Curs 6

Amintiri

O (S, Σ)-ecuație condiționată $(\forall X)t \doteq_s t'$ *if* H este formată din

- o mulțime de variabile X,
- doi termeni de același sort $t, t' \in T_\Sigma(X)_s$,
- o mulțime H de ecuații $u \doteq_{s'} v$, cu $u, v \in T_\Sigma(X)_{s'}$.

Amintiri

O (S, Σ)-ecuație condiționată $(\forall X)t \doteq_s t'$ *if* H este formată din

- o mulțime de variabile X,
- doi termeni de același sort $t, t' \in T_\Sigma(X)_s$,
- o mulțime H de ecuații $u \doteq_{s'} v$, cu $u, v \in T_\Sigma(X)_{s'}$.

O (S, Σ)-algebră $\mathcal{A} = (A_S, A_\Sigma)$ satisface o ecuație condiționată $(\forall X)t \doteq_s t'$ *if* H dacă pentru orice funcție S-sortată $e : X \to A_S$,

$$\tilde{e}_{s'}(u) = \tilde{e}_{s'}(v), \text{ or. } u \doteq_{s'} v \in H \Rightarrow \tilde{e}_s(t) = \tilde{e}_s(t').$$

Amintiri

O (S, Σ)-ecuație condiționată $(\forall X)t \doteq_s t'$ *if* H este formată din

- o mulțime de variabile X,
- doi termeni de același sort $t, t' \in T_\Sigma(X)_s$,
- o mulțime H de ecuații $u \doteq_{s'} v$, cu $u, v \in T_\Sigma(X)_{s'}$.

O (S, Σ)-algebră $\mathcal{A} = (A_S, A_\Sigma)$ satisface o ecuație condiționată $(\forall X)t \doteq_s t'$ *if* H dacă pentru orice funcție S-sortată $e : X \to A_S$,

$$\tilde{e}_{s'}(u) = \tilde{e}_{s'}(v), \text{ or. } u \doteq_{s'} v \in H \Rightarrow \tilde{e}_s(t) = \tilde{e}_s(t').$$

O (S, Σ)-algebră $\mathcal{A}$ este o Γ-algebră ($\mathcal{A}$ este model pentru Γ) dacă

$$\mathcal{A} \models \gamma, \text{ or. } \gamma \in \Gamma.$$

Amintiri

O (S,Σ)-ecuație condiționată $(\forall X)t \doteq_s t' \ \mathit{if}\ H$ este formată din

- o mulțime de variabile X,
- doi termeni de același sort $t, t' \in T_\Sigma(X)_s$,
- o mulțime H de ecuații $u \doteq_{s'} v$, cu $u, v \in T_\Sigma(X)_{s'}$.

O (S,Σ)-algebră $\mathcal{A} = (A_S, A_\Sigma)$ satisface o ecuație condiționată $(\forall X)t \doteq_s t' \ \mathit{if}\ H$ dacă pentru orice funcție S-sortată $e : X \to A_S$,

$$\tilde{e}_{s'}(u) = \tilde{e}_{s'}(v), \text{ or. } u \doteq_{s'} v \in H \Rightarrow \tilde{e}_s(t) = \tilde{e}_s(t').$$

O (S,Σ)-algebră $\mathcal{A}$ este o Γ-algebră ($\mathcal{A}$ este model pentru Γ) dacă

$$\mathcal{A} \models \gamma, \text{ or. } \gamma \in \Gamma.$$

Notăm cu $Alg(S,\Sigma,\Gamma)$ clasa tuturor Γ-algebrelor.

Amintiri

Un tip abstract de date este o clasă $\mathfrak{C}$ de (S, Σ)-algebre cu proprietatea că oricare două (S, Σ)-algebre din $\mathfrak{C}$ sunt izomorfe:

$$\mathcal{A}, \mathcal{B} \in \mathfrak{C} \Rightarrow \mathcal{A} \simeq \mathcal{B}.$$

Amintiri

Un tip abstract de date este o clasă $\mathfrak{C}$ de (S, Σ)-algebre cu proprietatea că oricare două (S, Σ)-algebre din $\mathfrak{C}$ sunt izomorfe:

$$\mathcal{A}, \mathcal{B} \in \mathfrak{C} \Rightarrow \mathcal{A} \simeq \mathcal{B}.$$

Dacă $\mathcal{A}_1$ și $\mathcal{A}_2$ sunt inițiale în $\mathfrak{K}$, atunci $\mathcal{A}_1 \simeq \mathcal{A}_2$.

Amintiri

Un tip abstract de date este o clasă $\mathfrak{C}$ de (S, Σ)-algebre cu proprietatea că oricare două (S, Σ)-algebre din $\mathfrak{C}$ sunt izomorfe:

$$\mathcal{A}, \mathcal{B} \in \mathfrak{C} \Rightarrow \mathcal{A} \simeq \mathcal{B}.$$

Dacă $\mathcal{A}_1$ și $\mathcal{A}_2$ sunt inițiale în $\mathfrak{K}$, atunci $\mathcal{A}_1 \simeq \mathcal{A}_2$.

T_Σ este (S, Σ)-algebră inițială, i.e.

pentru orice (S, Σ)-algebră $\mathcal{B}$ există un unic morfism $f : T_\Sigma \to \mathcal{B}$.

Amintiri

Un tip abstract de date este o clasă $\mathfrak{C}$ de (S, Σ)-algebre cu proprietatea că oricare două (S, Σ)-algebre din $\mathfrak{C}$ sunt izomorfe:

$$\mathcal{A}, \mathcal{B} \in \mathfrak{C} \Rightarrow \mathcal{A} \simeq \mathcal{B}.$$

Dacă $\mathcal{A}_1$ și $\mathcal{A}_2$ sunt inițiale în $\mathfrak{K}$, atunci $\mathcal{A}_1 \simeq \mathcal{A}_2$.

T_Σ este (S, Σ)-algebră inițială, i.e.

pentru orice (S, Σ)-algebră $\mathcal{B}$ există un unic morfism $f : T_\Sigma \to \mathcal{B}$.

$T_\Sigma(X)$ este (S, Σ)-algebră liber generată de X, i.e.

pentru orice (S, Σ)-algebră $\mathcal{B} = (B_S, B_\Sigma)$, orice funcție S-sortată $e : X \to B_S$ se extinde unic la un (S, Σ)-morfism $\tilde{e} : T_\Sigma(X) \to \mathcal{B}$.

Cuprins

Specificații algebrice

Specificații

O specificație este un triplet (S, Σ, Γ), unde

- (S, Σ) este o signatură multisortată
- Γ este o mulțime de ecuații condiționate

Specificații

O specificație este un triplet (S, Σ, Γ), unde

- (S, Σ) este o signatură multisortată
- Γ este o mulțime de ecuații condiționate

Specificația (S, Σ, Γ) definește clasa modelelor $Alg(S, \Sigma, \Gamma)$, care reprezintă semantica ei.

Specificații echivalente

Definitie

Două specificații (S, Σ, Γ_1) și (S, Σ, Γ_2) sunt echivalente dacă definesc aceeași clasă de modele, i.e.

$$\mathcal{A} \models \Gamma_1 \Leftrightarrow \mathcal{A} \models \Gamma_2$$

- Dacă Γ și Θ sunt mulțimi de ecuații condiționate a.î. $\Gamma \models \Theta$, atunci (S, Σ, Γ) și $(S, \Sigma, \Gamma \cup \Theta)$ sunt specificații echivalente.

Semantica unei teorii în **Maude**

În **Maude**, o teorie `fth ... endfth` are ca semantică $Alg(S, \Sigma, \Gamma)$

- S mulțimea sorturilor
- Σ mulțimea simbolurilor de operații
- Γ mulțimea ecuațiilor definite în modul, iar fiecare ecuație

  `eq t = t'` și `ceq t = t' if H`

  este cuantificată de variabilele care apar în `t` și `t'`.

Amintiri

Pe T_Σ definim congruența semantică determinată de Γ:

$$\equiv_{\Gamma, T_\Sigma} := \bigcap \{Ker(f) \mid f : T_\Sigma \to \mathcal{B},\ \mathcal{B} \models \Gamma\}$$

Teorema ($\star$)

$T_\Sigma/_{\equiv_{\Gamma, T_\Sigma}}$ este Γ-algebra inițială.

Semantica unui modul în **Maude**

Fie (S, Σ) o signatură multisortată și Γ o mulțime de ecuații condiționate.

$$\mathfrak{I}_{(S,\Sigma,\Gamma)} = \{\mathcal{I} \mid \mathcal{I} \ \Gamma\text{-algebra inițială}\}$$

- $\mathfrak{I}_{(S,\Sigma,\Gamma)}$ este un tip abstract de date

Semantica unui modul în **Maude**

Fie (S, Σ) o signatură multisortată și Γ o mulțime de ecuații condiționate.

$$\mathfrak{I}_{(S,\Sigma,\Gamma)} = \{\mathcal{I} \mid \mathcal{I}\ \Gamma\text{-algebra inițială}\}$$

- $\mathfrak{I}_{(S,\Sigma,\Gamma)}$ este un tip abstract de date

În **Maude**, un modul `fmod ... endfm` definește tipul abstract de date $\mathfrak{I}_{(S,\Sigma,\Gamma)}$ și construiește efectiv algebra $T_\Sigma/_{\equiv_{\Gamma,T_\Sigma}}$

- S mulțimea sorturilor
- Σ mulțimea simbolurilor de operații
- Γ mulțimea ecuațiilor definite în modul, iar fiecare ecuație

  `eq t = t'` și `ceq t = t' if H`

  este cuantificată de variabilele care apar în `t` și `t'`.

Specificație corectă

Fie (S, Σ) o signatură multisortată și $\mathcal{A}$ o (S, Σ)-algebră.

Definitie

O specificație (S, Σ, Γ) este adecvată pentru $\mathcal{A}$ dacă $\mathcal{A}$ este Γ-algebră inițială, i.e.

$$\mathcal{A} \in \mathfrak{I}_{(S,\Sigma,\Gamma)}.$$

Exemplu

Exemplu

- $S = \{s\}$
- $\Sigma = \{0 :\to s,\ succ : s \to s\}$
- $\Gamma = \{(\forall x) succ(succ(succ(succ(x)))) \doteq x\}$

Exemplu

Exemplu

- $S = \{s\}$
- $\Sigma = \{0 :\to s,\ succ : s \to s\}$
- $\Gamma = \{(\forall x) succ(succ(succ(succ(x)))) \doteq x\}$

(S, Σ, Γ) este o specificație adecvată pentru $\mathcal{A} = (\mathbb{Z}_4, 0, succ)$, unde $A_{succ}(x) := (x + 1) \bmod 4$.

Exemplu

Exemplu

- $S = \{s\}$
- $\Sigma = \{0 :\to s,\ succ : s \to s\}$
- $\Gamma = \{(\forall x) succ(succ(succ(succ(x)))) \doteq x\}$

(S, Σ, Γ) este o specificație adecvată pentru $\mathcal{A} = (\mathbb{Z}_4, 0, succ)$, unde $A_{succ}(x) := (x + 1) \text{mod } 4$.

Se reduce la a arăta că $\mathcal{A}$ este Γ-algebra inițială, i.e.

1. $\mathcal{A} \models \gamma$, or. $\gamma \in \Gamma$,
2. pt. or. Γ-algebră $\mathcal{B}$, există un unic morfism $f : \mathcal{A} \to \mathcal{B}$.

Exemplu

Exemplu

$\mathcal{A} = (\mathbb{Z}_4, 0, succ)$, unde $A_{succ}(x) := (x+1)\text{mod } 4$.

1. $\mathcal{A} \models (\forall x) succ(succ(succ(succ(x)))) \doteq x$

Exemplu

Exemplu

$\mathcal{A} = (\mathbb{Z}_4, 0, succ)$, unde $A_{succ}(x) := (x+1) \bmod 4$.

1. $\mathcal{A} \models (\forall x) succ(succ(succ(succ(x)))) \doteq x$
   - Fie $e : X \to \mathbb{Z}_4$, unde $X = \{x\}$.

Exemplu

Exemplu

$\mathcal{A} = (\mathbb{Z}_4, 0, succ)$, unde $A_{succ}(x) := (x+1) \bmod 4$.

1. $\mathcal{A} \models (\forall x) succ(succ(succ(succ(x)))) \doteq x$
   - Fie $e : X \to \mathbb{Z}_4$, unde $X = \{x\}$.
   - Avem

$$\begin{aligned} \tilde{e}(succ(succ(succ(succ(x))))) &= A_{succ}(A_{succ}(A_{succ}(A_{succ}(e(x))))) \\ &= (e(x) + 4) \bmod 4 \\ &= e(x) = \tilde{e}(x) \end{aligned}$$

Exemplu

Exemplu

$\mathcal{A} = (\mathbb{Z}_4, 0, succ)$, unde $A_{succ}(x) := (x+1) \text{mod } 4$.

1. $\mathcal{A} \models (\forall x) succ(succ(succ(succ(x)))) \doteq x$
   - Fie $e : X \to \mathbb{Z}_4$, unde $X = \{x\}$.
   - Avem

$$\begin{aligned} \tilde{e}(succ(succ(succ(succ(x))))) &= A_{succ}(A_{succ}(A_{succ}(A_{succ}(e(x))))) \\ &= (e(x)+4) \text{mod } 4 \\ &= e(x) = \tilde{e}(x) \end{aligned}$$

2. Fie $\mathcal{B}$ o Γ-algebră.

   Existența: Definim $f : \mathbb{Z}_4 \to B$ prin

Exemplu

Exemplu

$\mathcal{A} = (\mathbb{Z}_4, 0, succ)$, unde $A_{succ}(x) := (x+1) \bmod 4$.

1. $\mathcal{A} \models (\forall x) succ(succ(succ(succ(x)))) \doteq x$
   - Fie $e : X \to \mathbb{Z}_4$, unde $X = \{x\}$.
   - Avem

$$\begin{aligned} \tilde{e}(succ(succ(succ(succ(x))))) &= A_{succ}(A_{succ}(A_{succ}(A_{succ}(e(x))))) \\ &= (e(x)+4) \bmod 4 \\ &= e(x) = \tilde{e}(x) \end{aligned}$$

2. Fie $\mathcal{B}$ o Γ-algebră.

   Existența: Definim $f : \mathbb{Z}_4 \to B$ prin
   - $f(0) := B_0$
   - $f(x+1) := B_{succ}(f(x))$, pt. $0 \leq x \leq 2$

Exemplu

Exemplu

2. Arătăm că f este morfism:

Exemplu

Exemplu

2. Arătăm că f este morfism:
   - $f(A_0) = f(0) = B_0$

Exemplu

Exemplu

2. Arătăm că f este morfism:
   - $f(A_0) = f(0) = B_0$
   - $f(A_{succ}(x)) = f(x+1) = B_{succ}(f(x))$, pt. $0 \leq x \leq 2$

Exemplu

Exemplu

2. Arătăm că f este morfism:
   - $f(A_0) = f(0) = B_0$
   - $f(A_{succ}(x)) = f(x+1) = B_{succ}(f(x))$, pt. $0 \leq x \leq 2$
   - Trebuie să mai arătăm că $f(A_{succ}(3)) = B_{succ}(f(3))$:

Exemplu

Exemplu

2. Arătăm că f este morfism:
   - $f(A_0) = f(0) = B_0$
   - $f(A_{succ}(x)) = f(x+1) = B_{succ}(f(x))$, pt. $0 \leq x \leq 2$
   - Trebuie să mai arătăm că $f(A_{succ}(3)) = B_{succ}(f(3))$:
     - $f(A_{succ}(3)) = f(0) = B_0$

Exemplu

Exemplu

2. Arătăm că f este morfism:
   - $f(A_0) = f(0) = B_0$
   - $f(A_{succ}(x)) = f(x+1) = B_{succ}(f(x))$, pt. $0 \leq x \leq 2$
   - Trebuie să mai arătăm că $f(A_{succ}(3)) = B_{succ}(f(3))$:
     - $f(A_{succ}(3)) = f(0) = B_0$
     - $B_{succ}(f(3)) = B_{succ}(B_{succ}(B_{succ}(B_{succ}(B_0))))$

Exemplu

Exemplu

2. Arătăm că f este morfism:
   - $f(A_0) = f(0) = B_0$
   - $f(A_{succ}(x)) = f(x+1) = B_{succ}(f(x))$, pt. $0 \leq x \leq 2$
   - Trebuie să mai arătăm că $f(A_{succ}(3)) = B_{succ}(f(3))$:
     - $f(A_{succ}(3)) = f(0) = B_0$
     - $B_{succ}(f(3)) = B_{succ}(B_{succ}(B_{succ}(B_{succ}(B_0))))$
     - Cum $\mathcal{B} \models (\forall x) succ(succ(succ(succ(x)))) \doteq x$, pt. $e' : X \to B$, $e'(x) := B_0$, obținem

       $$B_{succ}(B_{succ}(B_{succ}(B_{succ}(B_0)))) = \tilde{e}'(succ(succ(succ(succ(x))))) = \tilde{e}'(x) = B_0$$
     - Deci $f(A_{succ}(3)) = B_{succ}(f(3))$.

Exemplu

Exemplu

2. Arătăm că f este morfism:
   - $f(A_0) = f(0) = B_0$
   - $f(A_{succ}(x)) = f(x+1) = B_{succ}(f(x))$, pt. $0 \leq x \leq 2$
   - Trebuie să mai arătăm că $f(A_{succ}(3)) = B_{succ}(f(3))$:
     - $f(A_{succ}(3)) = f(0) = B_0$
     - $B_{succ}(f(3)) = B_{succ}(B_{succ}(B_{succ}(B_{succ}(B_0))))$
     - Cum $\mathcal{B} \models (\forall x) succ(succ(succ(succ(x)))) \dot{=} x$, pt. $e' : X \to B$, $e'(x) := B_0$, obținem

       $$B_{succ}(B_{succ}(B_{succ}(B_{succ}(B_0)))) = \tilde{e'}(succ(succ(succ(succ(x))))) = \tilde{e'}(x) = B_0$$
     - Deci $f(A_{succ}(3)) = B_{succ}(f(3))$.

Unicitatea: Fie $g : \mathcal{A} \to \mathcal{B}$ un morfism.
Arătăm că $g(x) = f(x)$, or. $x \in \{0, 1, 2, 3\}$, prin inducție:

- $g(0) = g(A_0) = B_0 = f(0)$
- $g(x+1) = g(A_{succ}(x)) = B_{succ}(g(x)) = B_{succ}(f(x)) = f(A_{succ}(x)) = f(x+1)$

Substituții

Substituție

Fie (S, Σ) o signatură multisortată și X, Y mulțimi de variabile.

- Am văzut că, pentru orice (S, Σ)-algebră $\mathcal{B} = (B_S, B_\Sigma)$, orice funcție S-sortată $e : X \to B_S$ se extinde unic la un morfism $\tilde{e} : T_\Sigma(X) \to \mathcal{B}$.

Substituție

Fie (S, Σ) o signatură multisortată și X, Y mulțimi de variabile.

- Am văzut că, pentru orice (S, Σ)-algebră $\mathcal{B} = (B_S, B_\Sigma)$, orice funcție S-sortată $e : X \to B_S$ se extinde unic la un morfism $\tilde{e} : T_\Sigma(X) \to \mathcal{B}$.
- Ce se întâmplă dacă $\mathcal{B}$ este liber generată de Y, i.e. $B \simeq T_\Sigma(Y)$?

Substituție

Fie (S, Σ) o signatură multisortată și X, Y mulțimi de variabile.

- Am văzut că, pentru orice (S, Σ)-algebră $\mathcal{B} = (B_S, B_\Sigma)$, orice funcție S-sortată $e : X \to B_S$ se extinde unic la un morfism $\tilde{e} : T_\Sigma(X) \to \mathcal{B}$.
- Ce se întâmplă dacă $\mathcal{B}$ este liber generată de Y, i.e. $B \simeq T_\Sigma(Y)$?

Definitie

O substituție a variabilelor din X cu termeni din $T_\Sigma(Y)$ este o funcție S-sortată

$$\tau : X \to T_\Sigma(Y).$$

Proprietăți

- $\{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}$ este notația uzuală pentru $\sigma : X \to T_\Sigma(X)$, cu $\sigma(x_i) = t_i$ și $\sigma(x) = x$, pt. $x \neq x_i$, or. $i = 1, \ldots, n$.

Proprietăți

- $\{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}$ este notația uzuală pentru $\sigma : X \to T_\Sigma(X)$, cu $\sigma(x_i) = t_i$ și $\sigma(x) = x$, pt. $x \neq x_i$, or. $i = 1, \ldots, n$.
- O substituție $\tau : X \to T_\Sigma(Y)$ se extinde unic la un (S, Σ)-morfism
$$\tilde{\tau} : T_\Sigma(X) \to T_\Sigma(Y)$$
  - $\tilde{\tau}_s(x) := \tau_s(x)$, or. $x \in X_s$,
  - $\tilde{\tau}_s(\sigma) := \sigma$, or. $\sigma :\to s$,
  - $\tilde{\tau}_s(\sigma(t_1, \ldots, t_n)) := \sigma(\tilde{\tau}_{s_1}(t_1), \ldots, \tilde{\tau}_{s_n}(t_n))$, or. $\sigma : s_1 \ldots s_n \to s$ și or. $t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$.

Proprietăți

- $\{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}$ este notația uzuală pentru $\sigma : X \to T_\Sigma(X)$, cu $\sigma(x_i) = t_i$ și $\sigma(x) = x$, pt. $x \neq x_i$, or. $i = 1, \ldots, n$.

- O substituție $\tau : X \to T_\Sigma(Y)$ se extinde unic la un (S, Σ)-morfism
$$\tilde{\tau} : T_\Sigma(X) \to T_\Sigma(Y)$$
  - $\tilde{\tau}_s(x) := \tau_s(x)$, or. $x \in X_s$,
  - $\tilde{\tau}_s(\sigma) := \sigma$, or. $\sigma :\to s$,
  - $\tilde{\tau}_s(\sigma(t_1, \ldots, t_n)) := \sigma(\tilde{\tau}_{s_1}(t_1), \ldots, \tilde{\tau}_{s_n}(t_n))$, or. $\sigma : s_1 \ldots s_n \to s$ și or. $t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$.

- Vom indentifica uneori $\tilde{\tau}$ cu τ.

Proprietăți

Fie X, Y și Z mulțimi de variabile.

- Compunerea substituțiilor $\tau : X \to T_\Sigma(Y)$ și $\mu : Y \to T_\Sigma(Z)$ este

$$\tau; \mu : X \to T_\Sigma(Z)$$

$$(\tau; \mu)_s(x) := (\tau; \tilde{\mu})_s(x),$$

$$\text{or. } x \in X_s.$$

- Compunerea substituțiilor este asociativă.
- Compunerea substituțiilor nu este în general comutativă.

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{a :\to s,\ f : s \to s,\ g : s \to s,\ p : sssss \to s\}$
- $X = \{x, y, z, u, v\}$
- $t = p(u, v, x, y, z) \in T_\Sigma(X)$

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{a :\to s,\ f : s \to s,\ g : s \to s,\ p : sssss \to s\}$
- $X = \{x, y, z, u, v\}$
- $t = p(u, v, x, y, z) \in T_\Sigma(X)$
- $\tau : X \to T_\Sigma(X)$, $\tau = \{x \leftarrow f(y),\ y \leftarrow f(a),\ z \leftarrow u\}$
- $\tilde{\tau}(t) = p(u, v, f(y), f(a), u)$

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{a :\to s,\ f : s \to s,\ g : s \to s,\ p : sssss \to s\}$
- $X = \{x, y, z, u, v\}$
- $t = p(u, v, x, y, z) \in T_\Sigma(X)$
- $\tau : X \to T_\Sigma(X)$, $\tau = \{x \leftarrow f(y),\ y \leftarrow f(a),\ z \leftarrow u\}$
- $\tilde{\tau}(t) = p(u, v, f(y), f(a), u)$
- $\mu : X \to T_\Sigma(X)$, $\mu = \{y \leftarrow g(a),\ u \leftarrow z,\ v \leftarrow f(f(a))\}$
- $\tilde{\mu}(t) = p(z, f(f(a)), x, g(a), z)$

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{a :\to s,\ f : s \to s,\ g : s \to s,\ p : sssss \to s\}$
- $X = \{x, y, z, u, v\}$
- $t = p(u, v, x, y, z) \in T_\Sigma(X)$
- $\tau : X \to T_\Sigma(X)$, $\tau = \{x \leftarrow f(y),\ y \leftarrow f(a),\ z \leftarrow u\}$
- $\tilde{\tau}(t) = p(u, v, f(y), f(a), u)$
- $\mu : X \to T_\Sigma(X)$, $\mu = \{y \leftarrow g(a),\ u \leftarrow z,\ v \leftarrow f(f(a))\}$
- $\tilde{\mu}(t) = p(z, f(f(a)), x, g(a), z)$
- $(\tilde{\tau}; \tilde{\mu})(t) = \tilde{\mu}(\tilde{\tau}(t)) = \tilde{\mu}(p(u, v, f(y), f(a), u)) =$
  $= p(z, f(f(a)), f(g(a)), f(a), z)$

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{a :\to s,\ f : s \to s,\ g : s \to s,\ p : sssss \to s\}$
- $X = \{x, y, z, u, v\}$
- $t = p(u, v, x, y, z) \in T_\Sigma(X)$
- $\tau : X \to T_\Sigma(X)$, $\tau = \{x \leftarrow f(y),\ y \leftarrow f(a),\ z \leftarrow u\}$
- $\tilde{\tau}(t) = p(u, v, f(y), f(a), u)$
- $\mu : X \to T_\Sigma(X)$, $\mu = \{y \leftarrow g(a),\ u \leftarrow z,\ v \leftarrow f(f(a))\}$
- $\tilde{\mu}(t) = p(z, f(f(a)), x, g(a), z)$
- $(\tilde{\tau}; \tilde{\mu})(t) = \tilde{\mu}(\tilde{\tau}(t)) = \tilde{\mu}(p(u, v, f(y), f(a), u)) =$
  $= p(z, f(f(a)), f(g(a)), f(a), z)$
- $(\tilde{\mu}; \tilde{\tau})(t) = \tilde{\tau}(\tilde{\mu}(t)) = \tilde{\tau}(p(z, f(f(a)), x, g(a), z))$
  $= p(u, f(f(a)), f(y), g(a), u)$

Algoritmul de unificare

Cazul monosortat

- (S, Σ) signatură monosortată, i.e. $S = \{s\}$.
- X mulțime de variabile și $T_\Sigma(X)$ termenii cu variabile din X.
- O ecuație constă în doi termeni $t, t' \in T_\Sigma(X)$ și o notăm

$$t \doteq t'.$$

- În cazul monosortat cuantificarea înaintea unei ecuații nu este necesară.
- Egalitatea termenilor:
  dacă $t = \sigma(t_1, \ldots, t_n)$ și $t' = \tau(t'_1, \ldots, t'_k)$ atunci

$$t = t' \Leftrightarrow \sigma = \tau,\ n = k \text{ și } t_i = t'_i, \text{ or. } i$$

$\doteq$ egalitate formală $\qquad$ $=$ egalitate efectivă

Unificare. Cazul monosortat

Fie (S, Σ) signatură monosortată și X mulțime de variabile.

Problema unificării:

Pentru o mulțime finită de ecuații $U = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$ găsiți un unificator.

Unificare. Cazul monosortat

Fie (S, Σ) signatură monosortată și X mulțime de variabile.

Problema unificării:

Pentru o mulțime finită de ecuații $U = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$
găsiți un unificator.

- Un unificator pentru U este o substituție $\nu : X \to T_\Sigma(X)$ a.î.
  $\nu(t_i) = \nu(t'_i)$, or. $i = 1, \ldots, n$.

Unificare. Cazul monosortat

Fie (S, Σ) signatură monosortată și X mulțime de variabile.

Problema unificării:

Pentru o mulțime finită de ecuații $U = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$
găsiți un unificator.

- Un unificator pentru U este o substituție $\nu : X \to T_\Sigma(X)$ a.î.
$$\nu(t_i) = \nu(t'_i), \text{ or. } i = 1, \ldots, n.$$
- Un unificator ν pentru U este un cel mai general unificator (cgu,mgu) dacă pentru orice alt unificator ν' pentru U, există o substituție μ astfel încât
$$\nu' = \nu; \mu.$$

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{0 :\to s,\ + : ss \to s,\ \star : ss \to s\}$
- $X = \{x, y\}$
- $t = x + (y \star y) = +(x, \star(y, y))$
- $t' = x + (y \star x) = +(x, \star(y, x))$

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{0 :\to s,\ + : ss \to s,\ \star : ss \to s\}$
- $X = \{x, y\}$
- $t = x + (y \star y) = +(x, \star(y, y))$
- $t' = x + (y \star x) = +(x, \star(y, x))$
- $\nu = \{x \leftarrow y, y \leftarrow y\}$
  - $\nu(t) = y + (y \star y)$
  - $\nu(t') = y + (y \star y)$
  - ν este cgu

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{0 :\to s,\ + : ss \to s,\ \star : ss \to s\}$
- $X = \{x, y\}$
- $t = x + (y \star y) = +(x, \star(y, y))$
- $t' = x + (y \star x) = +(x, \star(y, x))$
- $\nu = \{x \leftarrow y, y \leftarrow y\}$
  - $\nu(t) = y + (y \star y)$
  - $\nu(t') = y + (y \star y)$
  - ν este cgu
- $\nu' = \{x \leftarrow 0, y \leftarrow 0\}$
  - $\nu'(t) = 0 + (0 \star 0)$
  - $\nu'(t') = 0 + (0 \star 0)$

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{0 :\to s,\ + : ss \to s,\ \star : ss \to s\}$
- $X = \{x, y\}$
- $t = x + (y \star y) = +(x, \star(y, y))$
- $t' = x + (y \star x) = +(x, \star(y, x))$
- $\nu = \{x \leftarrow y, y \leftarrow y\}$
  - $\nu(t) = y + (y \star y)$
  - $\nu(t') = y + (y \star y)$
  - ν este cgu
- $\nu' = \{x \leftarrow 0, y \leftarrow 0\}$
  - $\nu'(t) = 0 + (0 \star 0)$
  - $\nu'(t') = 0 + (0 \star 0)$
  - $\nu' = \nu; \{y \leftarrow 0\}$

Exemplu

Exemplu

- $S = \{s\}$ și $\Sigma = \{0 :\to s,\ + : ss \to s,\ \star : ss \to s\}$
- $X = \{x, y\}$
- $t = x + (y \star y) = +(x, \star(y, y))$
- $t' = x + (y \star x) = +(x, \star(y, x))$
- $\nu = \{x \leftarrow y, y \leftarrow y\}$
  - $\nu(t) = y + (y \star y)$
  - $\nu(t') = y + (y \star y)$
  - ν este cgu
- $\nu' = \{x \leftarrow 0, y \leftarrow 0\}$
  - $\nu'(t) = 0 + (0 \star 0)$
  - $\nu'(t') = 0 + (0 \star 0)$
  - $\nu' = \nu; \{y \leftarrow 0\}$
  - ν' este unificator, dar nu este gcu

Algoritmul de unificare

- Pentru o mulțime finită de ecuații $U = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$ 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 ecuații $U = \{t_1 \doteq t_1', \ldots, t_n \doteq t_n'\}$ 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 ecuații $U = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$ 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: $R = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$

Algoritmul de unificare

Algoritmul constă în aplicarea nedeterministă a regulilor de mai jos:

Algoritmul de unificare

Algoritmul constă în aplicarea nedeterministă a regulilor de mai jos:

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

Algoritmul de unificare

Algoritmul constă în aplicarea nedeterministă a 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 nedeterministă a 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 dă 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
	$x \doteq t$, $S[x \leftarrow t]$	$R'[x \leftarrow t]$
Final	S	$\emptyset$

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

Exemplu

Exemplu

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

Exemplu

Exemplu

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

Exemplu

Exemplu

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

Exemplu

Exemplu

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ are gcu?

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

- $\nu = \{y \leftarrow z,\ x \leftarrow g(z),\ w \leftarrow h(g(z))\}$ este cgu pentru ecuațiile din U.

Exemplu

Exemplu

- $S = \{s\}$, $\Sigma = \{b :\to s,\ g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z\}$
- $U = \{g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)\}$ are gcu?

Exemplu

Exemplu

- $S = \{s\}$, $\Sigma = \{b :\to s,\ g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z\}$
- $U = \{g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)\}$ are gcu?

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 ecuațiile din U.

Exemplu

Exemplu

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)\}$ are gcu?

Exemplu

Exemplu

- $S = \{s\}$, $\Sigma = \{g : s \to s,\ h : s \to s,\ f : sss \to s\}$, $X = \{x, y, z, w\}$
- $U = \{g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)\}$ are gcu?

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 ecuațiile din U.

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

$\square$

Corectitudinea algoritmului

Lema 1

Mulțimea unificatorilor pentru reuniunea ecuațiilor din R și S nu se modifică prin aplicarea celor trei reguli ale algoritmului de unificare.

Corectitudinea algoritmului

Lema 1

Mulțimea unificatorilor pentru reuniunea ecuațiilor din R și 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 reuniunea ecuațiilor din R și 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 reuniunea ecuațiilor din R și 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 or. unificator ν pt. reuniunea ecuațiile din R și S, atât înainte cât și după aplicarea regulii REZOLVĂ, trebuie să satisfacă:
    $$\nu(x) = \nu(t).$$
  - Pt. or. unificator μ pt. $x \doteq t$ observăm că:
    $$(x \leftarrow t); \mu = \mu$$
    - $((x \leftarrow t); \mu)(x) = \mu(t) = \mu(x)$
    - $((x \leftarrow t); \mu)(y) = \mu(y)$, or. $y \neq x$
  - Deci,

$$\mu \text{ este un unificator pt. ec. din } R \text{ și } S \text{ înainte de REZOLVĂ}$$
$$\Leftrightarrow$$
$$\mu \text{ este un unificator pt. ec. din } R \text{ și } S \text{ după REZOLVĂ}$$

$\square$

Corectitudinea algoritmului

Lema 2

Variabilele care apar în partea stângă a ecuațiilor din S sunt distincte două câte două și nu mai apar în altă parte în S și R.

Corectitudinea algoritmului

Lema 2

Variabilele care apar în partea stângă a ecuațiilor din S sunt distincte două câte două și nu mai apar în altă parte în S și R.

Demonstrație

Exercițiu!

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.
- Definim substituția:

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

- ν este corect definită (vezi Lema 2).
- Cum variabilele x_i nu apar în termenii t_i, deducem că $\nu(t_i) = t_i = \nu(x_i)$, or. $i = 1, \ldots, k$.
- Deci ν este unificator pentru U (vezi Lema 1).

Corectitudinea algoritmului

Lema 3

ν definit mai sus cf. algoritmului de unificare este cgu pentru U.

Corectitudinea algoritmului

Lema 3

ν definit mai sus cf. algoritmului de unificare este cgu pentru U.

Demonstrație

- Pt. or. substituție s obținem că $\nu; s$ este unificator pt. U: pentru $t_i \doteq t'_i \in U$, avem
$$(\nu; s)(t_i) = s(\nu(t_i)) = s(\nu(t'_i)) = (\nu; s)(t'_i)$$
- Fie μ un alt unificator pt. U. Avem
  - $\mu(\nu(x_i)) = \mu(t_i) = x_i$, or. $i = 1, \ldots, k$,
  - $\mu(\nu(y)) = \mu(y)$, or. $y \neq x$.

  Deci $\nu; \mu = \mu$.
- În concluzie, ν este cgu deoarece or. alt unificator se poate scrie ca o compunere a lui ν cu o substituție.

□

Matching problem

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

  $$p \text{ matches } t \ (t \text{ este o instanță a lui } p)$$

  dacă există o substituție $\nu : X \to T_\Sigma(X)$ astfel încât

  $$\nu(p) = t.$$

Matching problem

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

  $$p \text{ matches } t \ (t \text{ este o instanță a lui } p)$$

  dacă există o substituție $\nu : X \to T_\Sigma(X)$ astfel încât

  $$\nu(p) = t.$$

Exemplu

- $p = x + (y \star y)$
- $t = (a + y) + (x \star x)$

Matching problem

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

  $$p \text{ matches } t \ (t \text{ este o instanță a lui } p)$$

  dacă există o substituție $\nu : X \to T_\Sigma(X)$ astfel încât

  $$\nu(p) = t.$$

Exemplu

- $p = x + (y \star y)$
- $t = (a + y) + (x \star x)$
- $\nu = \{x \leftarrow a + y,\ y \leftarrow x\}$
- $\nu(p) = (a + y) + (x \star x) = t$

Matching problem

- Fie t' termenul obținut din t prin înlocuirea fiecărei variabile x cu un simbol de constantă c_x

Matching problem

- Fie t' termenul obținut din t prin înlocuirea fiecărei variabile x cu un simbol de constantă c_x
- Substituția ν poate fi determinată aplicând algoritmul de unificare pentru ecuația $p \doteq t'$.

Matching problem

- Fie t' termenul obținut din t prin înlocuirea fiecărei variabile x cu un simbol de constantă c_x
- Substituția ν poate fi determinată aplicând algoritmul de unificare pentru ecuația $p \doteq t'$.

Exemplu

- $p = x + (y \star y)$
- $t = (a + y) + (x \star x)$

Matching problem

- Fie t' termenul obținut din t prin înlocuirea fiecărei variabile x cu un simbol de constantă c_x
- Substituția ν poate fi determinată aplicând algoritmul de unificare pentru ecuația $p \doteq t'$.

Exemplu

- $p = x + (y \star y)$
- $t = (a + y) + (x \star x)$
- $t' = (a + c_y) + (c_x \star c_x)$

Matching problem

- Fie t' termenul obținut din t prin înlocuirea fiecărei variabile x cu un simbol de constantă c_x
- Substituția ν poate fi determinată aplicând algoritmul de unificare pentru ecuația $p \doteq t'$.

Exemplu

- $p = x + (y \star y)$
- $t = (a + y) + (x \star x)$
- $t' = (a + c_y) + (c_x \star c_x)$
- Aplicăm algoritmul de unificare pt. $\{x + (y \star y) \doteq (a + c_y) + (c_x \star c_x)\}$ și obținem $\{x \doteq a + c_y,\ y \doteq c_x\}$.

Matching problem

- Fie t' termenul obținut din t prin înlocuirea fiecărei variabile x cu un simbol de constantă c_x
- Substituția ν poate fi determinată aplicând algoritmul de unificare pentru ecuația $p \doteq t'$.

Exemplu

- $p = x + (y \star y)$
- $t = (a + y) + (x \star x)$
- $t' = (a + c_y) + (c_x \star c_x)$
- Aplicăm algoritmul de unificare pt. $\{x + (y \star y) \doteq (a + c_y) + (c_x \star c_x)\}$ și obținem $\{x \doteq a + c_y,\ y \doteq c_x\}$.

- O problemă de matching poate fi rezolvată prin unificare.

Pe săptămâna viitoare!