

Programmazione logica e PROLOG

Esercitazione 1

Sommario

- Programmazione logica
- Base delle conoscenze
- Interrogazioni
- Regole Ricorsive
- Esecuzione dei programmi
- Modello operativo

Programmazione logica

Programmazione **dichiarativa**.

- Programma = descrizione del problema
- Esecuzione = verifica della verità di un'asserzione (goal)

R. Kowalski : Algorithm = Logic + Control.

Applicazioni della Programmazione Logica

PROLOG è il principale di linguaggio di programmazione basato sulla logica (sottoinsieme del *calcolo dei predicati del primo ordine*).

- programmazione logica e basi di dati
- programmazione logica e sistemi esperti

Principio della programmazione logica

Programma logico:

1. definizione del problema da risolvere mediante l'asserzione di **fatti** e **regole** ad esso relativi;
2. interrogazione del sistema automatico che **deduce** la risposta sulla base dei fatti e delle regole note (prova di teoremi).

Sillogismo aristotelico

- *Tutti gli uomini sono mortali*
- *Socrate è un uomo*

si deduce *Socrate è mortale.*

Sillogismo aristotelico in PROLOG

```
mortale(X) :- uomo(X).
```

```
uomo(socrate).
```

La deduzione che Socrate è mortale si ottiene con:

```
?mortale(socrate)
```

La base delle conoscenze

Un programma PROLOG è costituito da un insieme di *clausole*, ossia di *asserzioni condizionate o incondizionate*.

asserzione incondizionata *fatto*:

`padre(daniele,jacopo).`

`ama(enzo,X).`

In PROLOG i nomi dei predicati e delle costanti iniziano con minuscole.

Regole

asserzione condizionata *regola*:

$A :- B, C, \dots, D.$

A è vero se lo sono B, C, ... , D,

- A è detta **conclusione**,
- B, C, ... , D sono le **premesse**
- A, B, C, D sono atomi **atomi**

Se t_1, \dots, t_n sono costanti o variabili e P è un simbolo di predicato n-ario $P(t_1, \dots, t_n)$ è un atomo.

Esempi di regole

`nonno(X,Z) :- padre(X,Y), padre(Y,Z).`

`nonno(X,Z) :- padre(X,Y), madre(Y,Z).`

`figlio(X,Y) :- padre(Y,X).`

`figlio(X,Y) :- madre(Y,X).`

`nonno`, `figlio` si possono considerare come definizioni di **procedure**

Interrogazione del sistema

interrogazione del sistema *goal*

? A,B,C,...,D.

? padre(daniele,jacopo).

SI.

La base delle conoscenze

```
padre(daniele,michela).  
padre(daniele,jacopo).  
padre(eriberto,daniele).  
padre(antonio,eriberto).  
madre(alma,eriberto).  
madre(annamaria,daniele).  
madre(annamaria,marcello).  
madre(annamaria,sandro).
```

```
bella(michela).  
bella(anna).
```

```
fem(michela).
```

Le interrogazioni

? bella(X).

SI michela.

Per avere altre risposte: ;

SI anna

Congiunzioni di goal:

? nonno(eriberto,X), bella(X).

? nonno(X,Z), bella(Z).

Regole ricorsive

```
discendente(X,Y):-figlio(X,Y). % 1
discendente(X,Y):-figlio(Z,Y),discendente(X,Z). % 2
figlio(X,Y):-padre(Y,X). % 3
figlio(X,Y):-madre(Y,X). % 4

? discendente(michela,eriberto).
```

Grafo Orientato

```
/* Un grafo orientato */
```

```
    arco(a,b).  arco(a,c).  arco(b,d).  
    arco(c,d).  arco(d,e).  arco(f,g).
```

```
/* Chiusura transitiva della relazione arco */
```

```
/*
```

```
*   connesso(Nodo1,Nodo2) :- Nodo1 connesso a Nodo2  
*   nel grafo definito dalla relazione arco/2.
```

```
*/
```

```
connesso(Nodo,Nodo).  
connesso(Nodo1,Nodo2) :- arco(Nodo1,NodoInt),  
                        connesso(NodoInt,Nodo2).
```

Modello operativo del PROLOG

- interprete astratto
- unificazione
- ricerca delle soluzioni

L'interprete astratto

Input: un goal G ed un programma P

Output: un'istanza di G , conseguenza logica di P se esiste,
altrimenti NO

begin

$R := G$; R risolvante

finito := false;

dimostra il goal nel risolvante;

if $R = \{ \}$

then ritorna G

else NO

end

Dimostra il goal

while not $R = \{ \}$ and not finito **do**

begin

scegli un goal A dal risolvente

scegli una clausola $A' :- B_1, \dots, B_n$ (ridenominata)

tale che $\theta = \text{unifica}(A, A')$

if scelte esaurite

then finito:=true;

else begin

sostituisci A con B_1, \dots, B_n in R

applica θ ad R e G ;

end

end

L'albero di ricerca

- la radice è il goal iniziale;
- ogni nodo ha tanti successori quante sono le clausole la cui testa unifica con uno dei goal presenti nel nodo. Ogni successore ha un risolvante ottenuto da quello del padre sostituendo al goal selezionato il corpo della clausola corrispondente ed applicando ad esso l'unificatore.

Ogni nodo contiene un risolvante. Se il risolvante è vuoto il nodo è un *nodo di successo*. Un nodo privo di successori, che non sia di successo, è un *nodo di fallimento*.

Ogni nodo di successo rappresenta una soluzione. Se l'albero non può essere espanso e non contiene nodi di successo il goal fallisce.

Le scelte del PROLOG

- la scelta del goal da valutare determina la struttura dell'albero di ricerca;
- la scelta della clausola determina l'ordine dei successori di un nodo.

Nell'interprete PROLOG la scelta del goal da dimostrare viene realizzata esaminando i goal da sinistra a destra e le clausole vengono usate nell'ordine in cui sono scritte. Il risolvete viene gestito tramite una pila. Costruzione dell'albero in profondità.

Varia l'ordine delle regole

```
discendente(X,Y):- figlio(X,Y).
```

```
discendente(X,Y) :- figlio(Z,Y),discendente(X,Z).
```

```
figlio(X,Y) :- madre(Y,X).
```

```
figlio(X,Y):- padre(Y,X).
```

```
? discendente(rita, andrea).
```

Varia l'ordine nel corpo della regola

L'ordine con cui si scrivono le regole e i predicati nel corpo di una regola è significativo in PROLOG.

```
discendente(X,Y):- discendente(X,Z),figlio(Z,Y). %1'  
discendente(X,Y):- figlio(X,Y).  
figlio(X,Y):- padre(Y,X).  
figlio(X,Y):- madre(Y,X).
```

Esercizi

1. Definire la relazione fratello e quindi la relazione cugino
2. Costruire l'albero di ricerca per ?- discendente(michela,eriberto).
e verificare le differenze con ?- discendente2(michela,eriberto).

```
discendente2(X,Y):-figlio(Z,Y),discendente2(X,Z). % 2  
discendente2(X,Y):-figlio(X,Y). % 1
```

3. Costruire l'albero di ricerca ?- stessaGenerazione(jacopo,Y)
e spiegare il comportamento del programma sg:

```
sg (X,X).  
sg (X,Y):-  
    genitore(Z,X),sg(Z,W),genitore(W,Y).
```