Algoritmo di unificazione sintattica

21.11.2011

Problema dell'UNIFICAZIONE: stabilire se si possano rendere

sintatticamente uguali

due termini '*istanziando*' opportunamente le loro variabili, cioè sostituendole in modo '*uniforme*' con dei termini.

ISTANZIARE una variabile significa sostituirla con un termine

- se applicata a un termine, quest'operazione produce un termine ;
- se applicata a una formula, quest'operazione produce una formula .

UNIFORME è una sostituzione che tratta tutte le occorrenze di ciascuna variabile allo stesso modo.
(**Osservazione:** Tale era anche la sostituzione di enunciati proposizionali alle lettere di una tautologia nel calcolo proposizionale).

Esempi:

- $X + 1 + Y$ *è* unificabile con $f(Z) + Y + Y$
  - una sostituzione di *massima generalità* è

$$\begin{array}{lcl} X & \mapsto & f(Z) \\ Y & \mapsto & 1 \end{array}$$

  - una sostituzione di *minima generalità* è

$$\begin{array}{lcl} X & \mapsto & f(0) \\ Y & \mapsto & 1 \\ Z & \mapsto & 0 \end{array}$$

- Invece $X + 1 + Y$ *non è* unificabile con $f(Z) + Y + 0$
- *Neppure* $X + 1 + Y$ è unificabile con $f(Z) + Y + a$
- $X + 1$ *non è* unificabile con 1
  (La sostituzione $X \mapsto 0$ sarebbe un'unificazione '*semantica*')

Problemi:

- È possibile unificare

$$h(X_0, X_1 \ldots, X_n) \quad \text{con} \quad h(g(X_1, X_1), g(X_2, X_2) \ldots, g(X_{n+1}, X_{n+1}))$$

$$?$$

- È unificabile

$$h(X_{n+1}, X_1 \ldots, X_n) \quad \text{con} \quad h(g(X_1, X_1), g(X_2, X_2) \ldots, g(X_{n+1}, X_{n+1}))$$

$$?$$

- È unificabile

$$f(0) \quad \text{con} \quad 0$$

$$?$$

- È unificabile

$$f(X) \quad \text{con} \quad X$$

$$?$$

Ci può rispondere Prolog ?

Risposte: sí, no, no, no

In Prolog, l'*unificazione* è un meccanismo di basso livello: può essere equiparato al meccanismo di passaggio di parametri dei linguaggi procedurali di programmazione (ma ha molti altri compiti !).

Nei linguaggi funzionali (eredi di LISP quali ML o Haskell), il *matching* ha un ruolo simile a quello che l'unificazione ha in Prolog; ma il matching mantiene la sostanziale *asimmetria* della distinzione fra parametri formali e parametri attuali.

In matematica, il problema dell'*unificazione semantica* domina la scena—sia pure sotto altro nome:

- risolubile in casi favorevoli ;
- irresolubile in casi importanti (vi ricordate il X problema di Hilbert ?)

Unificheremo *termini* cercando ℓ'unificatore di massima generalità
(è *quasi* unico, come si vede unificando tra loro due variabili)

Con ciò implicitamente risolveremo un problema storico della deduzione automatica: quello di individuare letterali '*complementarmente unificabili*'.

```
ama(X, coniuge(X))          :-  sposato(X).
sposato(mariposa).
¬ ama(X, coniuge(mariposa))
```

Importante! In esempi più intricati di questo ricordate, prima di tentare unificazioni, di ridenominare le variabili di una clausola ogni volta che la usate !

Interviene qui una parente quantificata del Modus Ponens, il principio di risoluzione (1965) di John Alan Robinson.

Algoritmo di Robinson

Se dovete unificare

- una variabile con se stessa:

  nessun problema

- una variabile con un'altra:

  sostituite una all'altra

- una variabile con una costante (o viceversa):

  sostituite la costante alla variabile

- una variabile con un termine composto (o vicev.):

  controllate che la variabile *non* figuri nel termine

- due termini (costanti o composti)

  $$h(t_1, \ldots, t_n) \;=\; k(s_1, \ldots, s_m) \quad :$$

  accertatevi che il simbolo portante sia lo stesso, i.e. che $n = m$ ed $h \equiv k$; quindi, per $i = 1, \ldots, n$

  - trovate l'unificatore μ di massima generalità fra t_i ed s_i e applicatelo in permanenza a $t_{i+1}, \ldots, t_n$ ed a $s_{i+1}, \ldots, s_n$

Algoritmi efficienti di unificazione

- Alberto Martelli, Ugo Montanari (Pisa, Università e CNR)
- Mark N. Wegman, Mike Paterson (IBM)

proposero indipendentemente ca. 1976 algoritmi di unificazione di costo lineare o quasi, che però richiedono che i termini vengano rappresentati come sistemi di equazioni e le sostituzioni come '*binding*'

Unificazione per termini razionali infiniti

Uno dei piú seri problemi, nell'implementazione dell'unificazione, ha a che vedere con la necessità di effettuare l'*occurs-check*, ossi nel verificare che una variabile non compaia mai nel termine a cui viene legate.

Soluzione prospettata da vari studiosi, fra cui Alain Colmerauer (uno dei co-fondatori di Prolog): introdurre nel linguaggio i termini infiniti razionali.

Chi ha proposto il primo algoritmo di unificazione sintattica ?

Forse l'inventore non è stato davvero Robinson, ma —almeno tre o quattro anni prima—

. . . Dag Prawitz, oppure Martin Davis . . .

Gérard Huet si accorse nel 1977 che l'inventore era stato (ca. 1930) Jacques Herbrand !

O forse l'inventore è stato lui ?