URI Online Judge | 1670

Lógica Booleana

Contest Local, Universidade de Ulm Alemanha

Timelimit: 1

Proposições são fórmulas lógicas que consistem em símbolos de proposição e operadores conectivos. Eles são definidos recursivamente pelas seguintes regras:

1. Todos os símbolos de proposição (neste problema, caracteres alfabéticos minúsculos, por exemplo, a e z) são proposições.
2. Se P é uma proposição, (!P) é uma proposição, e P é uma subfórmula direta dela.
3. Se P e Q são proposições, (P&Q), (P|Q), (P-->Q), e (P<->Q) são proposições, e P e Q são subfórmulas diretas delas.
4. Nada mais é uma proposição.

As operações !, &, |, -->, e <-> denotam negação, conjunção, disjunção, implicação e equivalência, respectivamente. A proposição P é uma subfórmula de uma proposição R se P=R ou se P é uma subfórmula direta de uma proposição Q e Q é uma subfórmula de R.

Seja P uma proposição e atribui-se valores boleanos (isto é, 0 ou 1) a todos os símbolos de proposição que ocorrem em P. Isto induz um valor booleano para todas as subfórmulas de P, de acordo com a semântica padrão dos operadores lógicos:

Negação	Conjunção	Disjunção	Implicação	Equivalência
!0=1	0&0=0	$0 \mid 0=0$	0-->0=1	0<->0=1
!1=0	0&1=0	$0 \mid 1=1$	0-->1=1	0<->1=0
	1&0=0	$1 \mid 0=1$	1-->0=0	1<->0=0
	1&1=1	$1 \mid 1=1$	1-->1=1	1<->1=1

Dessa forma, o valor de P pode ser calculado. Este valor depende da escolha da atribuição de valores booleanos aos símbolos proposição. Se P contém n símbolos proposição diferentes, existem 2^n atribuições diferentes. Para avaliar todas as tarefas possíveis, podemos utilizar tabelas de verdade.

Uma tabela verdade contém uma linha por atribuição (ou seja, 2^n linhas no total). Cada linha contém os valores de todas as subfórmulas sob a designação escolhida. O valor de uma subfórmula está alinhado com o símbolo da proposição, se a subfórmula é um símbolo proposição, e, de outra forma, com o centro do operador.

Entrada

A entrada contém vários casos de teste, cada um em uma linha separada. Cada caso de teste denota uma proposição e pode conter quantidades arbitrárias de espaços no meio. O arquivo de entrada termina imediatamente após o símbolo de nova linha após o último caso de teste.

Saída

Para cada caso de teste seu programa deve gerar uma tabela verdade para a proposição denotada. Comece a tabela verdade repetindo a linha de entrada. Avalie a proposição (e as suas subfórmulas) para todas as atribuições para as suas variáveis, e use uma linha para cada atribuição. A linha deve ter o mesmo comprimento que a linha de entrada correspondente e deve conter apenas espaços e os caracteres 0 e 1. Imprima uma linha em branco após cada caso de teste.

Deixe os símbolos de proposição (s1, ..., sn) na proposição denotada classificados em ordem alfabética. Então, todas as atribuições de 0 a s1 devem preceder as atribuições de 1 a s1. Dentro de cada um destes blocos de atribuições, todas as atribuições de 0 a s2 devem preceder as atribuições de 1 a s2, e assim por diante.

Exemplo de Entrada

```
((b --> a) <-> ((! a) --> (! b)))
((y & a) - ->(c |c))
```

Exemplo de Saída

```
((b --> a) <-> ((! a) --> (! b)))
  0  1   0   1    1 0   1    1 0
  1  0   0   1    1 0   0    0 1
  0  1   1   1    0 1   1    1 0
  1  1   1   1    0 1   1    0 1

((y & a) - ->(c |c))
  0 0 0   1   0 00
  1 0 0   1   0 00
  0 0 0   1   1 11
  1 0 0   1   1 11
  0 0 1   1   0 00
  1 1 1   0   0 00
  0 0 1   1   1 11
  1 1 1   1   1 11
```