

MAC0239 EP1 - Tableaux Semânticos

Gabriel de Russo e Carmo - 9298041

Matheus de Mello Santos Oliveira - 8642821

Victor de Oliveira Colombo - 8988657

Victor Wichmann Raposo - 9298020

Uso

A comunicação entre nosso programa e o usuário se dá por meio da entrada e saída padrão.

- **Dependências:**

JDK-8

- **Compilando e Executando:**

```
javac Tableaux.java  
java Tableaux
```

- **Input:**

Na primeira linha recebe o número de premissas ***N***.

As ***N*** linhas seguintes descrevem cada premissa: todas as subfórmulas devem ser parentizadas, com exceção das atômicas, e bem formadas. O formato das fórmulas pode ser encontrado no enunciado.

A última linha recebe a única consequência lógica seguindo as

mesmas restrições.

- **Output:**

Se o sequente é válido ou um contra-exemplo.

Organização

Conforme a especificação, foi utilizada a linguagem Java para implementação.

O projeto conta com duas classes:

- **ExpressionTree:** Responsável pela representação de uma expressão na forma de árvore. Possui uma classe interna **Node** que modela o nó da árvore.
 - **Tableaux:** Responsável pela prova do teorema em si. Utiliza a ExpressionTree para armazenar as expressões. Possui as seguintes classes internas para auxílio:
 - **Triple:** Como o Java não contém suporte nativo a n-uplas, foi necessário a implementação de uma estrutura para armazenar triplas.
 - **Formula:** Associa um valor a uma ExpressionTree (true ou false).
 - [Javadoc](#)
-

Expressões

As expressões são armazenadas na forma de uma árvore binária. Embora sejam mais complexas de construir, sua manipulação é mais

simples quando comparadas com expressões puramente textuais.

Para construir tal árvore usamos uma estratégia recursiva: Cada expressão na forma $A X B$, sendo A e B subexpressões e X um operador, é transformada num nó X pai de duas subárvores geradas por A e B .

O processo tem fim quando se chega a um átomo.

Nessa etapa surgiu a dificuldade do tratamento do operador *not* pois ele é **unário**, ou seja, é aplicado sobre apenas um operando. Foi necessário tratar separadamente este caso: o nó correspondente contém apenas o filho direito.

Resolução

Esta etapa foi implementado o algoritmo Tableaux Semântico para resolução, que consiste superficialmente em:

1. Enquanto existem ramos abertos não saturados aplique expansões alfas e betas.
2. Após todos os ramos estarem saturados cheque por ramos abertos. Se existe algum o sequente é inválido e a valoração deste é um contra-exemplo. Caso contrário o teorema é válido.

Embora o algoritmo teórico seja simples, ele possui uma série de detalhes de implementação omitidos.

Um grande impasse foi encontrar o meio de representar o tableaux. Optamos por não representar a árvore explicitamente, guardando apenas o ramo atual, seguindo a estratégia de Busca em Profundidade juntamente com uma Pilha de Ramos, conforme sugerido no Capítulo 4 do livro *Lógica Matemática para Cientistas de Computação*, Flávio

Soares Corrêa da Silva & Marcelo Finger & Ana Cristina Vieira de Melo.

Para melhoria de performance utilizamos algumas otimizações:

- Aplicar expansões alfas antes das expansões betas.
 - "Aproveitar" o processo de aplicar expansões alfas para marcar futuras expansões betas.
 - Aplicar expansões betas nos menores ramos possíveis.
-

Conclusão

O projeto foi de grande aprendizado para a equipe, uma vez que ele nos ensinou uma visão diferente da que estávamos acostumados em matérias anteriores, onde heurísticas e decisões de projetos foram essenciais para a uma boa execução.