Examen Sesiune

Problema 2: Algoritmul de unificare

Primul set de ecuații: {f(x, g(y)) = f(h(z), g(h(a))), y = h(z)}

Pasul 1: Aplicăm descompunerea pentru f(x, g(y)) = f(h(z), g(h(a)))

- Obținem două ecuații: x = h(z) și g(y) = g(h(a))
- Mulțimea nouă: {x = h(z), g(y) = g(h(a)), y = h(z)}

Pasul 2: Aplicăm descompunerea pentru g(y) = g(h(a))

- Obținem: y = h(a)
- Mulțimea nouă: {x = h(z), y = h(a), y = h(z)}

Pasul 3: Avem un conflict: y = h(a) și y = h(z)

- Pentru consistență, trebuie h(a) = h(z)
- Aplicând descompunerea, obținem: a = z
- Mulțimea nouă: {x = h(z), y = h(a), z = a}

Pasul 4: Substituim z = a în x = h(z)

- Obținem: x = h(a)
- Mulțimea finală: {x = h(a), y = h(a), z = a}

Cel mai general unificator este: {x ↦ h(a), y ↦ h(a), z ↦ a}

Al doilea set de ecuații: {f(z, a) = f(g(y), y), y = f(x, x)}

Pasul 1: Aplicăm descompunerea pentru f(z, a) = f(g(y), y)

- Obținem: z = g(y) și a = y
- Mulțimea nouă: {z = g(y), a = y, y = f(x, x)}

Pasul 2: Substituim a = y în y = f(x, x)

- Obținem: a = f(x, x)
- Mulțimea nouă: {z = g(y), a = f(x, x), y = a}

Pasul 3: Substituim y = a în z = g(y)

- Obținem: z = g(a)
- Mulțimea nouă: {z = g(a), a = f(x, x), y = a}

Pasul 4: Substituim a = f(x, x) în z = g(a)

- Obținem: z = g(f(x, x))
- Mulțimea finală: {z = g(f(x, x)), a = f(x, x), y = f(x, x)}

Cel mai general unificator este: {z ↦ g(f(x, x)), a ↦ f(x, x), y ↦ f(x, x)}

Problema 3: SLD-respingere pentru program Prolog

Program:

```
p(X, f(Y)) :- q(X), q(Y).
q(f(X)) :- r(X).
r(a).
```

Ținta: p(Y, X)

Pasul 1: Rezolvăm p(Y, X)

- Unificăm cu p(X, f(Y)) din prima clauză
- Substituție: $\theta_1 = \{Y \mapsto X_1, X \mapsto f(Y_1)\}$
- Noua țintă: $q(X_1), q(Y_1)$

Pasul 2: Rezolvăm $q(X_1)$

- Unificăm cu q(f(X)) din a doua clauză
- Substituție: $\theta_2 = \{X_1 \mapsto f(X_2)\}$
- Noua țintă: $r(X_2), q(Y_1)$

Pasul 3: Rezolvăm $r(X_2)$

- Unificăm cu r(a)
- Substituție: $\theta_3 = \{X_2 \mapsto a\}$
- Noua țintă: $q(Y_1)$

Pasul 4: Rezolvăm $q(Y_1)$

- Unificăm cu q(f(X)) din a doua clauză
- Substituție: $\theta_4 = \{Y_1 \mapsto f(X_3)\}$
- Noua țintă: $r(X_3)$

Pasul 5: Rezolvăm $r(X_3)$

- Unificăm cu r(a)
- Substituție: $\theta_5 = \{X_3 \mapsto a\}$
- Țintă vidă: succes!

Compunând toate substituțiile:

- $Y = X_1 = f(X_2) = f(a)$
- $X = f(Y_1) = f(f(X_3)) = f(f(a))$

Valorile variabilelor din țintă în substituția calculată:

- $Y = f(a)$
- $X = f(f(a))$

Problema 4: Tiparea λ-termenilor

Termenul 1: x(λz.(y z))

Pentru a determina dacă acest termen are tip:

1. Pentru subtermenul (y z):
   - Presupunem că z are tipul τ_1
   - Pentru ca aplicația să fie validă, y trebuie să aibă tipul $\tau_1 \to \tau_2$
   - Astfel, (y z) are tipul τ_2
2. Pentru λz.(y z):
   - Abstractizarea are tipul $\tau_1 \to \tau_2$
3. Pentru întregul termen x(λz.(y z)):
   - Pentru ca aplicația să fie validă, x trebuie să aibă tipul $(\tau_1 \to \tau_2) \to \tau_3$
   - Întregul termen are tipul τ_3

Concluzie: Termenul are tip. Putem construi un context de tipare Γ unde:

- $\Gamma(x) = (\tau_1 \to \tau_2) \to \tau_3$
- $\Gamma(y) = \tau_1 \to \tau_2$ și termenul are tipul τ_3.

Termenul 2: a(x y)(x(y z))

Pentru a determina dacă acest termen are tip:

1. Pentru subtermenul (x y):
   - x trebuie să aibă tipul $\sigma_1 \to \sigma_2$
   - y trebuie să aibă tipul σ_1
   - (x y) are tipul σ_2
2. Pentru subtermenul (y z):
   - y trebuie să aibă tipul $\tau_1 \to \tau_2$
   - z trebuie să aibă tipul τ_1
   - (y z) are tipul τ_2
3. Pentru subtermenul x(y z):
   - x trebuie să aibă tipul $\tau_2 \to \tau_3$
   - x(y z) are tipul τ_3
4. Pentru întregul termen a(x y)(x(y z)):
   - a trebuie să aibă tipul $\sigma_2 \to (\tau_3 \to \rho)$
   - Întregul termen are tipul ρ

Problema: Variabila x apare în două contexte diferite, necesitând tipuri diferite:

- În (x y), x are tipul $\sigma_1 \to \sigma_2$
- În x(y z), x are tipul $\tau_2 \to \tau_3$

De asemenea, y apare cu două tipuri diferite:

- În (x y), y are tipul σ_1
- În (y z), y are tipul $\tau_1 \to \tau_2$

Aceste constrângeri conduc la o ecuație recursivă de tip care nu poate fi satisfăcută în calculul lambda tipat simplu.

Concluzie: Termenul nu poate avea tip în sistemul de tipuri lambda tipat simplu.

Problema 5:
Demonstrarea echivalenței dintre semanticile operaționale big-step și small-step pentru instrucțiunea if

În demonstrarea echivalenței dintre semanticile operaționale big-step și small-step, trebuie să arătăm că instrucțiunea `if` are același comportament semantic în ambele abordări. Voi prezenta această demonstrație pas cu pas.

Notații preliminare:

- $\langle S, \sigma \rangle \Downarrow \sigma'$ - notația pentru semantica big-step (evaluarea programului S în starea σ produce starea finală σ')
- $\langle S, \sigma \rangle \to \langle S', \sigma' \rangle$ - notația pentru un pas în semantica small-step
- $\langle S, \sigma \rangle \to^* \langle S', \sigma' \rangle$ - tranzitiva reflexivă a relației → (zero sau mai multe pași)
- $B[b]\sigma$ - evaluarea expresiei booleene b în starea σ (rezultă true sau false)

Regulile semantice pentru instrucțiunea if:

Big-step:

1. Regula pentru ramura true:

$$\frac{B[b]\sigma = \text{true} \quad \langle S_1, \sigma \rangle \Downarrow \sigma'}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \Downarrow \sigma'}$$

2. Regula pentru ramura false:

$$\frac{B[b]\sigma = \text{false} \quad \langle S_2, \sigma \rangle \Downarrow \sigma'}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \Downarrow \sigma'}$$

Small-step:

1. Evaluarea condiției:

$$\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to \langle S_1, \sigma \rangle \quad \text{dacă } B[b]\sigma = \text{true}$$
$$\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to \langle S_2, \sigma \rangle \quad \text{dacă } B[b]\sigma = \text{false}$$

Demonstrație:

Parte 1: Implicația de la small-step la big-step

Trebuie să arătăm că dacă $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$, atunci $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \Downarrow \sigma'$.

Cazul 1: Dacă $B[b]\sigma = \text{true}$:

1. Primul pas în semantica small-step: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to \langle S_1, \sigma \rangle$
2. Apoi avem: $\langle S_1, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$
3. Prin ipoteza de inducție, știm că $\langle S_1, \sigma \rangle \Downarrow \sigma'$
4. Aplicând regula big-step pentru if (ramura true), obținem: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \Downarrow \sigma'$

Cazul 2: Dacă $B[b]\sigma = \text{false}$:

1. Primul pas în semantica small-step: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to \langle S_2, \sigma \rangle$
2. Apoi avem: $\langle S_2, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$
3. Prin ipoteza de inducție, știm că $\langle S_2, \sigma \rangle \Downarrow \sigma'$
4. Aplicând regula big-step pentru if (ramura false), obținem: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \Downarrow \sigma'$

Parte 2: Implicația de la big-step la small-step

Trebuie să arătăm că dacă $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \Downarrow \sigma'$, atunci $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$.

Cazul 1: Dacă derivarea în big-step folosește regula pentru ramura true:

1. Știm că $B[b]\sigma = \text{true}$ și $\langle S_1, \sigma \rangle \Downarrow \sigma'$
2. Prin ipoteza de inducție, avem $\langle S_1, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$
3. În semantica small-step, primul pas este: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to \langle S_1, \sigma \rangle$
4. Combinând cu punctul 2, obținem: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to \langle S_1, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$
5. Deci: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$

Cazul 2: Dacă derivarea în big-step folosește regula pentru ramura false:

1. Știm că $B[b]\sigma = \text{false}$ și $\langle S_2, \sigma \rangle \Downarrow \sigma'$
2. Prin ipoteza de inducție, avem $\langle S_2, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$
3. În semantica small-step, primul pas este: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to \langle S_2, \sigma \rangle$
4. Combinând cu punctul 2, obținem: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to \langle S_2, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$
5. Deci: $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, \sigma \rangle \to^* \langle \text{skip}, \sigma' \rangle$

Concluzie:

Am demonstrat echivalența dintre semantica big-step și semantica small-step pentru instrucțiunea if, în ambele direcții:

- Dacă o evaluare în semantica small-step produce un rezultat, același rezultat este obținut și în semantica big-step.
- Dacă o evaluare în semantica big-step produce un rezultat, același rezultat este obținut și în semantica small-step, prin aplicarea unei secvențe corespunzătoare de pași.

Această demonstrație completează cazul instrucțiunii if care a fost lăsat ca exercițiu în curs.