Linguaggio dei Termini

Dati 2 insieme: A, V non vuoti e disgiunti:

- A è l'insieme dei simboli:
- V è l'insieme delle variabili:
- T è l'insieme dei temini construito con l'insieme A e V secondo le regole:
  - Una variabile $v \in V$ è un termine;
  - Un atomo $a \in A$ è un termine;
  - Se $f \in A$ allora $f(t_1, t_2, \dots t_n)$ è un termine;
  - nient'altro è un termine.

Il linguaggio dei termini T è il linguaggio che si appoggia sull'alfabeto improprio $A \cup V$.

Un termine si dice non strutturato se è composto da un atomo o da una variabile, altrimenti viene detto strutturato.

L'atomo più a sinistra viene detto testa (non è una variabile).

Il numero di argomenti di un termine strutturato viene detto arità.

I termini strutturati possono essere usati per descrivere alberi.

Dato un termine $t \in T$, $vars(t)$ restituisce l'insieme delle variabili utilizzate per costruire t.

- Se $vars(t) \neq \emptyset$ allora t viene detto ground (albero senza simboli di variabile).
- Se $A = \emptyset$ e $V = \emptyset$ allora il risultato è un insieme detto Universo di Herbrand.

2 termini t_1 e t_2 si dicono sintatticamente equivalenti se sono lo stesso elemento di t.

Per semplificare la lettura e la scrittura è necessario introdurre alcune varianti sintattiche speciali:

1. Un atomo indicato con il simbolo **.** che viene usato per formare termini strutturati di 2 elementi.
2. Un atomo indicato con il simbolo [] che viene usato per indicare termini non strutturati.

Utilizzando la sintassi delle liste:

1. la lista [] è un termine chiamato lista vuota.
2. la lista [h | r] è un termine .(h,r) dove **h** rappresenta la testa della lista e **r** il resto della lista.
3. la lista $[t_1, t_2, \ldots, t_n]$ è il termine $.(t_1, .(t_2, ..., .(t_n, [])...))$.
4. la lista $[t_1, t_2, \ldots, t_n$ | r] è il termine $.(t_1, .(t_2, ..., .(t_n, r)...))$. dove r è detto resto della lista.

Oltre alle liste vengono introdotti degli operatori (unari o binari).

Ogni operatore viene descritto mediante 3 proprietà:

1. L'atomo da usare per indicare l'operatore: +, -
2. L'indice di precedenza dell'operatore.
   a. fx, fy operatori unari prefissi;
   b. xfx, xfy operatori binari infissi;
   c. xf, yf operatori unari postfissi;
3. Il tipo di operatore.

Unificazione di Termini

Una sostituzione è un insieme finito, eventualmente vuoto della forma:

$$\{t_1/x_1, t_2/x_2, ..., t_n/x_n\}$$

dove:

1. $x_1, x_2, \ldots, x_n$ sono variabili;

2. $t_1, t_2, \dots, t_n$ sono termini;

Se t è un termine e θ è una sostituzione, allora $t\theta$ è il termine che si ottiene sostituendo simultaneamente tutte le variabili nelle parti destre con i relativi termini di sinistra.

Scrivere $t\theta$ significa applicare θ a t.

Una sostituzione θ si dice unificatore se vale la seguente catena diuguaglianze:

$$t_1\theta = t_2\theta = ... = t_n\theta$$

Un insieme di termini si dice unificabile se esiste almeno un unificatore per l'insieme. Un insieme di termini può ammettere più unificatori.

Se un insieme di termini S è unificabile, allora ammette un unificatore più generale (MGU).

Il MGU di un insieme di termini S è definito come un unificatore $mgu(S)$ tale che qualsiasi sia un unificatore θ di S esista la sostituzione σ tale che $\sigma = mgu(S)$ o σ.

Dato l'insieme dei termini T costituito dagli insiemi degli atomi A e delle variabili V, un problema di unificazione (sintattica) è un insieme del tipo:

$$\{l_1 = r_1, l_2 = r_2, ..., l_n = r_n\}$$

dove ogni elemento del problema viene detto equazione.

L'algoritmo di Martinelli e Montanari permette di stabilire se un problema di unificazione è risolubile e nel caso trovare una soluzione. Esso parte dal problema di unificazione S e termina producendo uno dei possibili risultati:

1. il simbolo $\perp$ se il problema di unificazione non è risolubile.
2. un problema di unificazione equivalente a S ma nella forma $\{x_1 = t_1, x_2 = t_2, ..., x_n = t_n\}$ con solo variabili nelle parti sx delle equazioni da cui è possibile costruire immediatamente una soluzione del problema $\{t_1/x_1, t_2/x_2, ..., t_n/x_n\}$.

L'algoritmo può essere descritto mediante la funzione non deterministica unify che ha come unico argomento un problema di unificazione S. Essa si suddivide in più casistiche:

- caso decompose;
- caso eliminate;
- caso swap;
- caso delete;
- caso conflict;
- caso check;

Il caso check aumenta il carico computazionale, quindi delle volte si preferisce eliminare questo caso, ottenendo una procedura di unificazione senza occurs check.