Aula XII

Alexsandro Santos Soares
prof.asoares@gmail.com

Bacharelado em Sistemas de Informação
Faculdade de Computação
Universidade Federal de Uberlândia

11 de maio de 2011

Sumário

Sumário

Objetivos desta aula

- Implementar algumas operações da lógica proposicional.

Interpretação de fórmulas da LP

- Dadas uma fórmula E e uma interpretação I, então a interpretação de E, indicado por $I[E]$, é determinada pelas regras:
  - se $E = true$, então $I[E] = T$;
  - se $E = false$, então $I[E]F$;
  - se E é um símbolo proposicional P, então $I[E] = I[P]$ e $I[P] \in T, F$;
  - se $E = \neg H$, então $I[E] = T$ se $I[H] = F$ e $I[E] = F$ se $I[H] = T$;
  - se $E = (H \vee G)$, então $I[E] = T$ se $I[H] = T$ e/ou $I[G] = T$ e $I[E] = F$ se $I[H] = F$ e $I[G] = F$;
  - se $E = (H \wedge G)$, então $I[E] = T$ se $I[H] = T$ e $I[G] = T$ e $I[E] = F$ se $I[H] = F$ e/ou $I[G] = F$;
  - se $E = (H \rightarrow G)$, então $I[E] = T$ se $I[H] = F$ e/ou $I[G] = T$ e $I[E] = F$ se $I[H] = T$ e $I[G] = F$;
  - se $E = (H \leftrightarrow G)$, então $I[E] = T$ se $I[H] = I[G]$ e $I[E] = F$ se $I[H] \neq I[G]$.

Interpretação de fórmulas da LP em Prolog

- se $E = true$, então $I[E] = T$ e se $E = false$, então $I[E]F$:

```
interpretacao(true,t):- !.
interpretacao(false,f):- !.
```

Interpretação de fórmulas da LP em Prolog

- se $E = \mathit{true}$, então $I[E] = T$ e se $E = \mathit{false}$, então $I[E]F$:

```
interpretacao(true,t):- !.
interpretacao(false,f):- !.
```

- se E é um símbolo proposicional P, então $I[E] = I[P]$ e $I[P] \in T, F$:

```
interpretacao(P,I):-
    atom(P),
    pergunta(P,R), !,
    R = I.
```

Interpretação de fórmulas da LP em Prolog

- se $E = true$, então $I[E] = T$ e se $E = false$, então $I[E]F$:

```
interpretacao(true,t):- !.
interpretacao(false,f):- !.
```

- se E é um símbolo proposicional P, então $I[E] = I[P]$ e $I[P] \in T, F$:

```
interpretacao(P,I):-
   atom(P),
   pergunta(P,R), !,
   R = I.
```

- O predicado pergunta/2 pode ser implementado assim:

```
pergunta(P,I):- resposta(P,I), !.
pergunta(P,I):-
  repeat,
  write('Digite a interpretacao para '), write(P), write(': '),
  read(I),
  (I = t; I = f),
  assert(resposta(P,I)).
```

Interpretação de fórmulas da LP em Prolog – continuação

- se $E = \neg H$, então $I[E] = T$ se $I[H] = F$ e $I[E] = F$ se $I[H] = T$:

```
interpretacao(~H,t):-
   interpretacao(H,f), !.
interpretacao(~H,f):-
   interpretacao(H,t), !.
```

Interpretação de fórmulas da LP em Prolog – continuação

- se $E = \neg H$, então $I[E] = T$ se $I[H] = F$ e $I[E] = F$ se $I[H] = T$:

```
interpretacao(~H,t):-
   interpretacao(H,f), !.
interpretacao(~H,f):-
   interpretacao(H,t), !.
```

- se $E = (H \vee G)$, então $I[E] = T$ se $I[H] = T$ e/ou $I[G] = T$ e $I[E] = F$ se $I[H] = F$ e $I[G] = F$:

```
interpretacao(H v G,t):-
   interpretacao(H,t),
   interpretacao(G,t), !.
interpretacao(H v G,t):-
   interpretacao(H,t), !.
interpretacao(H v G,t):-
   interpretacao(G,t), !.
interpretacao(H v G, f):-
   interpretacao(H,f),
   interpretacao(G,f), !.
```

Interpretação de fórmulas da LP em Prolog – continuação

- se $E = (H \wedge G)$, então $I[E] = T$ se $I[H] = T$ e $I[G] = T$ e $I[E] = F$ se $I[H] = F$ e/ou $I[G] = F$:

```
interpretacao(H & G,t):-
   interpretacao(H,t),
   interpretacao(G,t),
   !.
interpretacao(H & G, f):-
   interpretacao(H,f),
   interpretacao(G,f),
   !.
interpretacao(H & G, f):-
   interpretacao(H,f),
   !.
interpretacao(H & G, f):-
   interpretacao(G,f),
   !.
```

Interpretação de fórmulas da LP em Prolog – continuação

- se $E = (H \to G)$, então $I[E] = T$ se $I[H] = F$ e/ou $I[G] = T$ e $I[E] = F$ se $I[H] = T$ e $I[G] = F$:

```
interpretacao(H => G,t):-
   interpretacao(H,f),
   interpretacao(G,t),
   !.
interpretacao(H => G,t):-
   interpretacao(H,f),
   !.
interpretacao(H => G,t):-
   interpretacao(G,t),
   !.
interpretacao(H => G, f):-
   interpretacao(H,t),
   interpretacao(G,f),
   !.
```

Interpretação de fórmulas da LP em Prolog – continuação

- se $E = (H \leftrightarrow G)$, então $I[E] = T$ se $I[H] = I[G]$ e $I[E] = F$ se $I[H] \neq I[G]$.

```
interpretacao(H <=> G, t):-
   interpretacao(H,I),
   interpretacao(G,I),
   !.
interpretacao(H <=> G, f):-
   interpretacao(H,Ih),
   interpretacao(G,Ig),
   Ih \= Ig.
```

- Como o predicado faz uso de `pergunta/2` faz uso de `assert`, precisamos "limpar" a memória antes de cada sessão:

```
interpreta(F,I):-
     retractall (resposta(_,_)),
    interpretacao(F,I).
```

Interpretação de fórmulas da LP em Prolog – teste

- Vamos realizar alguns testes com o novo predicado:

```
?- interpreta(p v q, I).
Digite a interpretacao (t ou f) para p: t.
Digite a interpretacao (t ou f) para q: t.
I = t.

?- interpreta(p v q => p, I).
Digite a interpretacao (t ou f) para p: t.
I = t.

?- interpreta(p v q => p, I).
Digite a interpretacao (t ou f) para p: f.
Digite a interpretacao (t ou f) para q: t.
I = f.

?- interpreta(p v q => p, I).
Digite a interpretacao (t ou f) para p: f.
Digite a interpretacao (t ou f) para q: f.
I = t.
```

Outra versão da interpretação de fórmulas da LP

- Se permitirmos que somente símbolos verdade apareçam em fórmulas, podemos construir uma outra versão da interpretação.
- Usaremos os predicados `true` para T e `false` para F.

```
~ H    :- not(H).

H v _  :- H, !.
_ v G  :- G.

H & G  :- H, G.

H => G :- not(H) v G.

H <=> G :-
   H => G,
   G => H.
```

- O predicado `false` é sinônimo de `fail`.

Teste da segunda versão da interpretação

- Podemos agora digitar diretamente as fórmulas no interpretador:

```
?- true v true.
true.

?- true v _ => true.
true.

?- false v true => false.
false.

?- false v false => false.
true.

?- true <=> true.
true.

?- true <=> false.
false.

?- false <=> false.
true.
```

Tabela verdade com duas variáveis

- Com o auxílio dos predicados que acabamos de definir, escreva um predicado `tabela/3` que imprime a tabela verdade de uma dada fórmula com duas variáveis.
- Exemplo:

```
?- tabela(X,Y, X & (X v Y)).
true  true  true
true  false true
false true  false
false false false
```

Tabela verdade com duas variáveis – solução

```
tabela(X,Y,F):-
    valor_verdade(X),
    valor_verdade(Y),
    tv(X,Y,F),
    fail .
tabela(_,_,_).

valor_verdade(true).
valor_verdade(false).

tv(X,Y,F):-
    write(X), write(' '), write(Y), write(' '),
    avalia(F),
    nl .

avalia(F):- F, !, write(true).
avalia(_):- write(false).
```

Tabela verdade com qualquer número de variáveis

- Generalize o predicado o problema anterior de tal forma que a fórmula lógica possa conter qualquer número de variáveis.
- Defina `tabela/2` de tal forma que `tabela(Lista,F)` imprima a tabela verdade para a fórmula F que utiliza as variáveis lógicas contidas em `Lista`.
- Exemplo:

```
?- tabela([X,Y,Z], X & (Y v Z) <=> X & Y v X & Z).
true  true  true  true
true  true  false true
true  false true  true
true  false false true
false true  true  true
false true  false true
false false true  true
false false false true
```

Tabela verdade geral – solução

```
tabela(ListaVars,F):- valores_verdade(ListaVars), tv(ListaVars,F), fail.
tabela(_,_).

valores_verdade([]).
valores_verdade([V|Vs]):-
    valor_verdade(V),
    valores_verdade(Vs).

tv(ListaVars,F):-
    escreve_variaveis(ListaVars),
    avalia(F),
    nl.

escreve_variaveis([]).
escreve_variaveis([V|Vs]):-
    write(V), write(' '),
    escreve_variaveis(Vs).
```