Prova automatizada de teoremas em lógica proposicional

Projeto Orientado em Computação

Aluno: Frederico Martins Biber Sampaio

Orientador: Moisés Henrique Ramos Pereira

Coorientadora: Miriam Lourenço Maia

UNIBH - Centro Universitário de Belo Horizonte

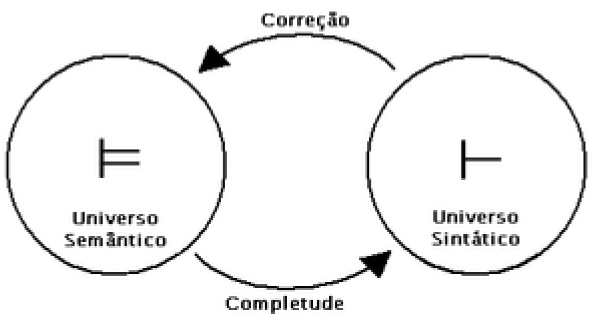
Motivação

- Elaborar um software para auxílio didático e em pesquisa acadêmica, em especial alunos de matemática discreta.
- Contribuir para divulgação da área de automação de sistemas de inferência.

Problema SAT

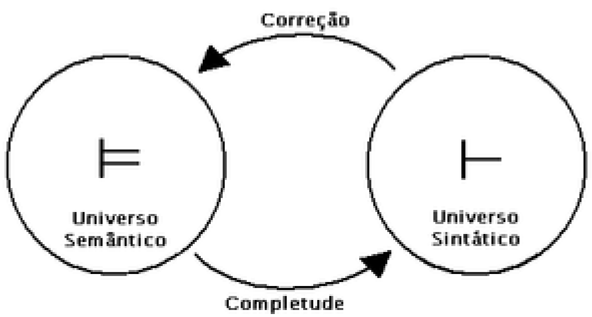
- O problema da "satisfatibilidade" booleana (SAT) trata da verificação de fórmula em lógicas proposicionais.
- Existe alguma interpretação que torne uma fórmula verdadeira?
- Teorema de Cook, de 1971, origem do termo NP-completo.
- O problema SAT é o primeiro problema NP-completo estudado.

Sistemas lógicos



- Universo semântico de fórmula lógica é obtido pela sua interpretação, ou seja, valores aplicados a suas variáveis.
- Universo sintático é obtido pelas derivações, ou seja, pela aplicação das inferências sobre os elementos simbólicos, sem considerar os resultados concretos (prova).

Solução do sistema SAT



- Considerando-se um sistema lógico, qual é a melhor forma de verificação SAT?
- Dado um sistema lógico, uma solução SAT por método sintático é equivalente a uma solução por método semântico?
- Sistema lógico completo: $\Gamma \vDash \varphi \rightarrow \Gamma \vdash \varphi$
- Sistema lógico correto: $\Gamma \vdash \varphi \rightarrow \Gamma \vDash \varphi$

Lógica de primeira ordem

- O teorema da completude de Gödel, de 1929, prova que as formulas em lógica de primeira ordem são completas e corretas.
- Lógica proposicional, além de poder ser considerada com subconjunto da lógica de primeira ordem, também é completa e correta.
- Assim, nesses casos, uma prova por dedução (sintática) equivale a uma demonstração por interpretação (semântica).

Métodos de verificação SAT

- Para lógica proposicional, a tabela verdade é um método semântico elaborada desde o início de 1920.
- Algoritmo DPLL (Davis-Putnam-Logemann-Loveland) é um método sintático de lógica proposicional que lida com fórmulas em CNF (forma conjuntiva normal).
- Dedução natural e a dedução sequencial, adotada em cursos e livros de matemática discreta, são métodos sintáticos para lógicas de predicado.
- Tableau semânticos, método de inferência, ou seja, sintático, que apresenta solução em forma de árvore de prova. Adequado para lógicas de predicado.

Tabela verdade

- Algoritmos proposto independentemente no início dos anos de 1920 por Ludwig Wittgenstein e Emil Leon Post.
- Busca por todas as interpretações de um sistema lógico.
- Provou que a lógica proposicional é *decidível*.
- Exemplo para o sistema: $(\neg A \vee B) \vdash (A \rightarrow B)$:

```
/-----------------------------\
|A|B|((~(A) v B) -> (A -> B))|
|-+-+-------------------------|
|T|T|  F    T    T*    T      |
|T|F|  F    F    T*    F      |
|F|T|  T    T    T*    T      |
|F|F|  T    T    T*    T      |
\-----------------------------/
```

Tableau semântico

- Proposto originalmente por Evert Willem Beth em forma de tabela dividida em coluna verdadeira e falsa.
- Baseado no princípio de eliminação de regras de inferência com cortes provado por Gerhard Gentzen.
- Utiliza regras de inferência simples (sem cortes).

$$\begin{array}{l} A \wedge B \\ \hline A \\ B \end{array} \qquad \begin{array}{c|c} \multicolumn{2}{c}{\neg(A \wedge B)} \\ \hline \neg A & \neg B \end{array} \qquad \begin{array}{c|c} \multicolumn{2}{c}{A \vee B} \\ \hline A & B \end{array} \qquad \begin{array}{l} \neg(A \vee B) \\ \hline \neg A \\ \neg B \end{array}$$

$$\begin{array}{c|c} \multicolumn{2}{c}{A \to B} \\ \hline \neg A & B \end{array} \qquad \begin{array}{l} \neg(A \to B) \\ \hline A \\ \neg B \end{array} \qquad \begin{array}{l} \neg\neg A \\ \hline A \end{array}$$

Prova por contradição

- Prova baseada em contradição.

  - $\Gamma \cup \{\neg\varphi\} \vdash \perp \quad \rightarrow \quad \Gamma \vdash \varphi$

  - $\vdash \varphi$ (φ é uma tautologia)
    - por contradição equivale à $\{\neg\varphi\} \vdash \perp$

  - $\Gamma \vdash$ (Γ é uma contradição)
    - equivale à $\Gamma \vdash \perp$

Tableau de Smullyan

- Primeira apresentação do tableau de Beth utilizando árvore de prova, formato usado atualmente.
- Cada nó da árvore representa uma fórmula.
- Cada nó é sinalizado como V (verdadeiro) ou F (falso)

$$\frac{T\,A \wedge B}{\begin{array}{l} T\,A \\ T\,B \end{array}} \qquad \frac{F\,A \wedge B}{F\,A \mid F\,B} \qquad \frac{T\,A \vee B}{T\,A \mid T\,B} \qquad \frac{F\,A \vee B}{\begin{array}{l} F\,A \\ F\,B \end{array}}$$

$$\frac{T\,A \to B}{F\,A \mid T\,B} \qquad \frac{F\,A \to B}{\begin{array}{l} T\,A \\ F\,B \end{array}} \qquad \frac{T\neg A}{F\,A} \qquad \frac{F\neg A}{T\,A}$$

Tableau de Smullyan

- Exemplo para o sistema: $(\neg A \vee B) \vdash (A \rightarrow B)$

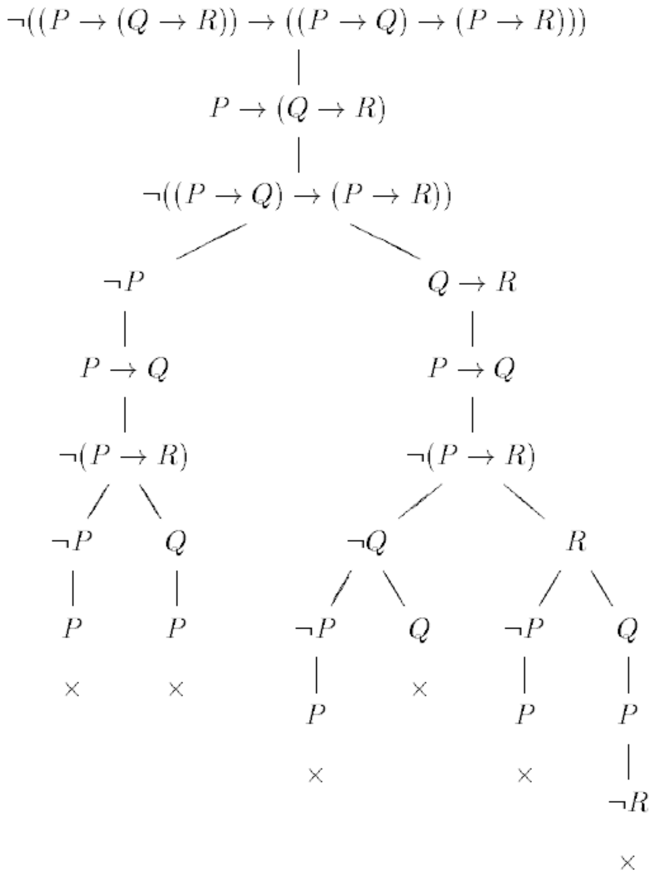
```
1 (T) (~(A)vB) {H}
2 (F) (A->B)  {H}
3 (T) A       {2 F->:l}
4 (F) B       {2 F->:r}

7 (T) ~(A) {1 Tv:l}            5 (T) B {1 Tv:r}
8 (F) A    {7 T~}              6 -X-   {5,4 closure}
9 -X-      {8,3 closure}
```

Classificação das fórmulas

- **α** (alfa): fórmulas cuja inferência expande linearmente um ramo, por exemplo, as fórmulas com negação (*not*) ou conjunção (*and*).
- **β** (beta): fórmulas cuja inferência expande a árvore por meio de uma nova ramificação como, por exemplo,as disjunções.
- **γ** (gama): fórmulas quantificadas universalmente.
- **δ** (delta): fórmulas quantificadas existencialmente.
- Literais ou atômicos são fórmulas que não podem derivar outras fórmulas por nenhuma regra de inferência, por serem indivisíveis.

Seleção do nó para inferência



- Prova com 29 nós (incluindo os fechamentos)

Seleção do nó para inferência

```
 1 (F) ((P->(Q->R))->((P->Q)->(P->R)))  {H}
 2 (T) (P->(Q->R))                      {1 F->:l}
 3 (F) ((P->Q)->(P->R))                 {1 F->:r}
 4 (T) (P->Q)                           {3 F->:l}
 5 (F) (P->R)                           {3 F->:r}
 6 (T) P                                {5 F->:l}
 7 (F) R                                {5 F->:r}

              /                   \

16 (F) P {2 T->:l}          8 (T) (Q->R) {2 T->:r}
17 -X-   {16,6 closure}

                                /          \

                14 (F) P {4 T->:l}        9 (T) Q {4 T->:r}
                15 -X-   {14,6 closure}

                                             /              \

                        12 (F) Q {8 T->:l}           10 (T) R {8 T->:r}
                        13 -X-   {12,9 closure}      11 -X-   {10,7 closure}
```

- Prova com 17 nós (incluindo os fechamentos)

Tableau com lema

- Busca evitar ramificações redundantes.

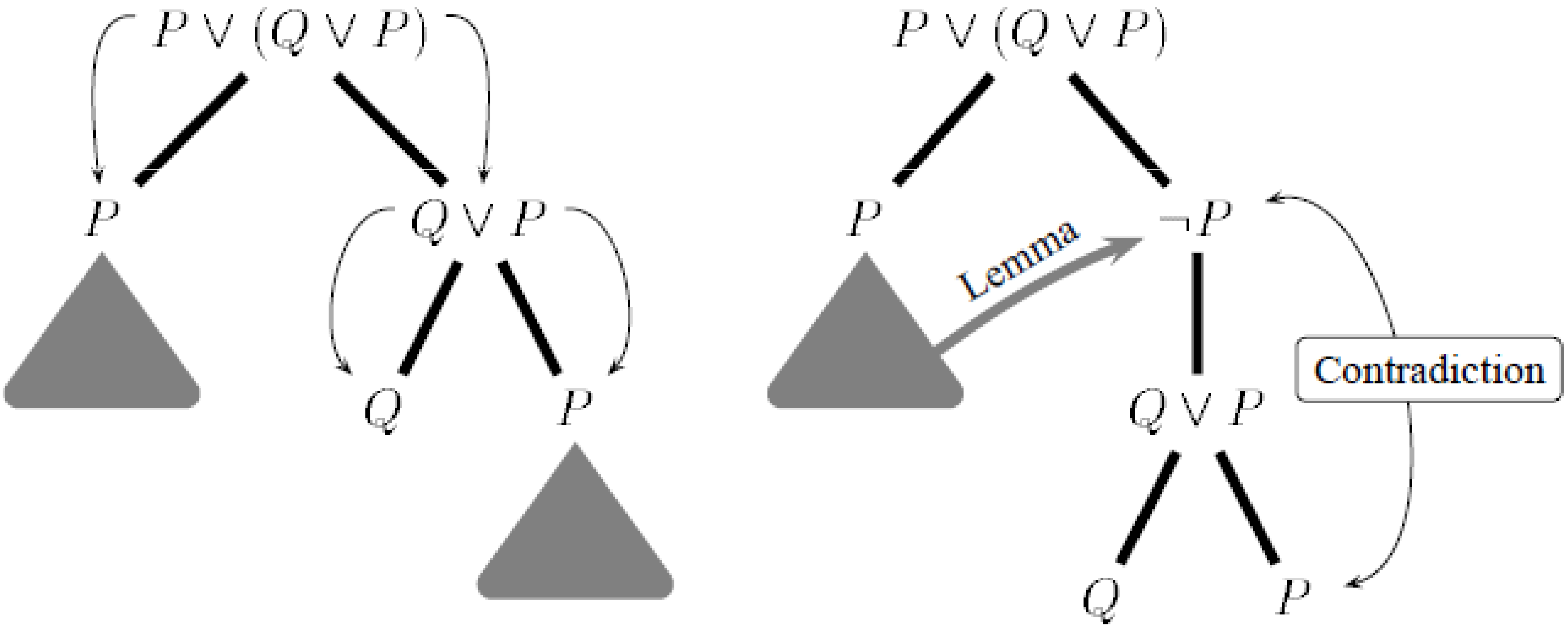


Tableau com lema

- $\Gamma \vdash \varphi, \Delta \rightarrow \Gamma \cup \{\neg\varphi\} \vdash \Delta$
  - O lema $\{\neg\varphi\}$ só é válido se φ é verdadeiro, ou seja, se φ leva ao fechamento do seu ramo original.
- A escolha do lema é o ponto fundamental (e complexo)
- Lema originado de uma fórmula falsa, gera um tableau com resultados incorretos.
- Os lemas não podem originar ramificações.

Tableau KE

- Proposto originalmente Marcello D'Agostino.
- Inclui regras de inferência com cortes.
  - ***modus ponens*, *modus tollens* e silogismo disjuntivo.**
- A única forma de ramificação é pelo princípio da bivalência, que é baseado no conceito de lema.

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma \vdash \Delta, A}{\Gamma \vdash \Delta}$$

Tableau KE

Disjunction Rules

$$\begin{array}{l} T\,A \lor B \\ F\,A \\ \hline T\,B \end{array}\ \mathrm{E}T{\lor}1 \qquad \begin{array}{l} T\,A \lor B \\ F\,B \\ \hline T\,A \end{array}\ \mathrm{E}T{\lor}2 \qquad \begin{array}{l} F\,A \lor B \\ \hline F\,A \\ F\,B \end{array}\ \mathrm{E}F{\lor}$$

Conjunction Rules

$$\begin{array}{l} F\,A \land B \\ T\,A \\ \hline F\,B \end{array}\ \mathrm{E}F{\land}1 \qquad \begin{array}{l} F\,A \land B \\ T\,B \\ \hline F\,A \end{array}\ \mathrm{E}F{\land}2 \qquad \begin{array}{l} T\,A \land B \\ \hline T\,A \\ T\,B \end{array}\ \mathrm{E}T{\land}$$

Implication Rules

$$\begin{array}{l} T\,A \to B \\ T\,A \\ \hline T\,B \end{array}\ \mathrm{E}T \to 1 \qquad \begin{array}{l} T\,A \to B \\ F\,B \\ \hline F\,A \end{array}\ \mathrm{E}T \to 2 \qquad \begin{array}{l} F\,A \to B \\ \hline T\,A \\ F\,B \end{array}\ \mathrm{E}F \to$$

Negation Rules

$$\begin{array}{l} T\neg A \\ \hline F\,A \end{array}\ \mathrm{E}T\neg \qquad \begin{array}{l} F\neg A \\ \hline T\,A \end{array}\ \mathrm{E}F\neg$$

Principle of Bivalence

$$\begin{array}{c|c} \hline T\,A & F\,A \end{array}\ \mathrm{PB}$$

Comparação de resultados

(A v B), (A v ¬B), (¬A v C), (¬A v ¬C) ⊨ ⊥

Tableau Clássico

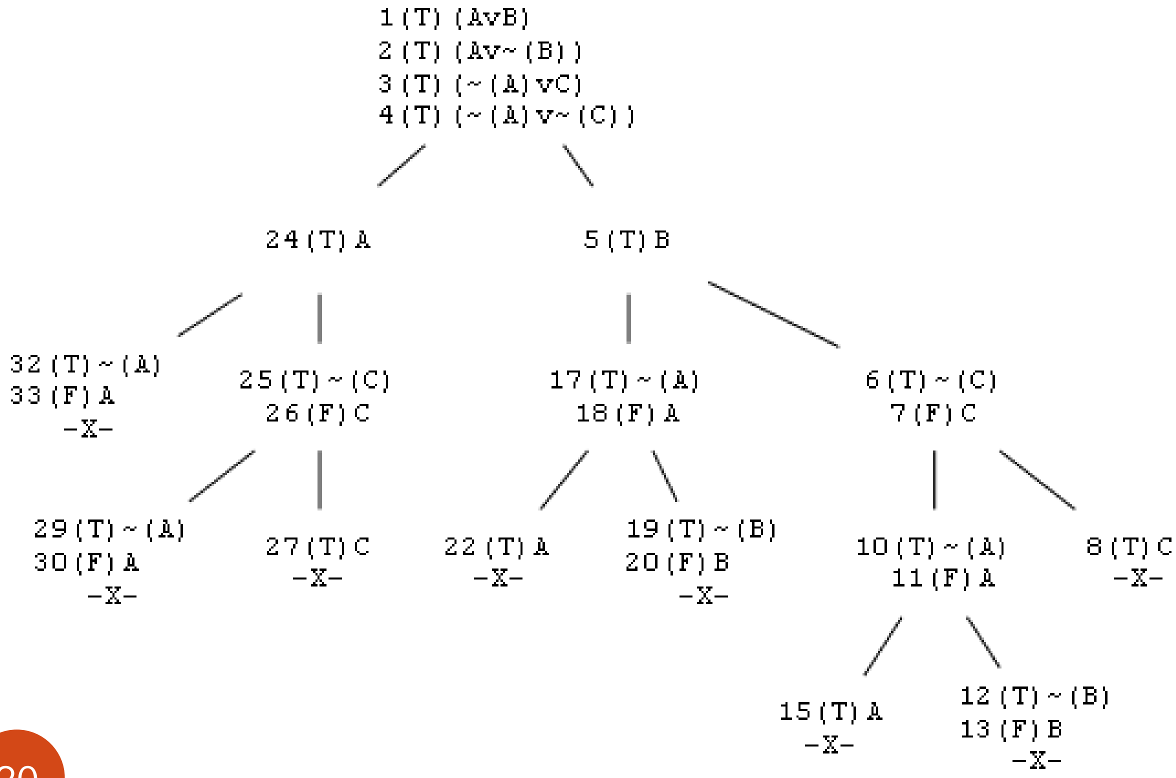
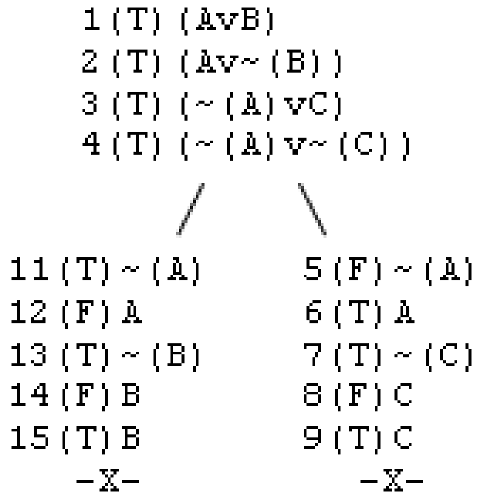


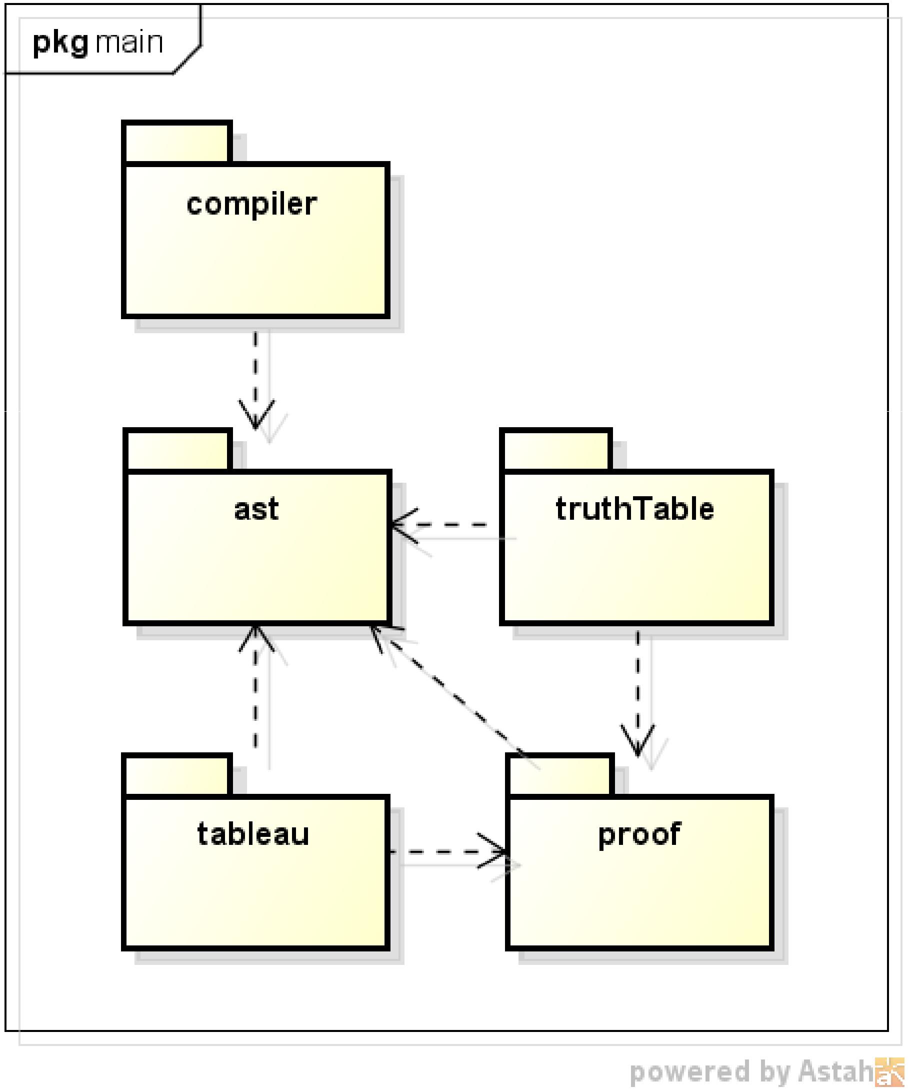
Tableau KE



Escolha do nó PB

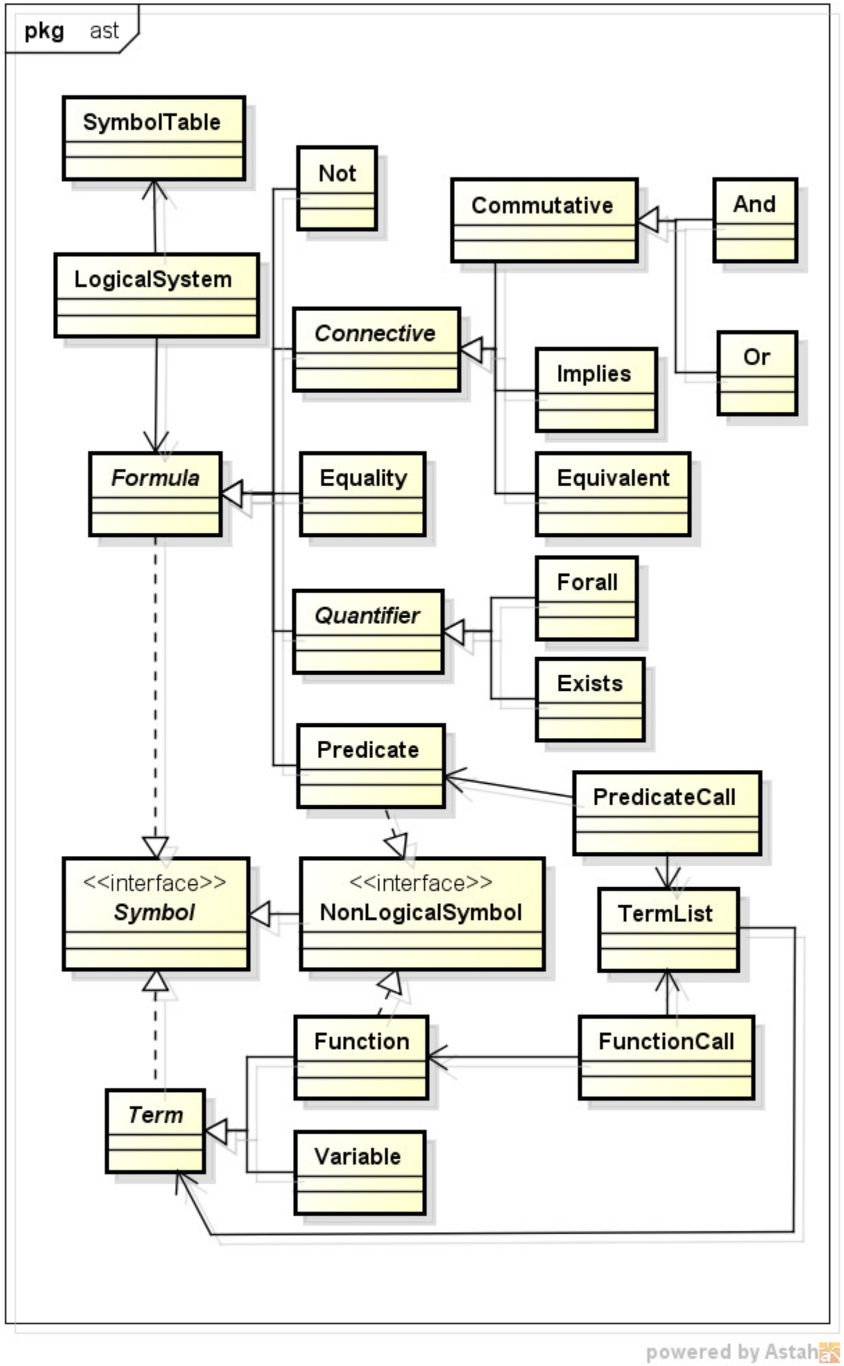
- É fundamental e existem diversas heurísticas.
- Fórmulas *E-analysed* , considerado o ramo ϕ, são:
  - Se a fórmula é do tipo α e as subfórmulas α_1 e α_2 ocorrem em ϕ .
  - Se a fórmula é do tipo β:
    - Se β_1 e $\neg\beta_2$ ocorre no ramo ϕ ou
    - Se $\neg\beta_1$ e β_2 ocorre no ramo ϕ.
- As fórmulas do PB são originadas de fórmulas β não *E-analysed*.
  - PB sempre será ou originado ou de β_1 ou β_2.
- Exemplo de heurística: escolher a subfórmula que ocorre mais vezes nas fórmulas β não analisadas (não *E-analysed*).

Design do software

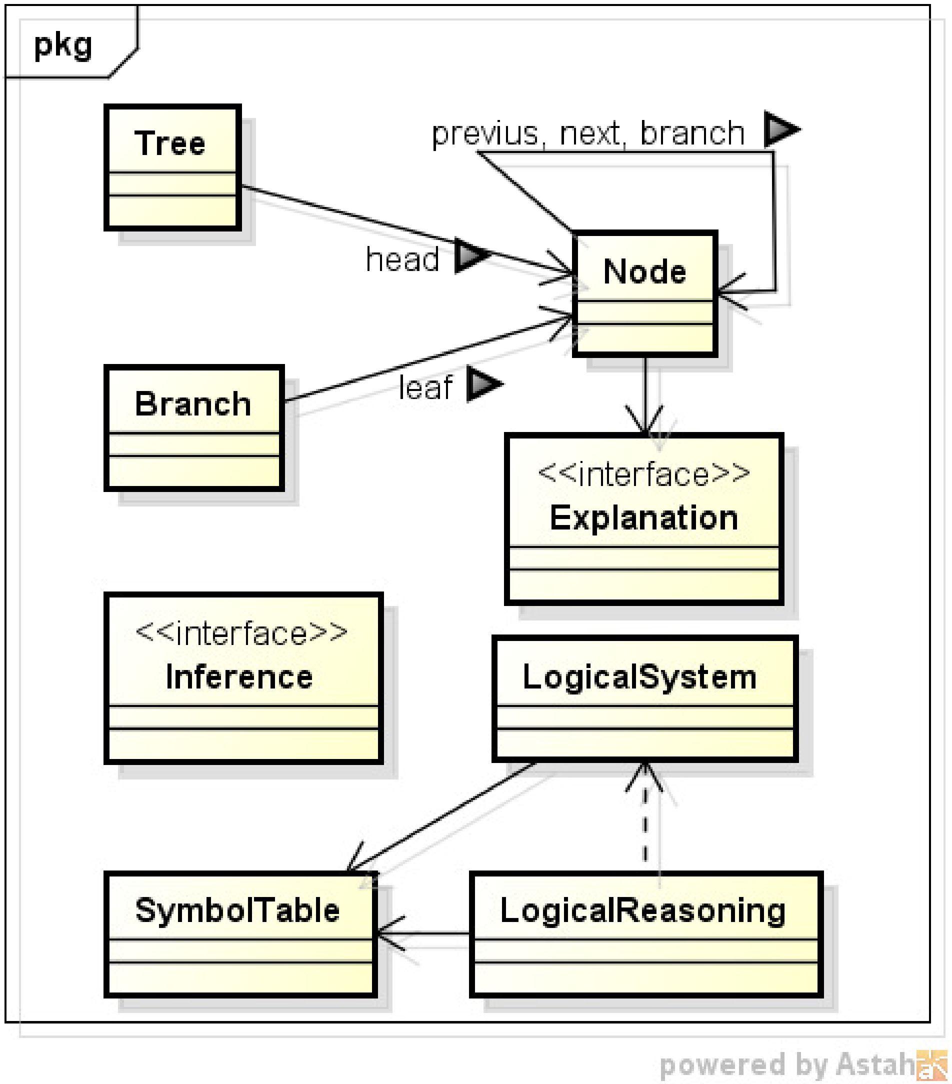


Módulo AST

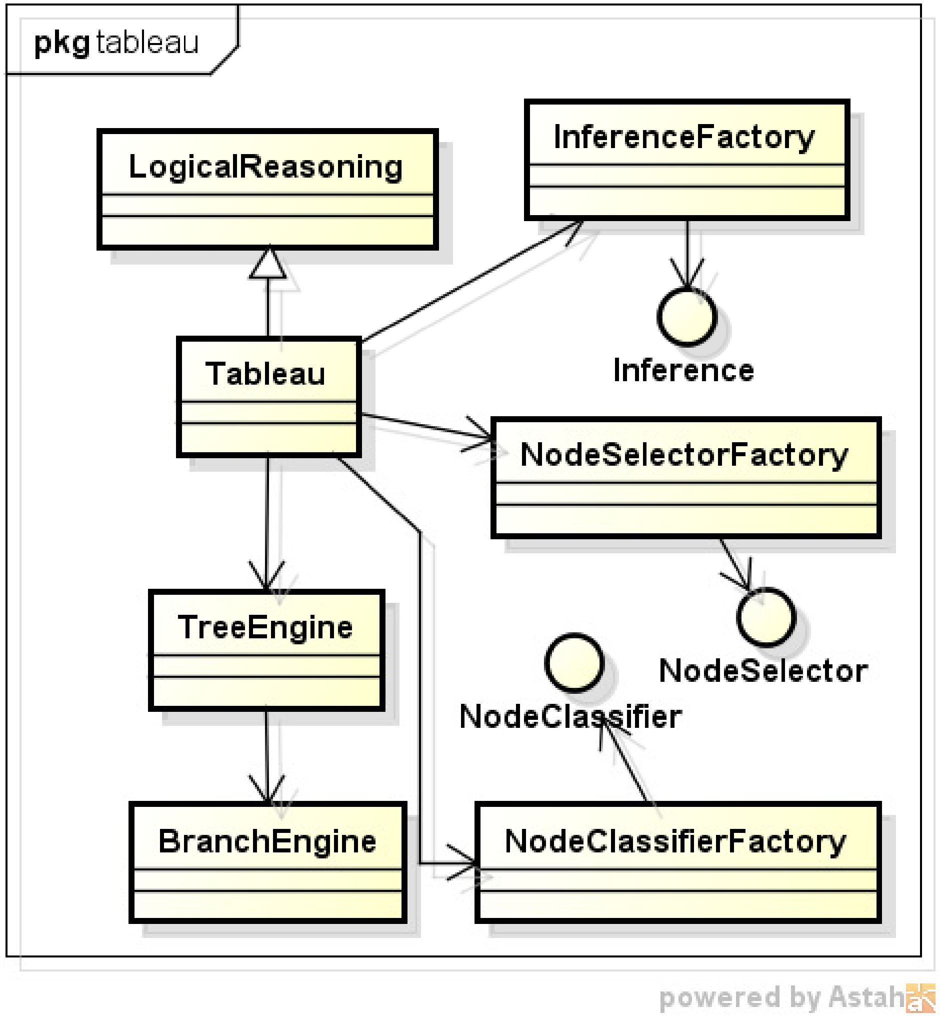
Objetos gerado pelo compilador



Módulo de mecanismos de prova



Módulo de tableau semântico



Exemplo da saída do terminal

```
Tableau:
Solução por Tableau Semântico ' em 2 ms
Inference rules = Smullyan's Tableau
Node selector   = Priority Node Selector
     1 | (F) ((P->(Q->R))->((P->Q)->(P->R)))  {H}
     2 | (T) (P->(Q->R))                      {1 F->:l}
     3 | (F) ((P->Q)->(P->R))                 {1 F->:r}
     4 | (T) (P->Q)                           {3 F->:l}
     5 | (F) (P->R)                           {3 F->:r}
     6 | (T) P                                {5 F->:l}
     7 | (F) R                                {5 F->:r}
     8 | | (T) (Q->R)                         {2 T->:r}
     9 | | | (T) Q                            {4 T->:r}
    10 | | | | (T) R                          {8 T->:r}
    11 | | | | -X-                            {10,7 closure}
    12 | | | (F) Q                            {8 T->:l}
    13 | | | -X-                              {12,9 closure}
    14 | | (F) P                              {4 T->:l}
    15 | | -X-                                {14,6 closure}
    16 | (F) P                                {2 T->:l}
    17 | -X-                                  {16,6 closure}
Total de vertices = 14
TAUTOLOGY!
```

Exemplo de compilação

`Ex:x(Ex:y(Q(x,h(y)))) |-`
`Fa:x(Ex:y((P(x,y)->Q(g(y,h(x))))))`

```
==> Lógica de Predicado
==> Sistema lógico de derivação
/-----------------------------------------------\
|               Tabela de símbolos              |
|--------+------------+--------+----------------|
| Escopo | Tipo       | Ocorr. | Lexema         |
|--------+------------+--------+----------------|
|      0 | Function   |      1 | g              |
|      0 | Predicate  |      2 | Q              |
|      0 | Predicate  |      1 | P              |
|      0 | Function   |      2 | h              |
|      0 | Variable   |      5 | y              |
|      0 | Variable   |      5 | x              |
\--------+------------+--------+----------------/
```

Perspectivas futuras

- Elaborar uma interface gráfica intuitiva e com resultado imediato, incluindo a possibilidade de gerar a árvore sintática das fórmulas.
- Aplicar o software na práticas de cursos de matemática discreta.
- Ampliar a pesquisa para adaptar os algoritmos de solução para lógica de primeira ordem.
- Pesquisar formas de geração de lema para ampliar a eficiência do tableau com lema e do tableau KE.
- Ampliar a linha de pesquisa para aplicação do software para outras formas de lógica.

Perspectivas futuras

- Elaborar versão com paralelismo.
- Controle de uso de recursos (memória e processador).
- Criar um módulo de teste do software.
- lógica de primeira ordem.
- Pesquisar formas de geração de lema para ampliar a eficiência do tableau com lema e do tableau KE.
- Melhorar o tratamento de erro do compilador, principalmente para apresentar informações mais detalhadas de eventuais erros sintáticos e semânticos.

Fim

https://github.com/fredmbs/logic

Obrigado!