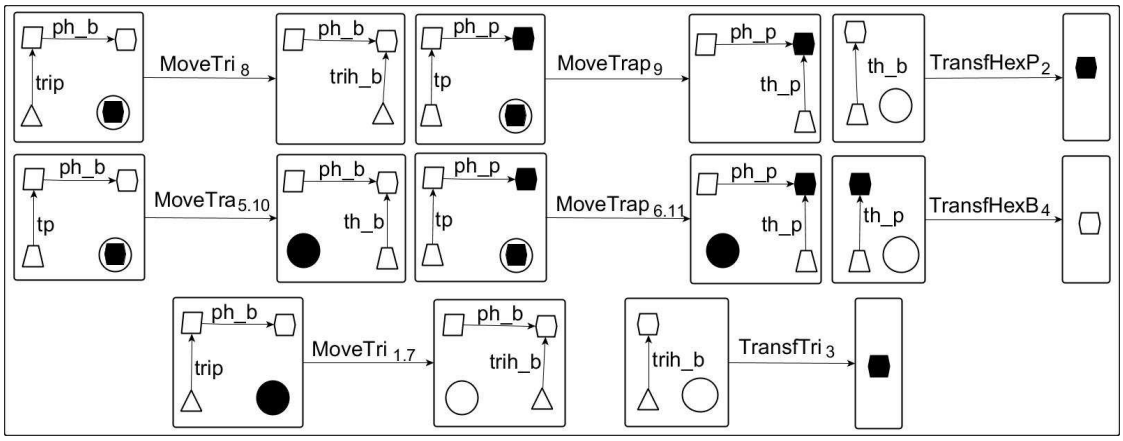

Figura 2: Regras da Gramática de grafos GG'

A ordem de aplicação imposta pela ER se mantem através dos novos vértices, por exemplo: será feito uma escolha não determinística entre as regras *($MoveTri_8$ + $MoveTrap_9$)*, caso uma dessas seja aplicada, a aplicação das regras acaba, pois nenhuma delas habilita a aplicação de outra regra. Caso haja uma escolha não determinística entre as regras $MoveTra_{5.10}$ + $MoveTrap_{6.11}$, a regra aplicada vai habilitar a aplicação da regra $MoveTri_{1.7}$, pois o vértice novo no lado

direito de qualquer uma das duas regras é o mesmo vértice adicionado no lado esquerdo da regra $MoveTri_{1.7}$.

4. CONCLUSÕES

Neste trabalho foi definida uma tradução de gramática de grafos controlada para gramática de grafos, que teve como resultado uma gramática GG', onde o controle da gramática GG' é dado pelos seus dados e não por uma estrutura de controle auxiliar, como em uma gramática de grafos controlada. Assim, através dessa tradução foi possível descrever GGC na linguagem Event-B e, com isso, integrar gramática de grafos controlada na técnica de prova de teoremas para gramática de grafos. Esta técnica permite a análise de propriedades de sistemas que foram especificados por uma gramática de grafos com estruturas de controle para a aplicação das regras.

5. REFERÊNCIAS BIBLIOGRÁFICAS

CAVALHEIRO, S. A. d. C. **Relational approach of graph grammars**. 2010. Doutorado em Ciência da computação— Universidade Federal do Rio Grande do Sul - UFRGS.

CRAIGEN, D.; GERHART, S.; RALSTON, T. **An International Survey of Industrial Applications of Formal Methods**: Volume 1 Purpose, Approach, Analysis,and Conclusions.

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 julho 2014. Online. Disponível em: http://www.event-b.org/

DWYER, M. B.; HATCLIFF, J.; ROBBY, R.; PUaSUaREANU, C. S.; VISSER, W. Formal Software Analysis Emerging Trends in Software Model Checking. In: **FUTURE OF SOFTWARE ENGINEERING**, 2007. FOSE '07, 2007. **Anais...**[S.l.: s.n.], 2007. p.120–136.

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.

RENSINK, A. **Explicit State Model Checking for Graph Grammars**, Springer, V. 5065, p.114–132, 2008