Curs 3

Cuprins

Termeni, substituții, unificatori

Termeni

Alfabet:

- $\mathcal{F}$ o multime de simboluri de functii de aritate cunoscuta
- $\mathcal{V}$ o multime numarabila de variabile
- $\mathcal{F}$ si $\mathcal{V}$ sunt disjuncte

Termeni

Alfabet:

- $\mathcal{F}$ o multime de simboluri de functii de aritate cunoscuta
- $\mathcal{V}$ o multime numarabila de variabile
- $\mathcal{F}$ si $\mathcal{V}$ sunt disjuncte

Termeni peste $\mathcal{F}$ si $\mathcal{V}$:

$$t \ ::= \ x \mid f(t_1, \ldots, t_n)$$

unde

- $n \geq 0$
- x este o variabila
- f este un simbol de functie de aritate n

Termeni

Notatii:

- constante: simboluri de functii de aritate 0
- $x, y, z, \ldots$ pentru variabile
- $a, b, c, \ldots$ pentru constante
- $f, g, h, \ldots$ pentru simboluri de functii arbitrare
- $s, t, u, \ldots$ pentru termeni
- $var(t)$ multimea variabilelor care apa in t
- ecuatii $s \doteq t$ pentru o pereche de termeni
- $Trm_{\mathcal{F},\mathcal{V}}$ multimea termenilor peste $\mathcal{F}$ si $\mathcal{V}$

Termeni

Exemplu

- $f(x, g(x, a), y)$ este un termen, unde f are aritate 3, g are aritate 2, a este o constanta
- $var(f(x, g(x, a), y)) = \{x, y\}$

Substituții

Definiție

O subtituție σ este o funcție (parțială) de la variabile la termeni, adică

$$\sigma : \mathcal{V} \to Trm_{\mathcal{F},\mathcal{V}}$$

Exemplu

În notația uzuală, $\sigma = \{x/a,\ y/g(w), z/b\}$. Substitutia σ este identitate pe restul variabilelor.

Notatie alternativa $\sigma = \{x \mapsto a, y \mapsto g(w), z \mapsto b\}$.

Substituții

- Substituțiile sunt o modalitate de a înlocui variabilele cu alți termeni.
- Substituțiile se aplică simultan pe toate variabilele.

Aplicarea unei substitutii σ unui termen t:

$$\sigma(t) = \begin{cases} \sigma(x), \text{ daca } t = x \\ f(\sigma(t_1), \ldots, \sigma(t_n)), \text{ daca } t = f(t_1, \ldots, t_n) \end{cases}.$$

Substituții

- Substituțiile sunt o modalitate de a înlocui variabilele cu alți termeni.
- Substituțiile se aplică simultan pe toate variabilele.

Aplicarea unei substitutii σ unui termen t:

$$\sigma(t) = \begin{cases} \sigma(x), \text{ daca } t = x \\ f(\sigma(t_1), \ldots, \sigma(t_n)), \text{ daca } t = f(t_1, \ldots, t_n) \end{cases} .$$

Exemplu

- $\sigma = \{x \mapsto f(x, y), y \mapsto g(a)\}$
- $t = f(x, g(f(x, f(y, z))))$
- $\sigma(t) = f(f(x, y), g(f(f(x, y), f(g(a), z))))$

Substituții

Două substituții σ_1 și σ_2 se pot compune

$$\sigma_1 ; \sigma_2$$

(aplicăm întâi σ_1, apoi σ_2).

Substituții

Două substituții σ_1 și σ_2 se pot compune

$$\sigma_1; \sigma_2$$

(aplicăm întâi σ_1, apoi σ_2).

Exemplu

- $t = h(u, v, x, y, z)$

Substituții

Două substituții σ_1 și σ_2 se pot compune

$$\sigma_1 ; \sigma_2$$

(aplicăm întâi σ_1, apoi σ_2).

Exemplu

- $t = h(u, v, x, y, z)$
- $\tau = \{x \mapsto f(y),\ y \mapsto f(a),\ z \mapsto u\}$
- $\sigma = \{y \mapsto g(a),\ u \mapsto z,\ v \mapsto f(f(a))\}$

Substituții

Două substituții σ_1 și σ_2 se pot compune

$$\sigma_1 ; \sigma_2$$

(aplicăm întâi σ_1, apoi σ_2).

Exemplu

- $t = h(u, v, x, y, z)$
- $\tau = \{x \mapsto f(y),\ y \mapsto f(a),\ z \mapsto u\}$
- $\sigma = \{y \mapsto g(a),\ u \mapsto z,\ v \mapsto f(f(a))\}$
- $(\tau ; \sigma)(t) = \sigma(\tau(t)) = \sigma(h(u, v, f(y), f(a), u)) =$
  $= h(z, f(f(a)), f(g(a)), f(a), z)$

Substituții

Două substituții σ_1 și σ_2 se pot compune

$$\sigma_1 ; \sigma_2$$

(aplicăm întâi σ_1, apoi σ_2).

Exemplu

- $t = h(u, v, x, y, z)$
- $\tau = \{x \mapsto f(y),\ y \mapsto f(a),\ z \mapsto u\}$
- $\sigma = \{y \mapsto g(a),\ u \mapsto z,\ v \mapsto f(f(a))\}$
- $(\tau ; \sigma)(t) = \sigma(\tau(t)) = \sigma(h(u, v, f(y), f(a), u)) =$
  $= h(z, f(f(a)), f(g(a)), f(a), z)$
- $(\sigma ; \tau)(t) = \tau(\sigma(t)) = \tau(h(z, f(f(a)), x, g(a), z))$
  $= h(u, f(f(a)), f(y), g(a), u)$

Unificare

- Doi termeni t_1 și t_2 se unifică dacă există o substituție σ astfel încât
$$\sigma(t_1) = \sigma(t_2).$$
- În acest caz, σ se numește unificatorul termenilor t_1 și t_2.

Unificare

- Doi termeni t_1 și t_2 se unifică dacă există o substituție σ astfel încât
$$\sigma(t_1) = \sigma(t_2).$$
- În acest caz, σ se numește unificatorul termenilor t_1 și t_2.
- În programarea logică, unificatorii sunt ingredientele de bază în execuția unui program.

Unificare

- Doi termeni t_1 și t_2 se unifică dacă există o substituție σ astfel încât
$$\sigma(t_1) = \sigma(t_2).$$
- În acest caz, σ se numește unificatorul termenilor t_1 și t_2.
- În programarea logică, unificatorii sunt ingredientele de bază în execuția unui program.
- Un unificator σ pentru t_1 și t_2 este un cel mai general unificator (cgu,mgu) dacă pentru orice alt unificator σ' pentru t_1 și t_2, există o substituție τ astfel încât
$$\sigma' = \sigma; \tau.$$

Unificator

Exemplu

- $t = x + (y * y) = +(x, *(y, y))$
- $t' = x + (y * x) = +(x, *(y, x))$

Unificator

Exemplu

- $t = x + (y * y) = +(x, *(y, y))$
- $t' = x + (y * x) = +(x, *(y, x))$
- $\sigma = \{x/y, y/y\}$
  - $\sigma(t) = y + (y * y)$
  - $\sigma(t') = y + (y * y)$
  - σ este cgu

Unificator

Exemplu

- $t = x + (y * y) = +(x, *(y, y))$
- $t' = x + (y * x) = +(x, *(y, x))$
- $\sigma = \{x/y, y/y\}$
  - $\sigma(t) = y + (y * y)$
  - $\sigma(t') = y + (y * y)$
  - σ este cgu
- $\sigma' = \{x/0, y/0\}$
  - $\sigma'(t) = 0 + (0 * 0)$
  - $\sigma'(t') = 0 + (0 * 0)$

Unificator

Exemplu

- $t = x + (y * y) = +(x, *(y, y))$
- $t' = x + (y * x) = +(x, *(y, x))$
- $\sigma = \{x/y, y/y\}$
  - $\sigma(t) = y + (y * y)$
  - $\sigma(t') = y + (y * y)$
  - σ este cgu
- $\sigma' = \{x/0, y/0\}$
  - $\sigma'(t) = 0 + (0 * 0)$
  - $\sigma'(t') = 0 + (0 * 0)$
  - $\sigma' = \sigma; \{y/0\}$

Unificator

Exemplu

- $t = x + (y * y) = +(x, *(y, y))$
- $t' = x + (y * x) = +(x, *(y, x))$
- $\sigma = \{x/y, y/y\}$
  - $\sigma(t) = y + (y * y)$
  - $\sigma(t') = y + (y * y)$
  - σ este cgu
- $\sigma' = \{x/0, y/0\}$
  - $\sigma'(t) = 0 + (0 * 0)$
  - $\sigma'(t') = 0 + (0 * 0)$
  - $\sigma' = \sigma; \{y/0\}$
  - σ' este unificator, dar nu este cgu

Algoritmul de unificare

Algoritmul de unificare

- Pentru o mulțime finită de termeni $\{t_1, \ldots, t_n\}$, $n \geq 2$, algoritmul de unificare stabilește dacă există un cgu.
- Există algoritmi mai eficienți, dar îl alegem pe acesta pentru simplitatea sa.

Algoritmul de unificare

- Pentru o mulțime finită de termeni $\{t_1, \ldots, t_n\}$, $n \geq 2$, algoritmul de unificare stabilește dacă există un cgu.
- Există algoritmi mai eficienți, dar îl alegem pe acesta pentru simplitatea sa.
- Algoritmul lucrează cu două liste:
  - Lista soluție: S
  - Lista de rezolvat: R

Algoritmul de unificare

- Pentru o mulțime finită de termeni $\{t_1, \ldots, t_n\}$, $n \geq 2$, algoritmul de unificare stabilește dacă există un cgu.

- Există algoritmi mai eficienți, dar îl alegem pe acesta pentru simplitatea sa.

- Algoritmul lucrează cu două liste:
  - Lista soluție: S
  - Lista de rezolvat: R

- Inițial:
  - Lista soluție: $S = \emptyset$
  - Lista de rezolvat: $R = \{t_1 \doteq t_2, \ldots, t_{n-1} \doteq t_n\}$

Algoritmul de unificare

Algoritmul constă în aplicarea regulilor de mai jos:

Algoritmul de unificare

Algoritmul constă în aplicarea regulilor de mai jos:

- SCOATE
  - orice ecuație de forma $t \doteq t$ din R este eliminată.

Algoritmul de unificare

Algoritmul constă în aplicarea regulilor de mai jos:

- SCOATE
  - orice ecuație de forma $t \doteq t$ din R este eliminată.
- DESCOMPUNE
  - orice ecuație de forma $f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n)$ din R este înlocuită cu ecuațiile $t_1 \doteq t'_1, \ldots, t_n \doteq t'_n$.

Algoritmul de unificare

Algoritmul constă în aplicarea regulilor de mai jos:

- SCOATE
  - orice ecuație de forma $t \doteq t$ din R este eliminată.
- DESCOMPUNE
  - orice ecuație de forma $f(t_1, \ldots, t_n) \doteq f(t_1', \ldots, t_n')$ din R este înlocuită cu ecuațiile $t_1 \doteq t_1', \ldots, t_n \doteq t_n'$.
- REZOLVĂ
  - orice ecuație de forma $x \doteq t$ sau $t \doteq x$ din R, unde variabila x nu apare în termenul t, este mutată sub forma $x \doteq t$ în S.
    În toate celelalte ecuații (din R și S), x este înlocuit cu t.

Algoritmul de unificare

Algoritmul se termină normal dacă $R = \emptyset$. În acest caz, S conține cgu.

Algoritmul de unificare

Algoritmul se termină normal dacă $R = \emptyset$. În acest caz, S conține cgu.

Algoritmul este oprit cu concluzia inexistenței unui cgu dacă:

1. În R există o ecuație de forma
$$f(t_1, \ldots, t_n) \doteq g(t'_1, \ldots, t'_k) \text{ cu } f \neq g.$$
2. În R există o ecuație de forma $x \doteq t$ sau $t \doteq x$ și variabila x apare în termenul t.

Algoritmul de unificare - schemă

	Lista soluție S	Lista de rezolvat R
Inițial	$\emptyset$	$t_1 \doteq t'_1, \ldots, t_n \doteq t'_n$
SCOATE	S	$R', \ t \doteq t$
	S	R'
DESCOMPUNE	S	$R', \ f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n)$
	S	$R', \ t_1 \doteq t'_1, \ldots t_n \doteq t'_n$
REZOLVĂ	S	R', $x \doteq t$ sau $t \doteq x$, x nu apare în t
	$x \doteq t, \ S[x/t]$	$R'[x/t]$
Final	S	$\emptyset$

$S[x/t]$: în toate ecuațiile din S, x este înlocuit cu t

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	REZOLVĂ

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(g(z), w, z)$	

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(g(z), w, z)$	DESCOMPUNE

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(g(z), w, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq g(z),\ h(g(y)) \doteq w,\ y \doteq z$	

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(g(z), w, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq g(z),\ h(g(y)) \doteq w,\ y \doteq z$	REZOLVĂ

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(g(z), w, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq g(z),\ h(g(y)) \doteq w,\ y \doteq z$	REZOLVĂ
$w \doteq h(g(y)),$ $x \doteq g(y)$	$g(y) \doteq g(z),\ y \doteq z$	REZOLVĂ

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(g(z), w, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq g(z),\ h(g(y)) \doteq w,\ y \doteq z$	REZOLVĂ
$w \doteq h(g(y))$, $x \doteq g(y)$	$g(y) \doteq g(z),\ y \doteq z$	REZOLVĂ
$y \doteq z, x \doteq g(z)$, $w \doteq h(g(z))$	$g(z) \doteq g(z)$	SCOATE

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(g(z), w, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq g(z),\ h(g(y)) \doteq w,\ y \doteq z$	REZOLVĂ
$w \doteq h(g(y))$, $x \doteq g(y)$	$g(y) \doteq g(z),\ y \doteq z$	REZOLVĂ
$y \doteq z, x \doteq g(z)$, $w \doteq h(g(z))$	$g(z) \doteq g(z)$	SCOATE
$y \doteq z, x \doteq g(z)$, $w \doteq h(g(z))$	$\emptyset$	

- $\sigma = \{y \mapsto z,\ x \mapsto g(z),\ w \mapsto h(g(z))\}$ este cgu.

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)\}$ au cgu?

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(y), y) \doteq f(g(z), b, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq g(z),\ h(y) \doteq b,\ y \doteq z$	- EȘEC -

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(y), y) \doteq f(g(z), b, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq g(z),\ h(y) \doteq b,\ y \doteq z$	- EȘEC -

- h și b sunt simboluri de operații diferite!
- Nu există unificator pentru acesti termeni.

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)\}$ au cgu?

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(y, w, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq y,\ h(g(y)) \doteq w,\ y \doteq z$	- EȘEC -

Exemplu

Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)\}$ au cgu?

S	R	
$\emptyset$	$g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)$	REZOLVĂ
$x \doteq g(y)$	$f(g(y), h(g(y)), y) \doteq f(y, w, z)$	DESCOMPUNE
$x \doteq g(y)$	$g(y) \doteq y,\ h(g(y)) \doteq w,\ y \doteq z$	- EȘEC -

- În ecuația $g(y) \doteq y$, variabila y apare în termenul $g(y)$.
- Nu există unificator pentru aceste ecuații.

Terminarea algoritmului

Propoziție

Algoritmul de unificare se termină.

Terminarea algoritmului

Propoziție

Algoritmul de unificare se termină.

Demonstrație

- Notăm cu
  - N_1: numărul variabilelor care apar în R
  - N_2: numărul aparițiilor simbolurilor care apar în R
- Este suficient să arătăm că perechea (N_1, N_2) descrește strict în ordine lexicografică la execuția unui pas al algoritmului:

  dacă la execuția unui pas (N_1, N_2) se schimbă în (N_1', N_2'), atunci
  $$(N_1, N_2) \geq_{lex} (N_1', N_2')$$

Demonstrație (cont.)

Fiecare regulă a algoritmului modifică N_1 și N_2 astfel:

	N_1	N_2
SCOATE	$\geq$	$>$
DESCOMPUNE	$=$	$>$
REZOLVĂ	$>$	

- N_1: numărul variabilelor care apar în R
- N_2: numărul aparițiilor simbolurilor care apar în R

□

Unificare în Prolog

- Ce se întâmplă dacă încercăm să unificăm X cu ceva care conține X? Exemplu: `?- X = f(X).`
- Conform teoriei, acești termeni nu se pot unifica.
- Totuși, multe implementări ale Prolog-ului sar peste această verificare din motive de eficiență.

```
?- X = f(X).
X = f(X).
```

Unificare în Prolog

- Ce se întâmplă dacă încercăm să unificăm X cu ceva care conține X? Exemplu: `?- X = f(X).`
- Conform teoriei, acești termeni nu se pot unifica.
- Totuși, multe implementări ale Prolog-ului sar peste această verificare din motive de eficiență.

```
?- X = f(X).
X = f(X).
```

- Putem folosi `unify_with_occurs_check/2`

```
?- unify_with_occurs_check(X,f(X)).
false.
```

Quiz time!

https://www.questionpro.com/t/AT4NiZrWmq

Corectitudinea algoritmului (optional)

Corectitudinea algoritmului

Lema 1

Mulțimea unificatorilor pentru ecuațiile din $R \cup S$ nu se modifică prin aplicarea celor trei reguli ale algoritmului de unificare.

Corectitudinea algoritmului

Lema 1

Mulțimea unificatorilor pentru ecuațiile din $R \cup S$ nu se modifică prin aplicarea celor trei reguli ale algoritmului de unificare.

Demonstrație

Analizăm fiecare regulă:

- SCOATE: evident

Corectitudinea algoritmului

Lema 1

Mulțimea unificatorilor pentru ecuațiile din $R \cup S$ nu se modifică prin aplicarea celor trei reguli ale algoritmului de unificare.

Demonstrație

Analizăm fiecare regulă:

- SCOATE: evident
- DESCOMPUNE: Trebuie să arătăm că

$$\begin{array}{ccc} \nu \text{ unificator pt.} & \Leftrightarrow & \nu \text{ unificator pt.} \\ f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n) & & t_i \doteq t'_i, \text{ or. } i = 1, \ldots, n. \end{array}$$

Corectitudinea algoritmului

Lema 1

Mulțimea unificatorilor pentru ecuațiile din $R \cup S$ nu se modifică prin aplicarea celor trei reguli ale algoritmului de unificare.

Demonstrație

Analizăm fiecare regulă:

- SCOATE: evident
- DESCOMPUNE: Trebuie să arătăm că

$$\begin{array}{ccc} \nu \text{ unificator pt.} & \Leftrightarrow & \nu \text{ unificator pt.} \\ f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n) & & t_i \doteq t'_i, \text{ or. } i = 1, \ldots, n. \end{array}$$

$$\begin{aligned} &\nu \text{ unif. pt. } f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n) \\ &\Leftrightarrow \nu(f(t_1, \ldots, t_n)) = \nu(f(t'_1, \ldots, t'_n)) \\ &\Leftrightarrow f(\nu(t_1), \ldots, \nu(t_n)) = f(\nu(t'_1), \ldots, \nu(t'_n)) \\ &\Leftrightarrow \nu(t_i) = \nu(t'_i), \text{or. } i = 1, \ldots, n \\ &\Leftrightarrow \nu \text{ unificator pt. } t_i \doteq t'_i, \text{or. } i = 1, \ldots, n \end{aligned}$$

Demonstrație (cont.)

- REZOLVĂ:
  - Se observă că orice unificator ν pentru ecuațiile din $R \cup S$, atât înainte cât și după aplicarea regulii REZOLVĂ, trebuie să satisfacă:
    $$\nu(x) = \nu(t).$$
  - Dacă μ este unificator pentru $x \doteq t$ observăm că:
    $$(x \leftarrow t); \mu = \mu$$
    unde $(x \leftarrow t)(x) = t$ și $(x \leftarrow t)(y) = y$ pentru orice $y \neq x \in V$.
    - $((x \leftarrow t); \mu)(x) = \mu(t) = \mu(x)$
    - $((x \leftarrow t); \mu)(y) = \mu(y)$, pentru orice $y \neq x$
  - Deci,

μ este un unificator pentru ecuațiile din $R \cup S$ înainte de REZOLVĂ

$$\Leftrightarrow$$

μ este un unificator pentru ecuațiile din $R \cup S$ după REZOLVĂ $\square$

Corectitudinea algoritmului

- Pres. că algoritmul de unificare se termină cu $R = \emptyset$.
- Fie $x_i \doteq t_i$, $i = 1, \ldots, k$, ecuațiile din S.
- Variabilele care apar în partea stângă a ecuațiilor din S sunt distincte două câte două și nu mai apar în termenii $t1$, ..., t_k.
- Definim substituția:

$$\nu(x_i) = t_i \text{ pentru orice } i = 1, \ldots, k.$$

- Observăm că $\nu(t_i) = t_i = \nu(x_i)$ oricare $i = 1, \ldots, k$, deci ν este un unificator pentru $R \cup S$.

Corectitudinea algoritmului

Lema 2

ν definit mai sus cf. algoritmului de unificare este cgu pentru $R \cup S$.

Corectitudinea algoritmului

Lema 2

ν definit mai sus cf. algoritmului de unificare este cgu pentru $R \cup S$.

Demonstrație

La ultimul pas $R = \emptyset$ și $\nu(x_i) = t_i$ oricare $i = 1, \ldots, k$

- Fie μ un alt unificator pentru S. Avem
  - $\mu(\nu(x_i)) = \mu(t_i) = \mu(x_i)$, or. $i = 1, \ldots, k$,
  - $\mu(\nu(y)) = \mu(y)$, or. $y \neq x$.

  Deci $\nu; \mu = \mu$. În concluzie, ν este cgu deoarece oricare alt

  unificator se poate scrie ca o compunere a lui ν cu o substituție. □

- Din Lema 1 rezultă că ν este unificator pentru problema inițială $\{u_1 \doteq u_2, \ldots, u_{n-1} \doteq u_n\}$, deci

$$\nu(u_1) = \cdots = \nu(u_n).$$

Complexitatea algoritmului

- Problema de unificare

  $R = \{x_1 \doteq f(x_0, x_0), x_2 \doteq f(x_1, x_1), \ldots, x_n \doteq f(x_{n-1}, x_{n-1})\}$

  are cgu $S = \{x_1 \leftarrow f(x_0, x_0), x_2 \leftarrow f(f(x_0, x_0), f(x_0, x_0)), \ldots\}$.

Complexitatea algoritmului

- Problema de unificare

  $R = \{x_1 \doteq f(x_0, x_0), x_2 \doteq f(x_1, x_1), \ldots, x_n \doteq f(x_{n-1}, x_{n-1})\}$

  are cgu $S = \{x_1 \leftarrow f(x_0, x_0), x_2 \leftarrow f(f(x_0, x_0), f(x_0, x_0)), \ldots\}$.

- La pasul Elimină, pentru a verifica că o variabilă x_i nu apare în membrul drept al ecuației (occur check) facem 2^i comparații.

Complexitatea algoritmului

- Problema de unificare

  $R = \{x_1 \doteq f(x_0, x_0), x_2 \doteq f(x_1, x_1), \ldots, x_n \doteq f(x_{n-1}, x_{n-1})\}$

  are cgu $S = \{x_1 \leftarrow f(x_0, x_0), x_2 \leftarrow f(f(x_0, x_0), f(x_0, x_0)), \ldots\}$.

- La pasul Elimină, pentru a verifica că o variabilă x_i nu apare în membrul drept al ecuației (occur check) facem 2^i comparații.

- Algoritmul de unificare prezentat anterior este exponențial. Complexitatea poate fi îmbunătățită printr-o reprezentare eficientă a termenilor.

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

Pe săptămâna viitoare!