

Corso di Linguaggi di Programmazione

Lezione 08

Alberto Ceselli
`alberto.ceselli@unimi.it`

Università degli Studi di Milano

19 Marzo 2013

Definizioni: riassunto

Termine	Significato	Esempio
Programmazione dichiarativa	algoritmi = logica + controllo	
Programmazione logica	algoritmi = calcolo predicativo del primo ordine + motore di inferenza	
Programmazione PROLOG	programma = database di regole + interprete astratto	
Termine (composto)	elemento di una relazione matematica (<i>funttore</i> + <i>argomenti</i>)	madre(anna, silvia).
Proposizione atomica	collezione di termini composti	madre(anna, silvia). madre(anna, alberto). madre(luisa,roberto).
Fatto	proposizione atomica, quantificata universalmente	$\forall X \text{ umano}(X)$.
Query	proposizione atomica, quantificata esistenzialmente	$\exists X \text{ umano}(X)$.
Clausola (di Horn)	<i>regola</i> nella forma $B \leftarrow A_1 \wedge A_2 \dots A_m$	nonna(X, Y) :- madre(X,Z), madre(Z,Y).
Fatti e queries	clausole di Horn <i>headless</i> ($B \leftarrow .$).	umano(alberto).
Istanziare variabili	associare (temporaneamente) valori a variabili logiche	{ X = anna }
Variabile libera	variabile a cui non è associato nessun valore	
Termine ground	termine senza variabili libere	
Unification	rinominare o <i>istanziare</i> variabili, al fine di rendere identici due termini	(vedi prossima slide)
Resolution	algoritmo di inferenza (and membro a membro)	da $Q1 \leftarrow P1$ e $Q2 \leftarrow Q1$ deduco $Q2 \leftarrow P1$

Unification quiz

- $?- a(b,c) \doteq a(X,Y).$
- $?- a(X,c(d,X)) \doteq a(2,c(d,Y)).$
- $?- a(X,Y) \doteq a(b(c,Y),Z).$
- $?- a(X,Y) \doteq b(X,Y).$
- $?- [X] \doteq [Y|Z].$
- $?- [X] \doteq [Y,V|Z].$
- $?- [] \doteq [Y|Z].$
- $?- tree(left, root, Right) \doteq tree(left, root, tree(a, b, tree(c, d, e))).$



Ricorsione e pattern matching

Al pari della programmazione funzionale

- Niente stato, niente cicli né assegnamenti
- ... solo **ricorsione**
- Niente *costrutti di selezione* (if ... then ... else, switch ... case)
- ... solo **pattern matching**

O meglio: costrutti imperativi ci sono (o possono essere *implementati*), ma non vogliamo usarli.

Liste

- **La** struttura dati in Prolog è la **lista**
- Es. notazione `liste.pl`
- `member`
- Domanda: cosa cambia cambiando l'ordine delle clausole in `member`?
- `suffix`, `prefix`

`append`, `delete`, `reverse`, codice, discussione ed esempio di funzionamento dell'interprete su `delete` e variante `select`.



Prolog fornisce supporto per *variabili intere* e relativa aritmetica

- $+$ $-$ $*$ $/$
- $=$ $<$ $>$ $=<$ $>=$
- Operatore $=$
- $2 = 2.$; Yes.
- $2 = 1 + 1.$; No.



Operatori

- Operatore **is** (semanticamente *molto diverso* da assegnamento)
- `2 is 2.`; Yes.
- `2 is 1 + 1.`; Yes.
- tutti i termini devono già essere ground, tranne la variabile a sinistra, che deve essere libera.
- `X is 1 + 1.`; `X = 2.`
- `2 * 1 is 1 + 1.`; No.
- Operatore **==**
- `2 * 1 == 1 + 1.`; Yes.
- tutti i termini devono già essere ground, compresa la variabile a sinistra.



Operatori

- Domanda: `?- X = Y + 1.`
- Domanda: `?- X is Y + 1.`



Prolog ed aritmetica

Esempio: somma.

Esempio: fattoriale.



Predicati come costruttori di tipo

- Prolog: liste eterogenee (`[1,a,pippo,mela(m), pera(p)]`)
- Posso *forzare* dei controlli di tipo?
- Esempio: `checkapples`.
- (nelle prossime lezioni : esempio isomorfismo tra grafi.
`binarytree`, `treemember`, `checked_treemember`,
`isotree`)

Esempio: isomorfismo tra alberi in Prolog

trees.pl

Liste

- `append`, `delete`, `reverse`
- codice, discussione ed esempio di funzionamento dell'interprete su `delete` e variante `select`



Esercizi

- Realizzare un programma prolog `palindrome(X)` che verifichi se una stringa `X` è palindroma o meno.
- Realizzare un programma Prolog che conti il numero di occorrenze di una costante in una lista: `count(EI,Lista,R)` :- *R è il numero di occorrenze di EI in Lista.*
- Realizzare un programma Prolog `double(X,Y)` :- *Y è una lista contenente esattamente due copie di ogni carattere della lista X.* Supponendo che `X` sia una lista di caratteri ordinata lessicograficamente, il programma deve verificare che `Y` sia una lista di caratteri contenente esattamente due copie, memorizzate consecutivamente, di ogni carattere nella lista `X`.

