

UNIVERSITÀ CA' FOSCARI DI VENEZIA
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea Specialistica in Informatica

**Corso di Linguaggi Logici
Approfondimento**

**Nondeterministic and deterministic Robinson's
Algorithm**

Studente: Marco Lionello

Anno Accademico 2007-2008

Introduzione

2

- Due versioni dell'algoritmo
 - Algoritmo non deterministico
 - Sceglie le coppie da unificare in maniera non deterministica
 - Usa rappresentazioni ad alberi
 - Algoritmo deterministico
 - Ottimizza il non determinismo dell'algoritmo precedente andando a prendere le coppie da unificare nell'ordine naturale di scansione testuale: da sinistra a destra
 - E' sufficiente una rappresentazione a stringa

- Definizione induttiva di alberi associati a termini
 - Ad ogni termine associamo un albero
 - L'albero associato con la variabile x ha un solo nodo etichettato con x stesso
 - L'albero associato a $f(t_1, \dots, t_n)$ è ottenuto attaccando l'albero associato a t_1, \dots, t_n in ordine sotto la radice etichettata da f .
 - L'albero associato a una costante c è un nodo etichettato con c stesso

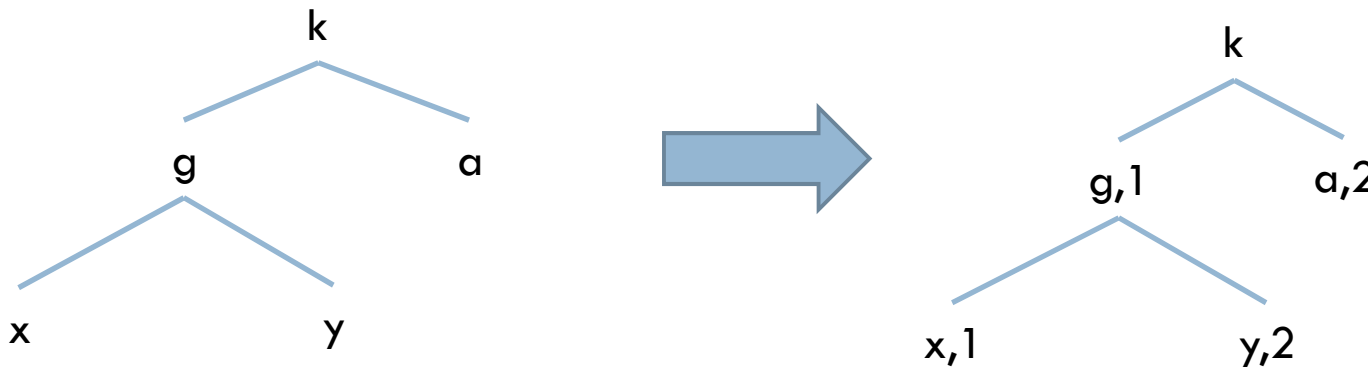
Definizioni (1/4)

3

□ Albero perfezionato

▣ Dato un albero associato ad un termine aggiungendo ad ogni nodo differente dalla radice un numero naturale indicante quale figlio è rispetto a suo padre otteniamo un “*albero perfezionato*”

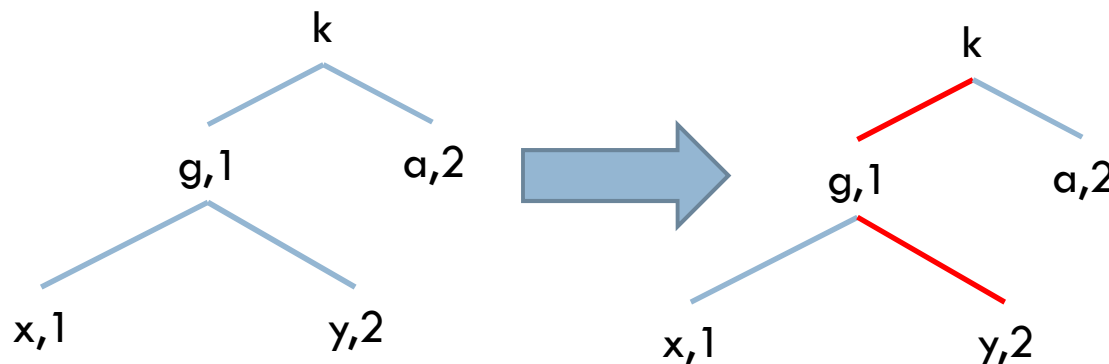
▣ Esempio semplice sul termine: $k(g(x,y),a)$



Definizioni (2/3)

4

- Percorso di accesso (access path)
 - Data un'occorrenza di un termine w in un termine s , il “percorso di accesso di w ” denota la sequenza di nodi dalla radice dell'albero perfezionato associato ad s fino alla radice del sottoalbero perfezionato associato a w .
 - Esempio: $s = k(g(x,y),a)$ $w = y$



L'access path di y è:

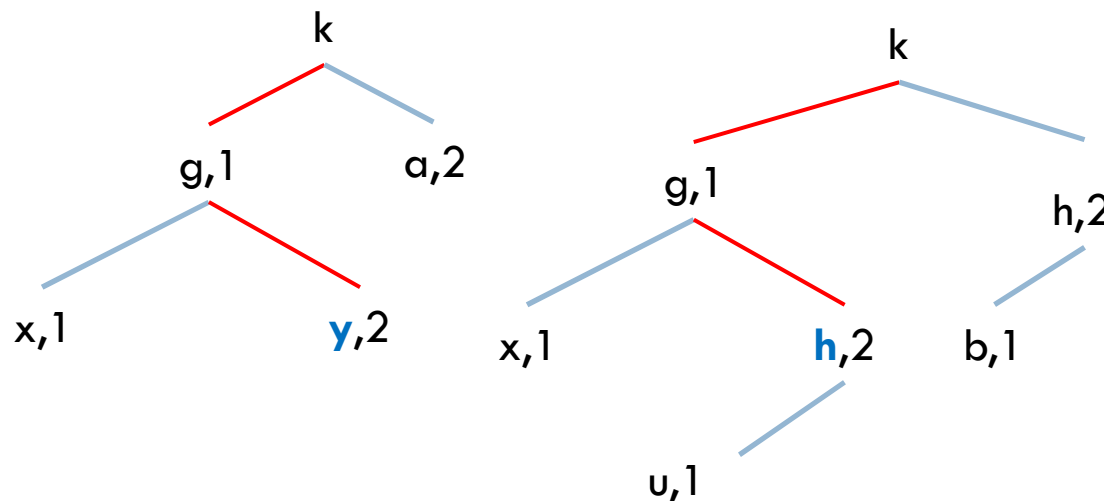
$\langle k \rangle, \langle g,1 \rangle, \langle y,2 \rangle$

Definizioni (3/4)

5

- Coppie in disaccordo (Disagreement pairs)
 - Si consideri una occorrenza di un termine w in un termine s e un'occorrenza di un termine u in un termine t . Una “coppia in disaccordo w,u ” si ha se il loro percorso di accesso differisce solo nell'etichetta dell'ultimo nodo.

- Esempio: y in $k(g(x,y),a)$ e $h(u)$ in $k(g(x,h(u)),h(b))$



Si! Sono coppie in disaccordo
Percorso di accesso di y :
 $\langle k \rangle, \langle g, 1 \rangle, \langle y, 2 \rangle$
Percorso di accesso di $h(u)$:
 $\langle k \rangle, \langle g, 1 \rangle, \langle h, 2 \rangle$
Differiscono solo per l'ultima etichetta

Definizioni (4/4)

6

- Coppia di termini semplici (pairs of simple terms)
 - ▣ Una coppia di termini si dice “*semplice*” se uno di essi è una variabile che non occorre nell’altro termine
 - ▣ Esempio:
 - $y, K(g(x,a),y)$ non è una coppia semplice
 - $a, K(g(x,a),y)$ con a costante non è una coppia semplice
 - $z, K(g(x,a),y)$ è una coppia semplice

- Sostituzione determinata
 - ▣ Data una coppia di termini semplici w, u si dice che la sostituzione $\{x/v\}$ è “*determinata da w, u* ” se $\{x,v\} = \{w,u\}$.
 - Nel caso in cui sia w che u sono variabili, allora due sostituzioni sono determinate da w, u .

Lemma del disaccordo (Disagreement Lemma)

7

Siano w, u una coppia in disaccordo di s e t .

1. Ogni unificatore di s e t è anche un unificatore di w e u
2. Se la coppia w, u è semplice, allora w e u sono unificabili. In realtà, ogni sostituzione determinata da w, u è uno strong mgu di w e u .
3. Se la coppia w, u non è semplice, allora w e u non sono unificabili

Dimostrazione

1. Ogni unificatore di s e t unifica ogni coppia di sottotermini di s e t con lo stesso path di accesso.
2. Se x non occorre in t , allora $x\{x/t\}=t\{x/t\}$. Il resto segue dal binding lemma
3. Due casi:
 1. Se ne w , ne u sono variabili, chiaramente w e u non sono unificabili, perché per definizione di coppie in disaccordo esse cominciano con un simbolo diverso.
 2. Se una di esse (es. w) è una variabile che occorre nell'altro termine (es. u) allora non esiste θ tale che $w\theta=u\theta$ perché $w\theta$ è un sottotermine proprio di $u\theta$.

Algoritmo non deterministico di Robinson

8

Imposta θ a ε

while $s\theta \neq t\theta$ **do**

Scegli non deterministicamente da $s\theta$ e $t\theta$ una coppia in disaccordo w, u ed esegui l'azione associata

1. w, u è una coppia semplice \rightarrow sia γ una sostituzione determinata da w, u . Imposta θ come $\theta\gamma$
2. w, u non è una coppia semplice \rightarrow termina con fallimento

L'algoritmo termina quando $s\theta = t\theta$ o quando avviene un fallimento. Per provare la correttezza dell'algoritmo bisogna provare il seguente teorema:

Teorema: L'algoritmo non deterministico di Robinson termina sempre. Se la coppia originale di termini unifica all'ora l'algoritmo termina e produce uno strong mgu θ di queste coppie altrimenti termina con fallimento

L'algoritmo termina sempre

Dimostrazione(1 / 3)

9

I due casi da analizzare sono 2:

- Sia $\gamma = \{x/v\}$ una sostituzione usata nell'azione 1. Quindi prima del passo 1 x occorre in $s\theta, t\theta$ ma non occorre in $s\theta\gamma, t\theta\gamma$ perché non occorre in v . In aggiunta, tutte le variabili che occorrono in $s\theta\gamma, t\theta\gamma$ occorrono anche in $s\theta, t\theta$. Quindi possiamo dire che l'azione 1 riduce il numero di variabili che occorrono in $s\theta, t\theta$. Questo implica la terminazione
- Se si applica la regola 2 l'algoritmo termina

Invariante dell'algoritmo

10

Uno statement si dice invariante di un ciclo while se mantiene ad ogni istante il corpo del ciclo while.

- Supponiamo che η è un unificatore di s e t . L'invariante dell'algoritmo è:

$$(inv) \eta = \theta\eta$$

- ▣ Dipende dall'inizializzazione di θ . Supponiamo che valga prima dell'azione 1.
- ▣ Per [1] del lemma del disaccordo abbiamo che $w\eta = u\eta$ dove w, u è la coppia considerata.
- ▣ Per [2] del lemma del disaccordo abbiamo che $\eta = \gamma\eta$ dove γ è la sostituzione determinata nell'azione 1.
- ▣ Quindi $\eta = \theta\gamma\eta$. Quindi (inv) è invariante sulla terminazione dell'azione 1.

Se termina con successo allora θ è uno strong mgu di s e t Dimostrazione(2/3)

11

- Supponiamo che l'algoritmo termini con successo
- Sulle condizioni di terminazione del ciclo while si ha che $s\theta = t\theta$
- Quindi θ è un unificatore di s e t
- Ma per ogni unificatore η di s e t l'invariante (inv) è verificato sulla terminazione. Quindi θ è uno strong mgu di s e t

Se termina con fallimento allora s e t non sono unificabili Dimostrazione(3/3)

12

- Supponiamo per assurdo che l'algoritmo termini con fallimento e s e t siano unificabili con unificatore η .
- Siano w, u la coppia in disaccordo che genera il fallimento
- Prima valevano le condizioni dell'invariante e quindi η è un unificatore di $s\theta$ e $t\theta$.
- Grazie al lemma del disaccordo (punto 1 e 3) applicato alle coppia in disaccordo w, u dei termini $s\theta$ e $t\theta$ la coppia w, u è semplice e quindi porta ad un assurdo

Algoritmo deterministico di Robinson

13

Imposta θ a ε ;

Imposta i puntatori al primo simbolo di $s\theta$ e $t\theta$;

while $s\theta \neq t\theta$ **do**

Avanza simultaneamente i puntatori a destra fino a che si trova una coppia di simboli diversi in $s\theta$ e $t\theta$;

Determina la coppia di termini w e u i cui simboli principali sono quelli identificati e esegui la operazione associata:

1. w, u è una coppia semplice \rightarrow sia γ una sostituzione determinata da w, u .
Imposta θ come $\theta\gamma$
2. w, u non è una coppia semplice \rightarrow termina con fallimento

- L'algoritmo termina quando $s\theta = t\theta$ altrimenti fallisce
- La dimostrazione correttezza è la stessa dell'algoritmo non deterministico

Esempi (1/2)

14

- $k(\underline{z}, f(x, b, z))$ e $k(\underline{h}(x), f(g(a), y, z))$
 - Scansionando i termini la prima coppia in disaccordo è $z, h(x)$. La coppia determina la sostituzione $\{z/h(x)\}$
 - Otteniamo:
$$k(h(x), f(\underline{x}, b, h(x))) \text{ e } k(h(x), f(g(a), y, h(x)))$$
 - Scansionando i termini la prima coppia in disaccordo è $x, g(a)$. La coppia determina la sostituzione $\{x/g(a)\}$
 - Otteniamo:
$$k(h(g(a)), f(g(a), \underline{b}, h(g(a)))) \text{ e } k(h(g(a)), f(g(a), y, h(g(a))))$$
 - Scansionando i termini la prima coppia in disaccordo è b, y . La coppia determina la sostituzione $\{y/b\}$
 - Otteniamo:
$$k(h(g(a)), f(g(a), b, h(g(a)))) \text{ e } k(h(g(a)), f(g(a), b, h(g(a))))$$
- Componendo le sostituzioni trovate abbiamo:
 - $\{z/h(x)\}\{x/g(a)\}\{y/b\} = \{z/h(g(a)), x/g(a), y/b\}$ che è uno strong mgu

Esempi (2/2)

15

□ $K(\underline{z}, f(x, b, z))$ e $k(\underline{h}(x), f(g(z), y, z))$

- Scansionando i termini la prima coppia in disaccordo è $z, h(x)$. La coppia determina la sostituzione $\{z/h(x)\}$

□ Otteniamo:

$k(h(x), f(\underline{x}, b, h(x)))$ e $k(h(x), f(\underline{g}(h(x)), y, h(x)))$

- Scansionando i termini la prima coppia in disaccordo è $x, g(h(x))$. La coppia determina non è semplice e quindi per il teorema i due termini non unificano

Bibliografia

16

- K.R. Apt. *From Logic Programming to Prolog*. Prentice Hall International, 1997.
- Leon Sterling, Ehud Shapiro: *The Art of Prolog*. SECOND EDITION, The MIT Press 1994