



Universidade: presente!



XXXI SIC

21.25. OUTUBRO • CAMPUS DO VALE

Editor Gráfico para a Ferramenta Verigraph

Arthur Lucena Fuchs

Orientação: Prof. Dra. Leila Ribeiro e Prof. Dr. Rodrigo Machado

Introdução

Gramáticas de grafos são um modelo matemático que permite a especificação de sistemas de forma visual e intuitiva. Para isso usamos grafos para representar estados do sistema e regras de reescrita de grafos (chamadas regras de transformação) para representar transições entre esses estados.

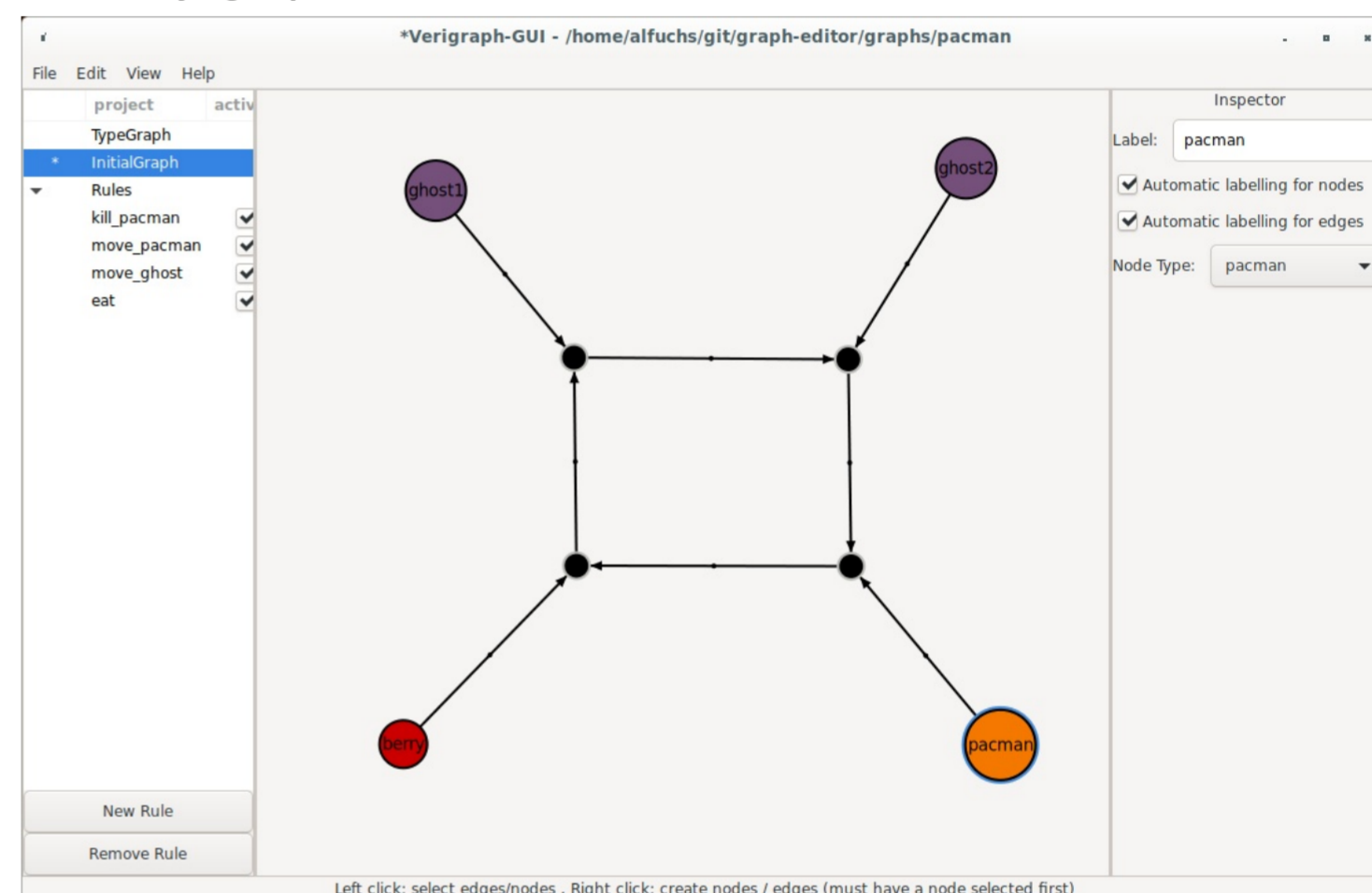
Em 2016 foi iniciado, dentro do projeto $\sqrt{\text{eriTeS}}$ [1], o desenvolvimento do Verigraph[2], uma ferramenta de transformação de grafos que disponibiliza diversas análises sobre modelos de gramáticas de grafos. O Verigraph suporta um alto nível de abstração não disponível em outras ferramentas, que foram implementadas visando tipos concretos de grafos.

Até o momento, o Verigraph disponibiliza uma interface via linha de comando, recebendo como entrada arquivos de texto gerados por outras ferramentas. O problema é que essas ferramentas se utilizam de gramáticas menos expressivas do que a gramática utilizado pelo Verigraph, dificultando a especificação de uma gramática aceitável para realizar algumas análises.

Por isso foi considerado necessário o desenvolvimento de um editor gráfico para especificar gramáticas de grafo para o Verigraph. O editor está em desenvolvimento no momento e permite a criação e manipulação de grafos e regras, gerando uma gramática simplificada para a ferramenta.

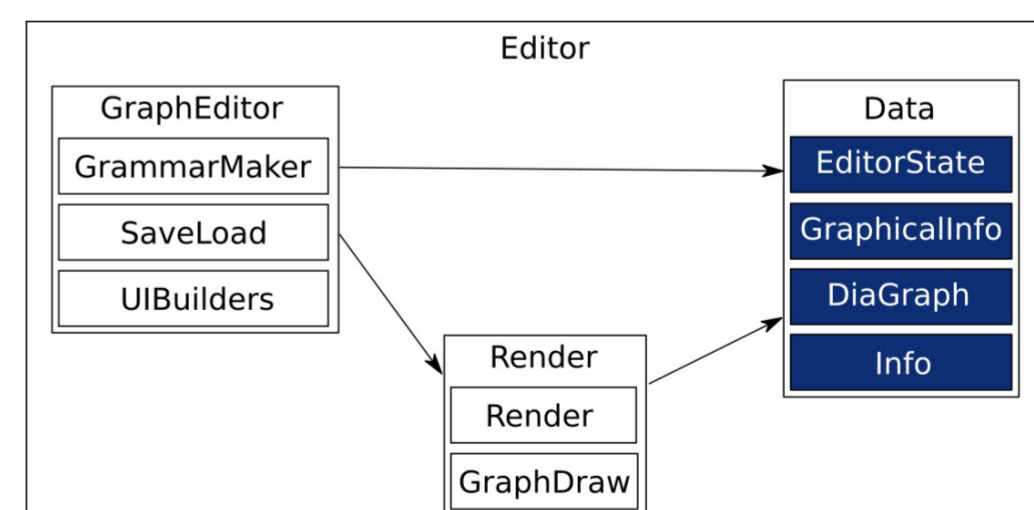
Verigraph-GUI

O editor, apelidado de Verigraph-GUI[3], pode ser visto na figura abaixo, que mostra o grafo inicial de um modelo que representa uma versão simplificada do jogo pacman.



O Verigraph-GUI foi desenvolvido usando a linguagem de programação Haskell para poder utilizar estruturas da ferramenta Verigraph e assim facilitar sua integração. Para criar e mostrar a interface foi utilizado o framework Gtk+3, através da biblioteca GI.Gtk.

A figura abaixo mostra como o editor foi organizado internamente. Retângulos em branco representam módulos e retângulos em azul representam estruturas de dados. Setas indicam dependências.



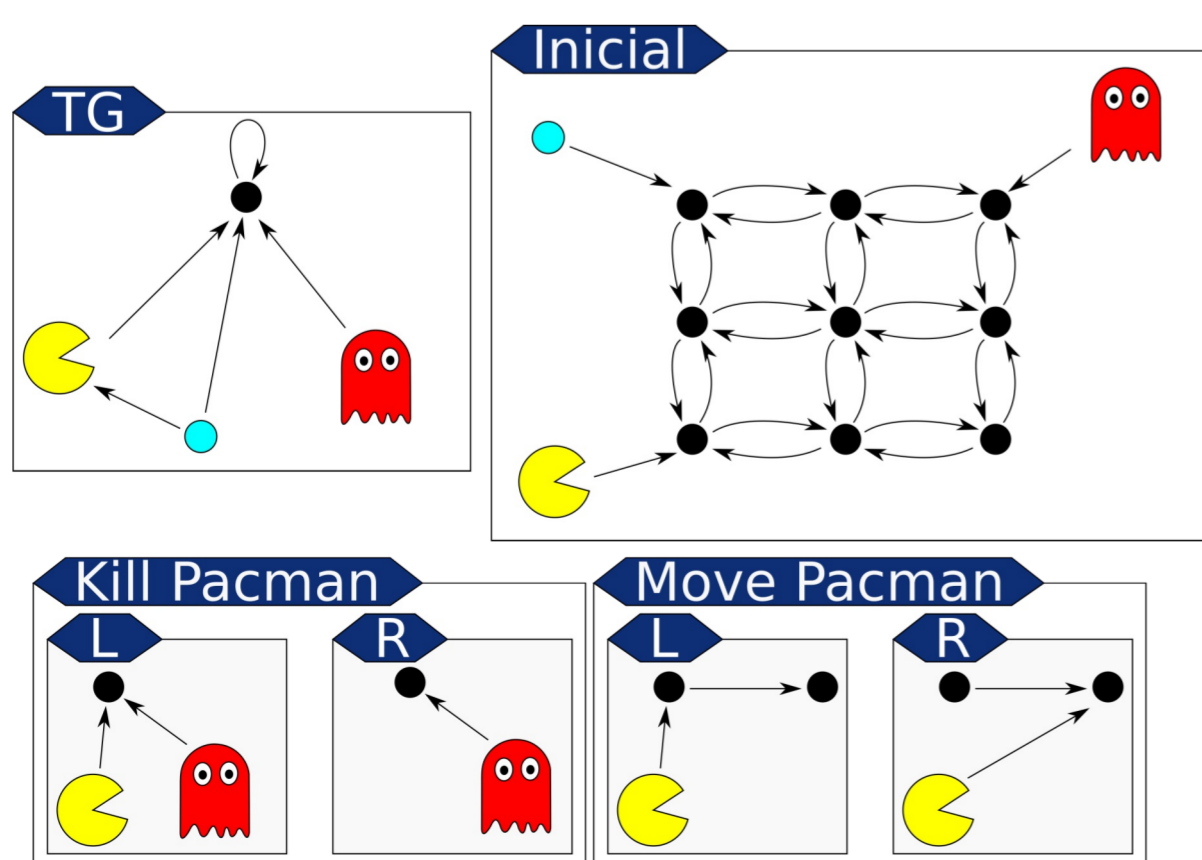
O Verigraph-GUI ainda não permite a especificação de gramáticas com *condições negativas de aplicação* ou *gramáticas de segunda ordem*. No momento editor também está limitado momentaneamente a exportar a gramática de grafos criada para o formato de arquivo padrão do AGG, ferramenta de transformação de grafos que vem sendo utilizada para especificar gramáticas de grafos para o Verigraph.

Gramática de Grafos

O editor em desenvolvimento permite criar gramáticas de grafos tipados. Grafos tipados são duplas $(G, \text{typeG: TG} \rightarrow G)$ onde TG é um grafo chamado de *grafo-tipo*, G é um grafo e typeG é um mapeamento de TG para G.

Uma gramática de grafos tipados é uma estrutura que contém um **grafo-tipo**, um **grafo inicial** e um conjunto de **regras de transformação**.

Para ilustrar, apresentamos a gramática abaixo. Essa gramática modela uma versão simplificada do famoso jogo pacman. TG é o grafo-tipo (ele define os tipos de elementos possíveis de se colocar em cada grafo) *Inicial* representa um possível nível do jogo e *Kill Pacman* e *Move Pacman* são duas regras possíveis para a gramática.



Referências

- [1] Website do grupo $\sqrt{\text{eriTeS}}$. <http://www.ufrgs.br/verites>
- [2] Repositório do Verigraph. <https://github.com/Verites/verigraph>
- [3] Repositório do Verigraph-GUI. <https://github.com/artFuchs/graph-editor>

apoio: