Interprete Prolog

L'esecuzione dei programmi Prolog avviene mediante una concatenazione all'indietro (*backward chaining*) in profondità. La computazione inizia dal goal che si vuole derivare, costituito da una congiunzione di formule atomiche (sottogoal): $G_1, G_2, \ldots, G_n$

Una regola $A :- B_1, \ldots, B_m$ è **applicabile** ad un goal atomico G_i se:

1. Le variabili della regola vengono rinominate (*standardizzate separatamente*)
2. A e G_i unificano

L'interprete può essere descritto ricorsivamente con la seguente formulazione *non deterministica*:

interprete(G)
if G vuoto
  then successo
  else sia $G = G_1, G_2, \ldots, G_n$
    sia R l'insieme delle regole applicabili a G_1
    if R è vuoto
      then fallimento
      else scegliere una regola $A :- B_1, \ldots, B_m \in R$
        sia σ il MGU di G_1 e A
        interprete($B_1\sigma, \ldots, B_m\sigma, G_2\sigma, \ldots, G_n\sigma$)

L'esecuzione ha successo, se esiste una computazione che termina con successo.

Questa formulazione è non deterministica perché non specifica quale regola scegliere in R. In realtà, l'interprete del Prolog si comporta in modo deterministico considerando le regole nell'ordine in cui sono scritte nel programma e facendo backtracking ogni volta che la computazione fallisce.

Se l'interprete termina con successo, resituisce anche una sostituzione per le variabili che compaiono nel goal.

ESEMPIO

Si consideri il seguente programma:

R1) $g(c, m)$.
R2) $g(m, l)$.
R3) $a(X, Y) :- g(X, Y)$.
R4) $a(X, Y) :- g(Z, Y), a(X, Z)$.

dove g sta per genitore e a per antenato.

Vediamo la traccia dell'esecuzione del goal $a(X, l)$. Per semplicità supponiamo che vengano rinominate solo le variabili della regola R4.

$a(X, l)$
  R3 $g(X, l)$
    R2 $\{X/m\}$ $\square$
  R4 $\{X1/X, Y1/l\}$ $g(Z1, l), a(X, Z1)$
    R2 $\{Z1/m)\}$ $a(X, m)$
      R3 $g(X, m)$
        R1 $\{X/c\}$ $\square$
      R4 $g(Z2, m), a(X, Z2)$
        R1 $\{Z2/c)\}$ $a(X, c)$
          R3 $g(X, c)$ FALLIMENTO
          R4 $g(Z3, c), a(X, Z3)$ FALLIMENTO

L'indentazione rappresenta il livello di profondità delle chiamate ricorsive dell'interprete. Ogni riga contiene:

- nome della regola applicata
- MGU del primo sottogoal con la testa della regola applicata
- il nuovo goal

Ad esempio, per applicare la regola R4 al goal $a(X, l)$ si rinominano le variabili della regola, ottenendo R4.1: $a(X1, Y1) :- g(Z1, Y1), a(X1, Z1)$. La testa della regola R4.1 unifica con il goal $a(X, l)$ con MGU $\sigma = \{X1/X, Y1/l\}$. Il nuovo goal sarà $g(Z1, l), a(X, Z1)$, ottenuto applicando la sostituzione σ al corpo di R4.1.

Quando si raggiunge il goal vuoto $\square$,l'esecuzione termina con successo. Quando si raggiunge FALLIMENTO, l'inteprete fa backtracking fino al primo punto di scelta e da lì riparte. Se la computazione termina con successo, è possibile forzare il backtracking dando ;.

PROBLEMA. Cosa succede se si inverte l'ordine dei due sottogoal nel corpo delle regola R4?