

MAC0239 - INTRODUÇÃO À LÓGICA E VERIFICAÇÃO DE PROGRAMAS
EP – Cálculo Proposicional - Implementação Tableaux

Nome: Luísa Menezes da Costa

Número USP: 12676491

Nome: Natalya Silva Aragão

Número USP: 12543422

Instruções:

Para utilizar o provador de teoremas, basta modificar na função *main()* as variáveis que serão utilizadas e inserir o sequente em fórmula = `ArgumentForm(premissas, conclusão)`.

Conjuntos de sequentes:

```
P,Q = vars('P', 'Q')
formula = ArgumentForm(
    P,      #premises
    conclusion = Q
)
```

```
P,Q = vars('P', 'Q')
formula = ArgumentForm(
    P >> Q, ~Q, #premises
    conclusion = ~P
)
```

```
P,Q = vars('P', 'Q')
formula = ArgumentForm(
    P >> Q, Q, #premises
    conclusion = ~P
)
```

```
P,Q = vars('P', 'Q')
```

```
formula = ArgumentForm(  
    P | Q, P, #premises  
    conclusion = ~Q  
)
```

Exemplo de saída para um sequente não válido:

```
Beta, [True, Proposition( P v Q )]  
[True, Proposition( P )]  
Beta, [False, Proposition( ~Q )]  
[True, Proposition( P )]  
[False, Proposition( Q )]
```

```
[[True, Proposition( P )], [False, Proposition( Q )]]
```

Relatório:

O provador de teoremas implementado recebe como entrada uma proposição que possui sua premissa e conclusão armazenada com os valores *verdadeiro* e *falso*, respectivamente, em um vetor.

Com isso, são obtidas e guardadas, em estruturas diferentes, valores booleanos que representam as posições das expressões que contemplam as regras alfa e beta. Isso possibilita aplicar como estratégia, no primeiro momento, todas as expansões alfa e armazená-las em um ramo.

Logo após, ocorre uma busca, por ordem reversa, pelas expressões que podem sofrer expansão através da regra beta. Para isso, o valor de *beta1* é inserido no ramo, a estrutura que armazena as posições das regras beta do ramo é atualizada e *beta2* é salvo, juntamente com o tamanho do ramo e com o vetor que contém as posições de expressões beta, em uma pilha. Assim, é feita uma busca em profundidade no qual é verificado se o ramo está saturado ou fechado.

Se o ramo estiver saturado, é retornado os valores de seus átomos. Caso ele esteja fechado, o ramo é reconstruído a partir dos valores contidos no topo da pilha. Todas as posições que excedem o tamanho do ramo guardado no topo da pilha são apagadas, assim como as posições de beta, que são modificadas. Ao final, *beta2* é inserido no ramo.

Caso não haja uma contradição no ramo, uma busca é feita para encontrar um ramo sem fórmulas conjugadas. Se a proposição é inválida, a saída do programa apresenta um contra-exemplo. De forma contrária, com uma proposição válida, as expansões são apresentadas linha por linha.

Uma das principais dificuldades enfrentadas foi aplicar as regras alfa e beta de maneira consistente, pois após realizar todas as expansões do tipo alfa e iniciar as expansões do tipo beta, novas regras do tipo alfa poderiam surgir. Além disso, inserir as expressões no ramo seguindo as condições armazenadas na pilha também criou alguns obstáculos.