

# KEMS - Um demonstrador de teoremas multi-estratégia

Adolfo Neto<sup>1</sup>

<sup>1</sup>Departamento Acadêmico de Informática – Universidade Tecnológica Federal do Paraná  
Av. Sete de Setembro, 3165, Rebouças, Curitiba, Brasil, CEP 80230-901.

adolfo@utfpr.edu.br

**Abstract.** Neste artigo apresentaremos o KEMS, um demonstrador de teoremas multi-estratégia de código aberto. O objetivo é apresentar as razões para usar e estudar o KEMS, além de mostrar onde e como contribuir com o projeto KEMS.

**Resumo.** In this paper we present KEMS, an open source multi-strategy theorem prover. Our objective is to show why to use and study KEMS, as well as where and how to contribute with the KEMS project.

## 1. Introdução

Demonstradores de teoremas são programas de computador que recebem como entrada um problema lógico e que produzem como saída:

1. uma resposta sim/não: “sim” caso o teorema seja válido num determinado sistema lógico, e “não” em caso contrário;
2. em alguns casos, além da resposta sim/não, os demonstradores de teoremas produzem uma demonstração (ou uma refutação) do possível teorema recebido como entrada.

Os problemas lógicos são descritos como uma ou mais fórmulas lógicas que enunciam um possível teorema.

Os demonstradores de teoremas são também conhecidos como “provadores” de teoremas, uma tradução mais direta porém menos correta do termo em inglês que descreve este tipo de sistema: *theorem prover*. Demonstradores de teoremas tem diversas aplicações em Inteligência Artificial, Verificação Formal de Sistemas, além do uso no ensino de lógica e matemática. Para maiores detalhes sobre o que são teoremas, demonstrações, sistemas lógicos, etc. um bom ponto de partida é a leitura de [da Silva et al. 2006].

O KEMS [Neto 2006, Neto 2007] é um demonstrador de teoremas multi-estratégia. KEMS é uma abreviação (parcial) de *KE Multi-Strategy theorem prover*. É um demonstrador de teoremas baseado no método de tablôs KE que permite que o usuário escolha entre várias estratégias qual delas quer usar para demonstrar os teoremas. Uma tela típica do KEMS contém a janela principal do KEMS, um editor de problemas e um visualizador de demonstrações (ver Figura 1).

O objetivo deste artigo é apresentar o KEMS como um software livre que está aguardando contribuições de estudantes de graduação e pós-graduação. A participação como desenvolvedor no projeto KEMS poderá auxiliar estes estudantes no aprendizado de lógica, programação e engenharia de software, disciplinas comuns a todos os cursos de graduação na área de computação no Brasil.

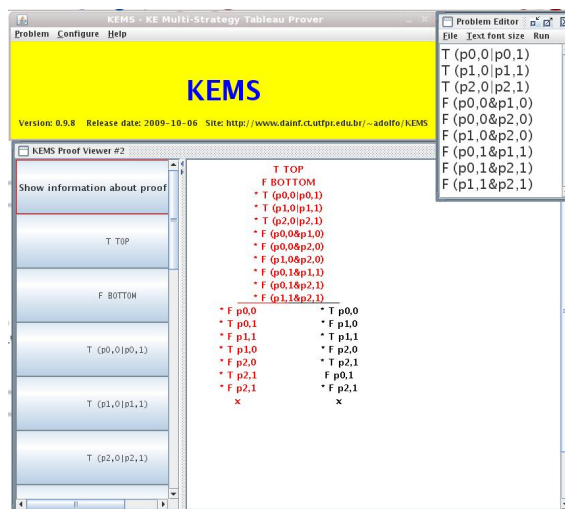


Figura 1. Tela com janelas do KEMS.

## 2. Métodos de Tablôs e o Sistema KE

Os métodos de tablôs são bastante conhecidos entre os lógicos. Vários livros de introdução à lógica, sejam os direcionados para alunos dos cursos da área de computação (por exemplo, [da Silva et al. 2006]), ou os voltados para estudantes de filosofia [Mortari 2001], usam os métodos de tablôs como exemplo de sistema dedutivo de fácil aprendizagem.

O método de tablôs mais conhecido é aquele que é comumente chamado de tablôs analíticos, popularizado entre os lógicos no livro de Raymond Smullyan [Smullyan 1968] mas desenvolvido anteriormente de forma independente pelos lógicos Lis e Beth.

Em 1988, Marco Mondadori desenvolveu um novo método de tablôs, baseado nos tablôs analíticos porém mais adequado para a lógica clássica (a mais utilizada dentre as lógicas). Depois disso, Marcello D'Agostino escreveu vários artigos (o mais completo sendo [D'Agostino 1999]) mostrando as qualidades deste sistema. A principal delas é a maior eficiência computacional em relação ao sistema de tablôs analíticos.

### 2.1. Implementações do Sistema KE

Desde então algumas implementações do sistema KE foram feitas. MacKE [Pitt 1995] e WinKE [D'Agostino and Endriss 1998] não são demonstradores de teoremas, mas sim são assistentes de demonstração interativos. Eles apenas ajudam o usuário a desenvolver uma demonstração, não tem por objetivo demonstrar teoremas sem o auxílio do usuário. São, portanto, mais voltados ao ensino de lógica. Além disso, não há código-fonte disponível para nenhum dos dois.

Outras implementações do sistema de tablôs KE foram feitas (veja em [Neto 2010]) mas por falta de espaço não serão discutidas aqui. Poucas possuem o código-fonte aberto, nenhuma outra utiliza a linguagem Java e nenhuma outra deixa seu código-fonte disponível num repositório de código-fonte livre.

Um dos diferenciais do KEMS em relação a outras implementações do sistema KE é que ele implementa o sistema de tablôs KE utilizando a linguagem Java [Sun Microsystems 1995a] (uma das linguagens mais utilizadas no mundo, se não a mais utilizada) e seu código-fonte está hospedado no repositório de código livre GitHub [GitHub Inc. 2010]. Por-

tanto o acesso a seu código é fácil e grande parte dos estudantes de graduação da área de computação podem entender o seu código, desde que conheçam a arquitetura do sistema e suas bases teóricas (lógica e o sistema KE).

### 3. Razões para usar e estudar o KEMS

São várias as razões que podemos identificar para que o KEMS seja utilizado como software e para que seu código-fonte seja estudado. Em primeiro lugar, o KEMS, em seu estágio atual, pode ser utilizado como ferramenta auxiliar no ensino da disciplina “lógica para computação” nos cursos de graduação da área de computação. Com ele podem ser produzidas demonstrações utilizando o sistema KE com várias lógicas (4 na versão atual).

Além disso, ele pode ser usado para ilustrar o conceito de algoritmo não-determinístico, pois cada estratégia é uma implementação determinística de um algoritmo não-determinístico (mais detalhes em [Neto 2007]).

Uma terceira razão é que o KEMS é um exemplo concreto de um sistema de um tamanho razoável. Em sua versão atual, ele contém mais de 30 mil linhas somente de código em Java – contadas usando o *software* CLOC [Danial 2006] – sem contar as partes escritas em outras linguagens.

No código do KEMS podem ser observados, além do uso da linguagem de programação Java, a utilização dos seguintes artefatos de software:

- a linguagem **AspectJ** [Eclipse Foundation 1997], uma extensão à linguagem Java que implementa o paradigma de orientação a aspectos;
- o gerador de analisadores sintáticos **CUP** [Hudson et al. 1999];
- o gerador de analisadores léxicos **JFlex** [Klein 2009];
- o pacote para registro (*logging*) **Log4j** [Apache Software Foundation 2007].

O código-fonte é desenvolvido usando o Ambiente Integrado de Desenvolvimento **Eclipse** [Eclipse contributors and others 2000], na versão 3.4.2, com o compilador para Java **Sun JDK 1.6** e o plugin **AJDT** [IBM Corp. and others 2000], versão 2.0.1.

Por fim, ao estudar o código-fonte do KEMS, podem ser vistas implementações de diversos Padrões de Projeto [Gamma et al. 1995] tais como:

- Facade (na classe `ProverFacade`);
- AbstractFactory (nas classes `FormulaFactory` e `SignedFormulaFactory`);
- Strategy (em várias classes que implementam as estratégias do sistema);
- Flyweight e Composite (nas classes que representam as fórmulas lógicas).

### 4. Espaço para Contribuições

Testes e avaliações com o KEMS já foram documentadas em [Neto 2007] e em outras publicações relacionadas ao projeto. Através destes testes e da análise do código, pudemos verificar que ainda há bastante espaço para melhorias no sistema, dentre as quais destacamos as seguintes:

- Melhorias na interface gráfica;
- Melhorias no desenho das árvores de demonstração;
- Criação, a partir do código do KEMS, de um pacote independente para a representação eficiente de fórmulas lógicas;

- Nova implementação dos analisadores léxicos e sintáticos;
- Documentação do código (muitas partes do código não estão adequadamente documentadas);
- Melhorar a eficiência da manipulação de cadeias de caracteres;

## 5. Considerações Finais

O KEMS é um *software* de código aberto que pode ser utilizado de várias formas por estudantes de cursos de graduação na área de computação. Mais informações sobre o projeto podem ser encontradas no site do KEMS:

<http://adolfoneto.wikidot.com/KEMS>

Lá podem ser encontradas publicações relacionadas ao projeto assim como o endereço eletrônico para contato com a equipe de desenvolvimento.

Desde 22 de março de 2010, o KEMS pode ser executado usando a tecnologia Java Web Start [Sun Microsystems 1995b]. Desta forma, não é necessário instalá-lo na máquina do usuário para que possa ser executado. O usuário apenas precisa ter uma máquina virtual Java instalada em seu computador. A conversão do jar do KEMS em uma aplicação Java Web Start foi feita principalmente por Emerson Shigueo Sugimoto. O *link* para a aplicação pode ser encontrado no site do KEMS.

## Referências

- Apache Software Foundation (2007). *Log4j*. Disponível em: <http://logging.apache.org/log4j/>. Acesso em: 18 fev. 2010.
- da Silva, F. S. C., Finger, M., and de Melo, A. C. V. (2006). *Lógica para Computação*. Thomson Learning.
- D’Agostino, M. (1999). Tableau methods for classical propositional logic. In et al., M. D., editor, *Handbook of Tableau Methods*, chapter 1, pages 45–123. Kluwer Academic Press.
- D’Agostino, M. and Endriss, U. (1998). WinKE: A Proof Assistant for Teaching Logic. In *Proceedings of the First International Workshop on Labelled Deduction, Freiburg*.
- Danial, A. (2006). *CLOC – Count Lines of Code*. Disponível em: <http://cloc.sourceforge.net/>. Acesso em: 18 fev. 2010.
- Eclipse contributors and others (2000). *Eclipse*. Disponível em: <http://eclipse.org/>. Acesso em: 18 fev. 2010.
- Eclipse Foundation (1997). *AspectJ*. Disponível em: <http://www.eclipse.org/aspectj/>. Acesso em: 18 fev. 2010.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns*. Addison-Wesley.
- GitHub Inc. (2010). *GitHub*. Disponível em: <http://github.com>. Acesso em: 18 fev. 2010.
- Hudson, S., Flannery, F., and Ananian, C. S. (1999). *CUP – LALR Parser Generator for Java*. Disponível em: <http://www2.cs.tum.edu/projects/cup/>. Acesso em: 18 fev. 2010.

- IBM Corp. and others (2000). *Eclipse AspectJ Development Tools*. Disponível em: <http://eclipse.org/ajdt>. Acesso em: 18 fev. 2010.
- Klein, G. (2009). *JFlex – The Fast Scanner Generator for Java*. Disponível em: <http://jflex.de/>. Acesso em: 18 fev. 2010.
- Mortari, C. (2001). *Introdução à Lógica*. Editora da Unesp.
- Neto, A. (2006). *KEMS*. Disponível em: <http://www.dainf.ct.utfpr.edu.br/~adolfo/KEMS>. Acesso em: 19 fev. 2010.
- Neto, A. (2007). *A Multi-Strategy Tableau Prover*. PhD thesis, University of Sao Paulo. Disponível em: <http://www.dainf.ct.utfpr.edu.br/~adolfo/Thesis/>. Acesso em: 19 fev. 2010.
- Neto, A. (2010). *Theorem Provers Based on Tableau Systems*. Disponível em: <http://bit.ly/9fwkz9>. Acesso em: 19 fev. 2010.
- Pitt, J. (1995). MacKE: Yet Another Proof Assistant & Automated Pedagogic Tool. In Baumgärtner, P., Hähle, R., and Posegga, J., editors, *Theorem Proving with Analytic Tableaux and Related Methods*. Springer-Verlag.
- Smullyan, R. (1968). *First-Order Logic*. Dover Publications.
- Sun Microsystems (1995a). *Java*. Disponível em: <http://java.sun.com/>. Acesso em: 18 fev. 2010.
- Sun Microsystems (1995b). *Java Web Start Technology*. Disponível em: <http://java.sun.com/javase/technologies/desktop/javawebstart/index.jsp>. Acesso em: 31 mar. 2010.