Programare logică

Algoritmul de unificare

- F. Baader, T. Nipkow,
Terms Rewriting and All That,
Cambridge University Press, 1998.

Substituţie

(S, Σ) signatură multisortată, X şi Y mulţimi de variabile
O substituţie a variabilelor din X cu termeni din $T_\Sigma(Y)$ este o funcţie $\nu : X \to T_\Sigma(Y)$. Substituţia ν se extinde la o funcţie $\tilde{\nu} : T_\Sigma(X) \to T_\Sigma(Y)$ după cum urmează:

- $\tilde{\nu}_s(x) := \nu(x)$ or. $x \in X_s$,
- $\tilde{\nu}_s(\sigma) := \sigma$ or. $\sigma :\to s$,
- $\tilde{\nu}_s(\sigma(t_1, \ldots, t_n)) := \sigma(\tilde{\nu}_{s_1}(t_1), \ldots, \tilde{\nu}_{s_n}(t_n))$ or. $\sigma : s_1 \ldots s_n \to s$, or. $t_1 \in T_\Sigma(X)_{s_1}$, ..., $t_n \in T_\Sigma(X)_{s_n}$.

- $\{x_1 \leftarrow t_1, \cdots, x_n \leftarrow t_n\}$ e notaţie pt. $\nu : X \to T_\Sigma(X)$, $\nu(x_i) := t_i$ or. $i = 1, \ldots, n$ şi $\nu(x) := x$ pt. $x \neq x_i$

Ecuaţii

(S, Σ) signatură monosortată, $S = \{s\}$
X mulţime de variabile,
$T_\Sigma(X)$ termenii cu variabile din X

- O ecuaţie este o pereche de termeni $\langle t, t' \rangle$, unde t, $t' \in T_\Sigma(X)$. Ecuaţia $\langle t, t' \rangle$ o vom nota $t \doteq t'$.
- Dacă $t = \sigma(t_1, \ldots, t_n)$ şi $t' = \tau(t'_1, \ldots, t'_k)$ atunci
  $t = t' \Leftrightarrow \sigma = \tau$, $n = k$, $t_i = t'_i$ or. i

$\doteq$ egalitate formală, $=$ egalitate efectivă

Unificare

- O problemă de unificare este o mulţime finită de ecuaţii
$$U = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$$
- Un unificator (o soluţie) pentru U este o substituţie $\nu : X \to T_\Sigma(X)$ a.î. $\nu(t_i) = \nu(t'_i)$ or. $i = 1, \ldots, n$.
- Notaţii
  - $Unif(U) :=$ mulţimea unificatorilor lui U
  - $Var(U) := \bigcup_{i=1}^{n}(Var(t_i) \bigcup Var(t'_i))$, unde $Var(t) :=$ mulţimea variabilelor care apar în $t \in T_\Sigma(X)$
  - dacă $\nu = \{x_1 \leftarrow t_1, \cdots, x_n \leftarrow t_n\}$ atunci $\{x_1 \leftarrow t_1, \cdots, x_n \leftarrow t_n\}U := \{\nu(t) \doteq \nu(t') \mid t \doteq t' \in U\}$

Unificare

- Spunem că problema de unificare $R = \{x_1 \doteq t_1, \ldots, x_n \doteq t_n\}$ este rezolvată dacă
  - $x_i \in X$, $x_i \neq x_j$ or. $i \neq j$
  - $x_i \notin \bigcup_{i=1}^{n} Var(t_i)$ or. $i = 1, \ldots, n$.
- $\{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}$ este cgu idempotent pentru R.
- Algoritmul transformă o problemă de unificare U într-o problemă de unificare R. Dacă $R = \emptyset$ atunci U nu are unificatori. În caz contrar, R este rezovată, iar substituţia determinată de R este cgu idempotent pentru U.

Algoritmul de unificare
$(S = \{s\}, \Sigma)$ signatură,
X mulţime de variabile

Intrare: $U = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$ Iniţializare: $R = U$

Se execută nedeterminist următorii paşi, atâta timp cât este posibil:

Şterge: $R \cup \{t \doteq t\} \Rightarrow R$

Orientează: $R \cup \{t \doteq x\} \Rightarrow R \cup \{x \doteq t\}$ dc. $x \in X$, $t \notin X$

Descompune:

- $R \cup \{\sigma(e_1, \ldots, e_n) \doteq \sigma(e'_1, \ldots, e'_n)\} \Rightarrow R \cup \{e_1 \doteq e'_1, \ldots, e_n \doteq e'_n\}$
- $R \cup \{\sigma(e_1, \ldots, e_n) \doteq \tau(e'_1, \ldots, e'_k)\} \Rightarrow \emptyset$ dc. $\sigma \neq \tau$

Elimină:

- $R \cup \{x \doteq t\} \Rightarrow \{x \doteq t\} \cup \{x \leftarrow t\}R$ dc.. $x \in Var(R) \setminus Var(t)$
- $R \cup \{x \doteq t\} \Rightarrow \emptyset$ dc. $x \in Var(t)$ şi $t \neq x$

Ieşire: dacă $R = \emptyset$ atunci nu există soluţii pentru U

dacă $R \neq \emptyset$ atunci R este cgu pentru U

Algoritmul de unificare

Se execută nedeterminist următorii paşi, atâta timp cât este posibil:

Şterge: $R \cup \{t \doteq t\} \Rightarrow R$

Orientează: $R \cup \{t \doteq x\} \Rightarrow R \cup \{x \doteq t\}$ dc. $x \in X$, $t \notin X$

Descompune:

- $R \cup \{\sigma(e_1, \ldots, e_n) \doteq \sigma(e'_1, \ldots, e'_n)\} \Rightarrow R \cup \{e_1 \doteq e'_1, \ldots, e_n \doteq e'_n\}$
- $R \cup \{\sigma(e_1, \ldots, e_n) \doteq \tau(e'_1, \ldots, e'_k)\} \Rightarrow \emptyset$ dc. $\sigma \neq \tau$

Elimină:

- $R \cup \{x \doteq t\} \Rightarrow \{x \doteq t\} \cup \{x \leftarrow t\}R$ dc. $x \in Var(R) \setminus Var(t)$
- $R \cup \{x \doteq t\} \Rightarrow \emptyset$ dc. $x \in Var(t)$ şi $t \neq x$

$\Sigma = \{g : s \to s, h : s \to s, f : sss \to s\}$,
$X = \{x, y, z, w\}$

$U = R = \{g(y) \doteq x, f(x, h(x), y) \doteq f(g(z), w, z)\}$

(2) $R = \{x \doteq g(y), f(x, h(x), y) \doteq f(g(z), w, z)\}$

(4) $R = \{x \doteq g(y), f(g(y), h(g(y)), y) \doteq f(g(z), w, z)\}$

(3) $R = \{x \doteq g(y), g(y) \doteq g(z), h(g(y)) \doteq w, y \doteq z\}$

(3) $R = \{x \doteq g(y), y \doteq z, h(g(y)) \doteq w, y \doteq z\}$

(4) $R = \{x \doteq g(z), y \doteq z, h(g(z)) \doteq w, z \doteq z\}$

(2)(1) $R = \{x \doteq g(z), y \doteq z, w \doteq h(g(z))\}$

(4) $R = \{x \doteq g(z), y \doteq z, w \doteq h(g(z))\}$ cgu

$\Sigma = \{a :\to s, g : s \to s, h : s \to s, f : sss \to s\}$,
$X = \{x, z, w\}$

$U = R = \{g(a) \doteq x, f(x, h(x), a) \doteq f(g(z), w, z)\}$

(2) $R = \{x \doteq g(a), f(x, h(x), a) \doteq f(g(z), w, z)\}$

(4) $R = \{x \doteq g(a), f(g(a), h(g(a)), a) \doteq f(g(z), w, z)\}$

(3) $R = \{x \doteq g(a), g(a) \doteq g(z), h(g(a)) \doteq w, a \doteq z\}$

(3) $R = \{x \doteq g(a), a \doteq z, h(g(a)) \doteq w, a \doteq z\}$

(2) $R = \{x \doteq g(a), z \doteq a, w \doteq h(g(a)), z \doteq a\}$

(4) $R = \{x \doteq g(a), z \doteq a, w \doteq h(g(a)), a \doteq a\}$

(1) $R = \{x \doteq g(a), z \doteq a, w \doteq h(g(a))\}$ cgu

$\Sigma = \{b :\to s, g : s \to s, h : s \to s, f : sss \to s\}$,
$X = \{x, y, z\}$

$U = R = \{g(y) \doteq x, f(x, h(x), y) \doteq f(g(z), b, z)\}$

(2) $R = \{x \doteq g(y), f(x, h(x), y) \doteq f(g(z), b, z)\}$

(4) $R = \{x \doteq g(y), f(g(y), h(g(y)), y) \doteq f(g(z), b, z)\}$

(3) $R = \{x \doteq g(y), g(y) \doteq g(z), h(g(y)) \doteq b, y \doteq z\}$

(3) $R = \{x \doteq g(y), y \doteq z, h(g(y)) \doteq b, y \doteq z\}$

(4) $R = \{x \doteq g(z), y \doteq z, h(g(z)) \doteq b, z \doteq z\}$

(3) $R = \emptyset$ deoarece $\boxed{\boldsymbol{b \neq h}}$

$$\Sigma = \{g : s \to s, h : s \to s, f : sss \to s\},$$
$$X \doteq \{x, y, z, w\}$$

$U = R = \{g(y) \doteq x, f(x, h(x), y) \doteq f(y, w, z)\}$

(2) $R = \{x \doteq g(y), f(x, h(x), y) \doteq f(y, w, z)\}$

(4) $R = \{x \doteq g(y), f(g(y), h(g(y)), y) \doteq f(y, w, z)\}$

(3) $R = \{x \doteq g(y), g(y) \doteq y, h(g(y)) \doteq w, y \doteq z\}$

(2) $R = \{x \doteq g(z), y \doteq g(y), h(g(y)) \doteq w, y \doteq z\}$

(4) $R = \emptyset$ deoarece $\boxed{y \in Var(g(y))}$

Terminare

Algoritmul de unificare se termină.

Fie R problema de unificare. Notăm

$n_{sol} := |\{x \in Var(R) | R = R' \cup \{x \doteq t\}, x \notin Var(R') \cup Var(t)\}|$,

$n_1 := |Var(R)| - n_{sol}$

$n_2 := \sum_{s \doteq t \in R}(|s| + |t|), \quad n_3 := |R|$

Fiecare pas al algoritmului modifică n_1, n_2, n_3

	n_1	n_2	n_3
Şterge	$\geq$	$>$	
Orientează	$\geq$	$=$	$>$
Descompune	$\geq$	$>$	
Elimină	$>$		

Dacă la execuţia unui pas (n_1, n_2, n_3) se schimbă în (n_1', n_2', n_3'), atunci $(n_1, n_2, n_3) >_{lex} (n_1', n_2', n_3')$. Algoritmul se termină deoarece (n_1, n_2, n_3) descreţe în ordine lexicografică la fiecare pas.

Corectitudine

Dacă $R \Rightarrow T$ şi $T \neq \emptyset$, atunci $Unif(R) = Unif(T)$.

Proprietatea este evident adevărată pentru Şterge şi Orientează.

Descompune. $R = R' \cup \{\sigma(e_1, \ldots, e_n) \doteq \sigma(e'_1, \ldots, e'_n)\}$

$$T = R' \cup \{e_1 \doteq e'_1, \ldots, e_n \doteq e'_n\}$$

• oricare $\nu : X \to T_\Sigma(X)$ substituţie

$\nu(\sigma(e_1, \ldots, e_n)) = \nu(\sigma(e'_1, \ldots, e'_n)) \Leftrightarrow \nu(e_i) = \nu(e'_i)$ or. i

Elimină: $R = R' \cup \{x \doteq t\}$,

$$T = \{x \doteq t\} \cup \{x \leftarrow t\}R',\ x \in Var(R') \setminus Var(t)$$

$\nu \in Unif(R) \Leftrightarrow \nu \in Unif(R')$ şi $\nu \in Unif(\{x \doteq t\})$

• dacă $\nu \in Unif(\{x \doteq t\})$ atunci $\{x \leftarrow t\}; \nu = \nu$

$\nu \in Unif(R) \Leftrightarrow \{x \leftarrow t\}; \nu \in Unif(R')$ şi $\nu \in Unif(\{x \doteq t\})$

$\Leftrightarrow \nu \in Unif(\{x \leftarrow t\}R')$ şi $\nu \in Unif(\{x \doteq t\})$

$\Leftrightarrow \nu \in Unif(T)$

Completitudine

Dacă $U \Rightarrow \cdots \Rightarrow R \Rightarrow \emptyset$ atunci R nu are soluţii.

Distingem două cazuri.

- La pasul Descompune, există $\sigma(e_1, \ldots, e_n) \doteq \tau(e'_1, \ldots, e'_k) \in R$ cu $\sigma \neq \tau$. Termenii $\sigma(e_1, \ldots, e_n)$ şi $\tau(e'_1, \ldots, e'_k)$ nu pot fi unificaţi deoarece încep cu caractere diferite.
- La pasul Elimină , există $x \doteq t \in R$ cu $x \in Var(t)$ şi $t \neq x$. Dacă ν este o substituţie, atunci $|\nu(t)| > |\nu(x)|$, deci x şi t nu au unificator.

R conţine o ecuaţie care nu poate fi unificată, deci R nu are unificator. Deoarece U şi R au aceeasi unificatori, rezultă ca U nu are unificatori.

Complexitate

- Problema de unificare
$U = \{x_1 \doteq f(x_0, x_0), x_2 \doteq f(x_1, x_1), \ldots, x_n \doteq f(x_{n-1}, x_{n-1})\}$
are cgu $R = \{x_1 \leftarrow f(x_0, x_0), x_2 \leftarrow f(f(x_0, x_0), f(x_0, x_0)), \ldots\}$.
- La pasul Elimină, pentru a verifica că o variabilă x_i nu apare în membrul drept al ecuaţiei (occur check) facem 2^i comparaţii.
- Algoritmul de unificare prezentat anterior este exponenţial. Complexitatea poate fi îmbunătăţită printr-o reprezentare eficientă a termenilor.

K. Knight, Unification: A Multidisciplinary Survey, ACM Computing Surveys, Vol. 21, No. 1, 1989.

CafeObj

```
{mod! MATCH{
[ Elem ]
op a : -> Elem
op _+_  : Elem Elem -> Elem
op _*_  : Elem Elem -> Elem
vars x y : Elem
eq x + (y * y) = x * y . }

MATCH> reduce (a + y) + (x * x ).
-- reduce in MATCH : ((a + y) + (x * x)):Elem
1>[1] rule: eq (x:Elem + (y:Elem * y))= (x * y)
    { x:Elem |-> (a + y:Elem), y:Elem |-> x:Elem }
1<[1]((a + y:Elem) + (x:Elem * x))-->((a + y:Elem)* x:Elem)
((a + y) * x):Elem
(0.000 sec for parse, 1 rewrites(0.000 sec), 3 matches)
```

Matching problem

- Fie p şi t termeni cu variabile din X. Spunem că

  p matches t (t este o instanţă a lui p)

  dacă există o substituţie $\nu : X \to T_\Sigma(X)$ a. î. $\nu(p) = t$.

  $p = x + (y * y)$, $t = (a + y) + (x * x)$

  $\nu(x) := a + y$, $\nu(y) := x$

- Fie t' termenul obţinut din t prin înlocuirea fiecărei variabile $x \in Var(T)$ cu o operaţie constantă c_x. Substituţia ν poate fi determinată aplicînd algoritmul de unificare ecuaţiei $p \doteq t'$.

  $x + (y * y) \doteq (a + c_y) + (c_x * c_x)$

  $\{x \doteq a + c_y,\ y \doteq c_x\}$

- O problemă de *matching* poate fi rezolvată prin unificare.