

ALGORITMO DI UNIFICAZIONE E RISOLUZIONE PREDICATIVA

Consideriamo la formula

$$F = \forall x \forall y r(fx, y) \wedge \forall u \exists v \neg r(u, v).$$

La formula F è chiaramente insoddisfacibile; la sua forma clausale è data da un insieme S con due clausole $C_1 = \{r(fx, y)\}$ e $C_2 = \{\neg r(u, gu)\}$. Nessun passo di risoluzione proposizionale, però, ci porta alla clausola vuota.

Consideriamo ora una *sostituzione* σ che trasforma variabili in termini. Possiamo applicare la sostituzione ad letterale o ad una clausola sostituendo le variabili x del letterale o della clausola con i rispettivi termini $\sigma(x)$. Il letterale e la clausola così ottenuti si denotano con $\ell\sigma$ e $C\sigma$.

Ad esempio, se σ è la sostituzione tale che $\sigma(y) = gfx$, $\sigma(u) = fx$ e σ lascia ferme le altre variabili, si ha:

$$C_1\sigma = \{r(fx, gfx)\}, \quad C_2\sigma = \{\neg r(fx, gfx)\}$$

e dall'insieme $\{C_1\sigma, C_2\sigma\}$ con un singolo passo di risoluzione proposizionale si ottiene la clausola vuota. Inoltre, come vedremo, $C \models C\sigma$, quindi il procedimento sopra descritto ci permette di stabilire che l'insieme $S = \{C_1, C_2\}$ è insoddisfacibile.

Questo esempio si generalizza: un passo di risoluzione predicativo consiste nella ricerca di una sostituzione σ che *renda possibile* un passo proposizionale, cioè tale che esistano $\ell_1, \dots, \ell_k$ in C_1 e $\ell'_1, \dots, \ell'_h$ in C_2 con $\ell_1\sigma = \ell_k\sigma = \ell'_1\sigma = \ell'_h\sigma = p(\bar{t})$. Un tale σ si dice *unificatore* dell'insieme $\{\ell_1, \dots, \ell_k, \ell'_1, \dots, \ell'_h\}$.

Ex:

$$F = \forall x \forall y (q(x) \vee q(fy) \vee \neg r(x, y)) \wedge \forall u \forall v (\neg q(u) \vee \neg q(fv) \vee r(a, a))$$

$$C_1 = \{q(x), q(fy), \neg r(x, y)\}, \quad C_2 = \{\neg q(u), \neg q(fv), r(a, a)\}.$$

Se $\sigma(x) = fy$, $\sigma(u) = fy$, $\sigma(v) = y$, otteniamo

$$C_1\sigma = \{q(fy), \neg r(fy, y)\}, \quad C_2\sigma = \{\neg q(fy), r(a, a)\},$$

da cui, per risoluzione proposizionale si può ottenere la clausola

$$C_3 = \{\neg r(fy, y), r(a, a)\}.$$

Quindi,

$$C_1, C_2 \models C_3.$$

Dobbiamo prima però risolvere il problema: dati t_1, t_2 termini, possiamo trovare un unificatore?

1. ALGORITMO DI UNIFICAZIONE E REGOLA DI RISOLUZIONE PREDICATIVA

1.1. Sostituzioni e sistemi di Herbrand.

Definizione 1.1. Una sostituzione è una funzione σ dall'insieme delle variabili del linguaggio ai termini tale che l'insieme $\{x \in Var : x \neq \sigma(x)\}$ è finito.

L'insieme $\{x \in Var : x \neq \sigma(x)\}$ viene detto *dominio* della sostituzione σ . L'immagine $\sigma(x)$ di una variabile x tramite una sostituzione σ viene anche denotata con $x\sigma$. Una sostituzione σ con dominio $\{x_1, \dots, x_n\}$ e $\sigma(x_i) = t_i$ per $i = 1, \dots, n$ viene anche denotata da $\{x_1|t_1, \dots, x_n|t_n\}$. Ad esempio indichiamo con $\{x|fa, y|w\}$ la sostituzione che manda x in fa , y in w e lascia ferme tutte le variabili diverse da x e y .

Una sostituzione viene estesa in maniera naturale all'insieme dei termini del linguaggio ponendo

$$c\sigma = c, \quad f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma).$$

Date due sostituzioni σ e τ indichiamo con $\sigma\tau$ la funzione che ad ogni variabile x associa il termine $(x\sigma)\tau$. Verificare per esercizio che tale funzione è, in effetti, una sostituzione, cioè che l'insieme $\{x : x \neq x\sigma\tau\}$ è finito.

Una sostituzione η si dice un'istanza di una sostituzione σ se esiste una sostituzione τ tale che $\sigma\tau = \eta$. Ad esempio, se

$$\sigma = \{x|fu, y|g(h(gv, u)), z|w\}, \quad \eta = \{x|fga, y|g(h(gfa, ga)), z|w\},$$

la sostituzione η è un'istanza della sostituzione σ perché se $\tau = \{u|ga, v|fa\}$ si ha $\sigma\tau = \eta$.

Definizione 1.2. Un sistema di Herbrand è un insieme finito di coppie di termini

$$(t_1, s_1), \dots, (t_n, s_n).$$

Una soluzione (detta anche unificatore) di un sistema di Herbrand $(t_1, s_1), \dots, (t_n, s_n)$ è una sostituzione σ tale che $t_i\sigma = s_i\sigma$, per ogni $i = 1, \dots, n$. Un unificatore di un sistema di Herbrand si dice di massima generalità (m.g.u.) se ogni altro unificatore η del sistema è istanza di σ . Due sistemi di Herbrand si dicono equivalenti se hanno le stesse soluzioni.

Un sistema di Herbrand viene anche denotato con

$$\begin{cases} t_1 = s_1; \\ \cdot \\ \cdot \\ \cdot \\ t_n = s_n \end{cases}$$

Lemma 1.1. Se η unifica il sistema

$$\begin{cases} x_1 = r_1; \\ \cdot \\ \cdot \\ \cdot \\ x_n = r_n \end{cases}$$

dove le variabili x_i sono a due a due distinte, allora la sostituzione $\{x_1/r_1, \dots, x_n/r_n\}\eta$ coincide con η .

Dimostrazione. Basta verificare che le sostituzioni $\{x_1/r_1, \dots, x_n/r_n\}\eta$ e η hanno lo stesso valore su ogni variabile y . Se $y = x_i$ per un qualche i abbiamo

$$x_i\{x_1/r_1, \dots, x_n/r_n\}\eta = r_i\eta = x_i\eta.$$

Se $y \neq x_i$ per ogni i si ha

$$y\{x_1/r_1, \dots, x_n/r_n\}\eta = y\eta.$$

□

1.2. Algoritmo di unificazione. Dato un sistema di Herbrand $E = \{t_1, s_1), \dots, (t_n, s_n)\}$ vogliamo decidere in maniera algoritmica se il sistema E ammette almeno una soluzione, e, in tal caso, trovare tutte le possibili soluzioni di E .

Notiamo per prima cosa che, per un tipo particolare di sistemi di Herbrand, i cosiddetti sistemi *risolti*, il problema ha una semplice soluzione.

Definizione 1.3. *Un sistema di Herbrand del tipo*

$$\begin{cases} x_1 = r_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n = r_n \end{cases}$$

tale che se $x_i \neq x_j$ per $i \neq j$ e le variabili x_i non compaiono a destra delle equazioni del sistema, si dice risolto

Lemma 1.2. *Dato un sistema risolto*

$$\begin{cases} x_1 = r_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n = r_n \end{cases}$$

la sostituzione $\sigma = \{x_1/r_1, \dots, x_n/r_n\}$ è un m.g.u. del sistema.

Dimostrazione. La sostituzione $\sigma = \{x_1/r_1, \dots, x_n/r_n\}$ unifica il sistema perché $x_i\sigma = r_i$ e $r_i\sigma = r_i$ (infatti r_i non contiene variabili del dominio di σ). Quindi σ è un unificatore del sistema.

Mostriamo ora che σ è un m.g.u. del sistema: se η unifica il sistema, allora, applicando il lemma ?? otteniamo

$$\sigma\eta = \{x_1/r_1, \dots, x_n/r_n\}\eta = \eta.$$

Quindi η si ottiene come istanza di σ (tramite lo stesso η). □

L'algoritmo di unificazione prende in input un sistema di Herbrand e cerca di trasformarlo in un sistema equivalente e risolto. Un passo dell'algoritmo consiste nel scegliere una coppia (un'equazione) del sistema di Herbrand e operare a seconda della forma dell'equazione nel modo seguente:

- (1) se è del tipo $c = c$, $x = x$, si elimina l'equazione e si prosegue con il sistema così ottenuto;
- (2) se è del tipo $f(t_1, \dots, t_n) = g(r_1, \dots, r_k)$, con f diverso da g oppure $c = d$ con c diverso da d oppure $c = f(t_1, \dots, t_n)$ oppure $f(t_1, \dots, t_n) = c$ per c, d costanti, l'algoritmo si ferma con la risposta *non unificabile*;

- (3) se è del tipo $x = t$ con x che compare in t e t diversa da una variabile, l'algoritmo si ferma con la risposta *non unificabile*;
- (4) se è del tipo $f(t_1, \dots, t_n) = f(s_1, \dots, s_n)$ si sostituisce l'equazione con le n equazioni

$$\begin{cases} t_1 = s_1 \\ \cdot \\ \cdot \\ \cdot \\ t_n = s_n, \end{cases}$$

si prosegue con il sistema così ottenuto;

- (5) se è del tipo $t = x$ con t diverso da una variabile, si sostituisce l'equazione con $x = t$;
- (6) se è del tipo $x = t$ con x che non compare in t , si sostituisce ogni altra equazione $r = s$ del sistema con l'equazione $r\{x|t\} = s\{x|t\}$;

l'algoritmo si ferma con risposta *unificabile* quando non è possibile fare più alcun passo, mentre si ferma con risposta *non unificabile* quando si trova un'equazione non unificabile (casi 2,3 della lista)

Notiamo che l'algoritmo si ferma con risposta *unificabile* se e solo se il sistema a cui si è arrivati è un sistema risolto.

Lemma 1.3. *Correttezza e Completezza dell'algoritmo di unificazione: l'algoritmo di unificazione si ferma dopo un numero finito di passi partendo da un qualsiasi sistema di Herbrand. Il sistema in input è unificabile se e solo se l'algoritmo raggiunge un sistema risolto, e, in questo caso il sistema di partenza ha lo stesso m.g.u. del sistema finale,*

Dimostrazione. Ci basta far vedere che l'algoritmo termina sempre dopo un numero finito di passi e che ogni passo dell'algoritmo se non si ferma con risposta *non unificabile* trasforma il sistema in un sistema equivalente. Notiamo inoltre che nei passi 2,3 il sistema raggiunto è effettivamente non unificabile.

Terminazione. Sia associata ad ogni sistema di Herbrand E la terna (a_0, a_1, a_2) , dove:

a_0 è il numero di variabili non risolte del sistema (una variabile si dice non risolta se appare a destra di un'equazione o in più di un'equazione o nella forma $x = x$).

a_1 è il numero di occorrenze di simboli funzionali o costanti a sinistra delle equazioni;

a_2 è il numero di equazioni del tipo $x = x$.

Quindi ad ogni sistema è associata una terna di numeri naturali. Queste terne possono essere ordinate tramite l'ordine lessicografico così definito:

$$(a, b, c) < (a', b', c') \Leftrightarrow (a < a') \vee (a = a' \wedge b < b') \vee (a = a' \wedge b = b' \wedge c < c').$$

In questo ordine non esistono catene infinite discendenti. Per dimostrare la terminazione dell'algoritmo di risoluzione basta allora notare che ad ogni passo dell'algoritmo la terna associata diminuisce nell'ordine lessicografico.

Correttezza Ogni passo trasforma il sistema in uno equivalente. Questo è ovvio per passi di tipo 1, 2, 3, 4, 5. Per quanto riguarda i passi di tipo 6, notiamo che, grazie al Lemma ??, un sistema

del tipo

$$\begin{cases} x = t; \\ t_1 = s_1 \\ \cdot \\ \cdot \\ \cdot \\ t_n = s_n \end{cases}$$

ha le stesse soluzioni del sistema

$$\begin{cases} x = t; \\ t_1\{x/t\} = s_1\{x/t\} \\ \cdot \\ \cdot \\ \cdot \\ t_n\{x/t\} = s_n\{x/t\}. \end{cases}$$

□

1.2.1. *Esercizi.* Applicare l'algoritmo di unificazione ai seguenti sistemi. Quando possibile, trovare un m.g.u. del sistema.

$$\begin{cases} h(x, fy) = v \\ w = fh(a, ffx) \\ h(a, ffu) = v \\ fh(a, ffa) = w. \end{cases}$$

$$\begin{cases} f(x, f(hx, a)) = f(y, z) \\ z = f(y, a) \end{cases}$$

$$\begin{cases} g(y) = x \\ f(x, hx, y) = f(gz, w, z) \end{cases}$$

2. RISOLUZIONE

Come vedremo, l'algoritmo di unificazione viene usato all'interno della regola di risoluzione predicativa: dato un letterale $p(t_1, \dots, t_n)$ e una sostituzione σ si definisce

$$\sigma(p(t_1, \dots, t_n)) := p(\sigma(t_1), \dots, \sigma(t_n)), \quad \sigma(\neg p(t_1, \dots, t_n)) := \neg p(\sigma(t_1), \dots, \sigma(t_n)).$$

Un insieme di letterali $\{\ell_1, \dots, \ell_k\}$ si dice *unificabile* se esiste una sostituzione σ tale che $\sigma(\ell_1) = \dots = \sigma(\ell_k)$.

Data una clausola $C = \{\ell_1, \dots, \ell_k\}$ e una sostituzione σ denotiamo con $C\sigma$ la clausola

$$C\sigma = \{\sigma(\ell_1), \dots, \sigma(\ell_k)\}.$$

Si noti che la formula che corrisponde alla clausola C è $\forall \bar{x}(\ell_1 \vee \dots \vee \ell_k)$, dove $\bar{x}$ contiene tutte le variabili libere di $(\sigma(\ell_1), \dots, \sigma(\ell_k))$, mentre la formula che corrisponde alla clausola $C\sigma$ è $\forall \bar{z}(\sigma(\ell_1) \vee \dots \vee \sigma(\ell_k))$.

$\dots \vee \sigma(\ell_k))$, dove $\bar{z}$ contiene tutte le variabili libere di $(\sigma(\ell_1) \vee \dots \vee \sigma(\ell_k))$. Usando il Lemma di sostituzione, si dimostra (vedi gli esercizi alla fine del paragrafo) che vale

$$C \models C\sigma.$$

2.1. Regola di Risoluzione Predicativa. Siano

$$C_1 = \{\ell_1, \dots, \ell_k\} \cup G, \quad C_2 = \{\ell'_1, \dots, \ell'_h\} \cup H,$$

due clausole predicative **a variabili separate** tali che l'insieme

$$\{\ell_1, \dots, \ell_k, \bar{\ell}'_1, \dots, \bar{\ell}'_h\}$$

sia unificabile. Se σ è un m.g.u di tale insieme, allora il risolvente di C_1, C_2 rispetto a σ è la clausola

$$Ris_\sigma(C_1, C_2) = G\sigma \cup H\sigma.$$

Se le clausole C_1, C_2 non sono a variabili separate, prima di applicare la regola di risoluzione si avrà l'accortezza di cambiare il nome delle variabili di una delle due clausole, in modo da ottenere clausole a variabili separate.

Esempio: Consideriamo le clausole

$$C_1 = \{p(x, fx), p(x, z), p(y, y), \neg q(y, x)\}, \quad C_2 = \{\neg p(fh(z, a), fu), \neg p(u, v), r(u), \neg q(u, v)\}.$$

Prima di cercare di risolvere tali clausole, le rinominiamo per ottenere clausole a variabili separate (che chiamiamo ancora C_1, C_2)

$$C_1 = \{p(x, fx), p(x, s), p(y, y), \neg q(y, x)\}, \quad C_2 = \{\neg p(fh(z, a), fu), \neg p(u, v), r(u), \neg q(u, v)\}.$$

Consideriamo i letterali $p(x, fx), p(x, s)$ in C_1 e i letterali $\neg p(fh(z, a), fu), \neg p(u, v)$ in C_2 . L'insieme

$$\{p(x, fx), p(x, s), p(fh(z, a), fu), p(u, v)\}$$

è unificabile. Per accorgersene, impostiamo un sistema di Herbrand (in cui abbiamo saltato certe equazioni cancellabili quali $x = x$ o ripetute):

$$\begin{cases} fx = s \\ x = fh(z, a) \\ fx = fu \\ x = u \\ fx = v \end{cases}$$

Utilizzando l'algoritmo di unificazione si ottiene in un numero finito di passi il sistema risolto

$$\begin{cases} s = ffh(z, a) \\ u = fh(z, a) \\ x = fh(z, a) \\ v = ffh(z, a) \end{cases}$$

che fornisce il seguente m.g.u.

$$\sigma = \{s/fh(z, a), u/fh(z, a), x/fh(z, a), v = ffh(z, a)\}.$$

Otteniamo quindi per risoluzione predicativa la clausola

$$Ris_\sigma(C_1, C_2) = \{p(y, y), \neg q(x, y)\}\sigma \cup \{r(u), \neg q(u, v)\}\sigma =$$

$$= \{p(y, y), \neg q(fh(z, a), y), r(fh(z, a)), \neg q(fh(z, a), f fh(z, a))\}.$$

Notiamo inoltre che in questo esercizio non è possibile cancellare con risoluzione più letterali di quelli scelti: l'insieme di letterali

$$\{p(x, fx), p(x, s), p(y, y), p(fh(z, a), fu), p(u, v)\}$$

non è infatti unificabile (si dovrebbe risolvere un sistema contenente il sottosistema

$$\begin{cases} x = y \\ fx = y, \end{cases}$$

che non è unificabile.

ESERCIZI SU RISOLUZIONE PREDICATIVA E UNIFICAZIONE

1. Trovare un unificatore di massima generalità per il seguente sistema di Herbrand, o dimostrare che tale unificatore non esiste:

$$\begin{cases} f(g(x, z)) = f(g(g(fa, y), ft)) \\ g(y, u) = g(x, f(g(a, y))) \end{cases}$$

2. Data una clausola C e una sostituzione σ . dimostrare che vale $C \models C\sigma$.
3. Nei tre casi seguenti, stabilire se sia possibile derivare la clausola vuota dalle clausole C_1, C_2 con un singolo passo di risoluzione. Indicare in ogni caso il sistema di Herbrand e i passi dell'algoritmo di unificazione utilizzati per risolvere il problema.

$$(a) \quad C_1 = \{s(h(x, fy), w), s(v, fh(a, ffx))\}, \quad C_2 = \{\neg s(h(a, ffu), fh(a, ffa)), \neg s(z, fh(a, ffa))\};$$

$$(b) \quad C_1 = \{q(x, fy), q(fy, w)\}, \quad C_2 = \{\neg q(z, ffz), \neg q(v, ffv)\};$$

$$(c) \quad C_1 = \{q(x, fy), p(y, w)\}, \quad C_2 = \{\neg q(z, fa), \neg p(fv, ffv)\};$$

4. Siano C_1, C_2 le seguenti clausole:

$$C_1 = \{r(gx, y)\}, \quad C_2 = \{\neg r(z, z), \neg r(z, fu)\}.$$

È possibile ottenere la clausola vuota con un singolo passo di risoluzione? E con due?

5. Considerate le clausole

$$C_1 = \{r(h(x, a, y), h(y, f(g(x, y)), b)), r(x, b)\}$$

$$C_2 = \{\neg r(h(b, z, f(u)), h(f(z), f(g(f(u), f(z))), b))\}$$

Effettuate tutti i possibili passi di risoluzione, indicando la sostituzione utilizzata e la clausola prodotta.

6. Date le clausole

$$C_1 := \{q(hx, f(v, z))\}, \quad C_2 := \{\neg q(h(f(x, y)), f(f(x, hy), h(f(a, x)))), \neg q(x, f(y, a))\},$$

Determinare l'insieme delle clausole derivabili con un singolo passo di risoluzione da C_1, C_2 , giustificando in particolare la presenza o meno nell'insieme della clausola vuota.

7. Determinare se l'insieme di enunciati seguente è soddisfacibile o meno, usando risoluzione (nota bene: le formule dell'insieme sono logicamente equivalenti a clausole)

$$\{\forall x\forall y(r(x,y) \rightarrow q(fx)), \quad \forall z(q(z) \vee r(gz,z)), \quad \forall w\neg q(w)\}.$$