

Problem C

Algoritmo de Horn em Lógica Proposicional

O objectivo deste problema é a implementação do algoritmo de Horn apresentado na Unidade Curricular de Lógica Computacional. Espera-se que a solução seja concisa, eficaz e elegante.

Problem

A sua tarefa é, dada uma fórmula lógica proposicional transformá-la na sua forma de Horn e aplicar o algoritmo de Horn para verificar se esta é uma contradição ou não.

Considere para esse efeito os tipos de dados OCaml seguintes para representar fórmulas proposicionais:

```
type variavel = string
type formula =
  Implica of formula * formula (* (Formula -> Formula) *)
  | Equivale of formula * formula (* (Formula <-> Formula) *)
  | Ou of formula * formula (* (Formula | Formula) *)
  | E of formula * formula (* (Formula & Formula) *)
  | Nao of formula (* !(Formula) *)
  | Var of variavel (* X1, X2 ... *)
  | Verdade (* TRUE (Top) *)
  | Falso (* FALSE (Bottom) *)
```

Input

A sintaxe da fórmula segue a gramática seguinte:

```
Formula ::= Var | TRUE | FALSE | (Formula -> Formula) | (Formula <-> Formula)
          | (Formula | Formula) | (Formula & Formula) | !(Formula)
```

em que *Var* representa o conjunto das variáveis. Admitimos aqui que tem todas elas a forma X_n em que n é um índice inteiro maior do que 0 e menor ou igual a 200. Admitimos aqui que se uma fórmula contém uma variável X_n então esta contém igualmente $X_1, X_2 \dots X_{(n-1)}$

O processo de leitura de tais formulas é dado no ficheiro `leitura_form.ml` que usa os módulos `Stream` e `Genlex` do OCaml para realizar a análise léxica e sintáctica da entrada.

Assim sendo, o input é organizado da seguinte forma:

Primeira linha: uma string contendo a fórmula f no formato anteriormente apresentado;

Segunda linha: o número n de variáveis contidas na fórmula (o maior n tal que X_n aparece em f).

Output

Uma linha com uma string representando a clausula de Horn que resulta da transformação da formula em input (na ordem exacta na qual a transformação dada na aulas é realizada).

Uma segunda linha com YES se a formula é uma contradição, NO no caso contrário.

Constraints

O índice das variáveis lógicas é um inteiro natural compreendido entre 0 e 200.

Sample Input

```
((X1 | !(X1)) & (X1 & !(X1)))  
1
```

Sample Output

```
((X1 -> X1) & ((TRUE -> X1) & (X1 -> FALSE)))  
YES
```