PROGRAMARE LOGICĂ ECUAŢIONALĂ
ediţia a III-a

Virgil Emil Căzănescu

October 4, 2008

Chapter 1

IN PRIMUL RAND SEMANTICA

Această variantă este independentă de lecţiile privind rescrierea rămâmând necesare cunoştiinţele privind algebrele multisortate. Menţionăm că cei doi membri ai unei egalităţi trebuie să aibă acelaş sort.

1.1 Introducere

Pentru orice signatură multisortată (S, Σ) şi orice mulţime de variabile X notăm Σ-algebra liber generată de X cu $T_\Sigma(X)$.

Definiţia 1.1.1 O *ecuaţie condiţionată* este

$$(\forall X) l \doteq_s r \textbf{ if } H$$

unde X este o mulţime S-sortată de variabile, l şi r sunt două elemente de sort s din $T_\Sigma(X)$ iar H o mulţime finită de egalităţi formale din $T_\Sigma(X) \times T_\Sigma(X)$. $\Box$

O ecuaţie condiţionată în care $H = \emptyset$ devine necondiţionată şi este numită pe scurt *ecuaţie*. În acest caz scriem doar $(\forall X) l \doteq_s r$ în loc de $(\forall X) l \doteq_s r \textbf{ if } \emptyset$.

Definiţia 1.1.2 Algebra $\mathcal{D}$ *satisface* ecuaţia condiţionată $(\forall X) l \doteq_s r \textbf{ if } H$, fapt notat prin

$$\mathcal{D} \models_\Sigma (\forall X) l \doteq_s r \textbf{ if } H$$

dacă pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ pentru care $h_{s'}(u) = h_{s'}(v)$ pentru orice $u \doteq_{s'} v \in H$, avem $h_s(l) = h_s(r)$. $\Box$

În cele ce urmează indicele Σ din $\models_\Sigma$ va fi omis. El va fi menţionat atunci când este pericol de confuzie.
Observăm că $\mathcal{D} \models (\forall X) l \doteq_s r$ dacă şi numai dacă $h_s(l) = h_s(r)$ pentru orice morfism $h\colon T_\Sigma(X) \longrightarrow \mathcal{D}$.

În continuare fixăm o mulţime Γ de ecuaţii condiţionate, numite axiome sau clauze.

Definiţia 1.1.3 Spunem că algebra $\mathcal{D}$ satisface Γ sau că $\mathcal{D}$ e o Γ-algebră şi scriem $\mathcal{D} \models \Gamma$ dacă $\mathcal{D}$ satisface toate ecuaţiile condiţionate din Γ. $\Box$

1.2 Semantică şi Corectitudine

Vom lucra într-o Σ-algebră $\mathcal{A}$.
Fie $\equiv_\Gamma$ relaţia pe A definită prin

$$a \equiv_\Gamma c \text{ dacă şi numai dacă } (\forall h : \mathcal{A} \longrightarrow \mathcal{M} \models \Gamma) h_s(a) = h_s(c).$$

Mai observăm că

$$\equiv_\Gamma = \bigcap \{Ker(h) \mid h : \mathcal{A} \longrightarrow \mathcal{M} \models \Gamma\}.$$

Deoarece nucleul unui morfism este o relaţie de congruenţă şi deoarece orice intersecţie de relaţii de congruenţe este o relaţie de congruenţă, deducem că $\equiv_\Gamma$ este o relaţie de congruenţă.

Congruenţa $\equiv_\Gamma$ este numită **congruenţă semantică.**

Dăm două reguli de deducţie ale rescrierii $\mathbf{Sub}_\Gamma$ şi $\mathbf{Rew}_\Gamma$.

$\mathbf{Sub}_\Gamma$ Pentru orice $(\forall X)\, l \doteq_s r$ **if** $H \in \Gamma$ şi orice morfism $h\colon T_\Sigma(X) \longrightarrow \mathcal{A}$
$(\forall u \doteq_{s'} v \in H) h_{s'}(u) \doteq_{s'} h_{s'}(v)$ implică $h_s(l) \doteq_s h_s(r)$.

$\mathbf{Rew}_\Gamma$ Pentru orice $(\forall X)\, l \doteq_s r$ **if** $H \in \Gamma$ şi orice morfism $h\colon T_\Sigma(X) \longrightarrow \mathcal{A}$
$(\forall u \doteq_{s'} v \in H)(\exists d \in A_{s'}) h_{s'}(u) \doteq_{s'} d$ şi $h_{s'}(v) \doteq_{s'} d$ implică $h_s(l) \doteq_s h_s(r)$.

Lemă 1.2.1 *Pentru orice morfism $f : \mathcal{A} \longrightarrow \mathcal{M} \models \Gamma$, $Ker(f)$ este închis la* $\mathbf{Rew}_\Gamma$*.*

Demonstraţie: Fie $(\forall X) l \doteq_s r$ **if** H în Γ şi $h : T_\Sigma(X) \longrightarrow \mathcal{A}$ un morfism cu proprietatea că pentru orice $u \doteq v \in H$ există c_{uv} cu proprietăţile $h(u)\ Ker(f)\ c_{uv}$ şi $h(v)\ Ker(f)\ c_{uv}$. Prin urmare $(h; f)(u) = f(c_{uv}) = (h; f)(v)$ pentru orice $u \doteq v \in H$. Deoarece $h; f\colon T_\Sigma(X) \longrightarrow \mathcal{M} \models \Gamma$ rezultă că $(h; f)_s(l) = (h; f)_s(r)$, deci $h_s(l)\ Ker(f)\ h_s(r)$.$\Box$

Deoarece orice relaţie reflexivă şi închisă la $\mathbf{Rew}_\Gamma$ este închisă şi la $\mathbf{Sub}_\Gamma$ obţinem lema următoare.

Lemă 1.2.2 *Pentru orice morfism $f : \mathcal{A} \longrightarrow \mathcal{M} \models \Gamma$, $Ker(f)$ este închis la* $\mathbf{Sub}_\Gamma$*.*

Propoziţie 1.2.3 *Congruenţă semantică este închisă la substituţie.*

Demonstraţie: Congruenţa semantică este o intersecţie de congruenţe închise la substituţie. Deoarece o intersecţie de congruenţe închise la substituţii este o congruenţă închisă la substituţii rezultă că $\equiv_\Gamma$ este o congruenţă închisă la substituţie. $\Box$

Propoziţie 1.2.4 *Dacă $\sim$ este o congruenţă închisă la substituţii, atunci $\mathcal{A}/\!\sim\, \models \Gamma$.*

Demonstraţie: Notăm cu $\rho : \mathcal{A} \longrightarrow \mathcal{A}/\!\sim$ morfismul de factorizare canonic.

Fie $(\forall X) l \doteq_s r$ **if** H în Γ şi $h : T_\Sigma(X) \longrightarrow \mathcal{A}/\!\sim$ un morfism astfel încât $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$. Cum $T_\Sigma(X)$ este algebră proiectivă există un morfism $f : T_\Sigma(X) \longrightarrow \mathcal{A}$ astfel încât $f; \rho = h$. Pentru orice $u \doteq_t v \in H$ deoarece $\rho_t(f_t(u)) = \rho_t(f_t(v))$ deducem $f_t(u) \sim f_t(v)$.

Deoarece $\sim$ este o congruenţă închisă la substituţii obţinem $f_s(l) \sim f_s(r)$. Prin urmare $\rho_s(f_s(l)) = \rho_s(f_s(r))$, de unde $h_s(l) = h_s(r)$. $\Box$

Fie $\mathcal{A}_\Gamma$ factorizarea lui $\mathcal{A}$ prin congruenţa $\equiv_\Gamma$ şi fie $\eta\colon \mathcal{A} \longrightarrow \mathcal{A}_\Gamma$ morfismul cât.

Teorema 1.2.5 $\mathcal{A}_\Gamma \models \Gamma$

Demonstraţie: Se aplică propoziţiile 1.2.3 şi 1.2.4. $\Box$

Teorema 1.2.6 *Pentru orice Γ-algebră $\mathcal{B}$ şi pentru orice morfism $h\colon \mathcal{A} \longrightarrow \mathcal{B}$ există şi este unic un morfism $h^\#\colon \mathcal{A}_\Gamma \longrightarrow \mathcal{B}$ astfel încât $\eta; h^\# = h$.*

Demonstraţie: Este suficient să arătăm că $a \equiv_\Gamma c$ implică $h(a) = h(c)$ şi apoi să aplicăm proprietatea de universalitate a algebrei cât. Într-adevăr $a \equiv_\Gamma c$ implică $h(a) = h(c)$ deoarece $h\colon \mathcal{A} \longrightarrow \mathcal{B} \models \Gamma$. $\Box$

Corolar 1.2.7 *Dacă $\mathcal{A}$ este Σ-algebră iniţială, atunci $\mathcal{A}_\Gamma$ este Γ-algebră iniţială.*

Demonstraţie: Fie $\mathcal{B}$ o Γ-algebră arbitrară.
Cum $\mathcal{A}$ este Σ-algebră iniţială rezultă că există un unic Σ-morfism $h : \mathcal{A} \to \mathcal{B}$.
Conform teoremei 1.2.6 există un unic Σ-morfism $h^\# : \mathcal{A}_\Gamma \to \mathcal{B}$ astfel încât $\eta; h^\# = h$. Arătăm că $h^\#$ este unicul morfism de Γ-algebre de la $\mathcal{A}_\Gamma$ la $\mathcal{B}$.
Dacă $f : \mathcal{A}_\Gamma \to \mathcal{B}$ este un alt morfism atunci $\eta; f$ este morfism de la $\mathcal{A}$ la $\mathcal{B}$, rezultând că $\eta; f = h$, deci $f = h^\#$.

Chapter 2

UNIFICAREA

Toate algebrele sunt presupuse libere. Prin substituţie vom înţelege un morfism între două algebre libere.

Problema unificării. Se dau un număr finit de perechi de termeni $l_i \doteq_{s_i} r_i$ şi se cere găsirea unui unificator, adică a unei substituţii u cu proprietatea $u(l_i) = u(r_i)$ pentru orice i.

2.1 Algoritmul de unificare

Vom lucra cu două liste: soluţie şi de rezolvat. Initial lista soluţie este vidă şi lista de rezolvat conţine mulţimea ecuaţiilor de unificat.

Algoritmul de unificare constă în execuţia nedeterministă a urmatorilor trei paşi.

1. **Scoate.** Se scoate din lista de rezolvat orice ecuaţie de forma $t \doteq t$.
2. **Descompune.** Orice ecuaţie din lista de rezolvat de forma

$$f(a_1, a_2, \ldots, a_n) \doteq f(b_1, b_2, \ldots, b_n)$$

   se înlocuieşte cu ecuaţiile $a_1 \doteq b_1$, $a_2 \doteq b_2$, ..., $a_n \doteq b_n$,
3. **Elimină.** Orice ecuaţie din lista de rezolvat, de forma $x \doteq t$ sau $t \doteq x$ în care x este o variabilă care nu apare în t este mutată sub forma $x \doteq t$ în lista soluţie. In toate celelalte ecuaţii x se substitue cu t.

Algoritmul este oprit cu concluzia inexistenţei unui unificator în urmatoarele două cazuri.
1) Existenţa în lista de rezolvat a unei ecuaţii de forma $f(a_1, a_2, \ldots, a_n) \doteq g(b_1, b_2, \ldots, b_m)$ cu $f \neq g$,
2) Existenţa în lista de rezolvat a unei ecuaţii de forma $x \doteq t$ sau $t \doteq x$ în care x este o variabilă care apare în t şi este diferită de t.

Oprirea normală a algoritmului se face prin epuizarea listei de rezolvat, caz în care, aşa cum vom proba mai jos, lista soluţie coincide cu cel mai general unificator.

2.2 Terminare

Terminarea algoritmului este probată folosind în ordine lexicografică două criterii exprimate prin numere naturale:

1. numărul variabilelor care apar în lista de rezolvat, care in funcţie de pasul algoritmului utilizat are următoarea comportare
   (a) **Scoate**: rămâne egal sau se micşorează,
   (b) **Descompune**: rămâne egal,
   (c) **Elimină**: se micşorează
2. numărul apariţiilor simbolurilor(semnelor) care apar în lista de rezolvat, care se micşorează în cele două cazuri care ne mai interesează **Scoate** şi **Descompune**.

2.3 Corectitudine

Corectitudinea algoritmului se bazează pe demonstrarea faptului că mulţimea unificatorilor pentru ecuaţiile din reuniunea celor două liste este un invariant, adică nu se modifică prin aplicarea celor trei paşi ai algoritmului.

Deoarece pentru pasul **Scoate** afirmaţia este evidentă ne referim doar la ceilalţi paşi.

Descompune: Observăm că pentru orice substituţie s egalitatea

$$s(f(a_1, a_2, \ldots, a_n)) = s(f(b_1, b_2, \ldots, b_n))$$

este echivalentă cu

$$f(s(a_1), s(a_2), \ldots, s(a_n)) = f(s(b_1), s(b_2), \ldots, s(b_n))$$

adică cu $s(a_i) = s(b_i)$ pentru orice $i \in [n]$.

Elimină: Observăm că orice unificator u pentru ecuaţiile din reuniunea celor două liste atât înainte de aplicarea pasului cât şi după aceasta trebuie să satisfacă egalitatea $u(x) = u(t)$. Pentru o substituţie s cu proprietatea $s(x) = s(t)$ observăm că

$$x \leftarrow t; s = s$$

deoarece $(x \leftarrow t; s)(x) = s(t) = s(x)$ şi $(x \leftarrow t; s)(y) = s(y)$ pentru orice alta variabilă y. Prin urmare pentru o astfel de substituţie

$$s(l) = s(r) \text{ dacă şi numai dacă } s((x \leftarrow t)(l)) = s((x \leftarrow t)(r))$$

ceea ce arată că un unificator pentru ecuaţiile din reuniunea celor două liste dinainte de aplicarea pasului este unificator şi pentru ecuaţiile din reuniunea celor două liste de după aplicarea pasului şi reciproc.

Algoritmul este oprit cu concluzia inexistenţei unui unificator deoarece în cele două situaţii menţionate se constată că mulţimea unificatorilor este vidă.

Să observăm că variabilele care apar în membrul stâng al ecuaţiilor din lista soluţie sunt diferite două câte două şi nu mai apar în nici una dintre celelalte ecuaţii din cele două liste.

Faptul poate fi dovedit prin inducţie.

Menţionăm că aplicarea primilor doi paşi ai algoritmului nu modifică lista soluţie şi nu produc apariţii noi de variabile în cele două liste.

Fie $x \doteq t$ ecuaţia introdusă în lista soluţie prin aplicarea pasului **Elimină**. Deoarece variabilele din membrul stâng ai listei soluţie precedentă nu apar în celelalte ecuaţii rezultă că variabila x este diferită de celelalte variabile care apar în membrul stâng al ecuaţiilor din lista soluţie. In plus prin substituirea lui x cu t în celelalte ecuaţii variabila x dispare din ele deoarece x nu apare în t. Deoarece nici x şi nici variabilele din membrul stâng ai listei soluţie precedentă nu apar în t rezultă că după efectuarea substituţiei lui x cu t variabilele din membrul stâng ai listei soluţie nu apar în restul ecuaţiilor.

Să presupunem că algoritmul s-a terminat prin epuizarea listei de rezolvat. Să notăm cu k numărul ecuaţiilor din lista soluţie şi cu $x_i \doteq t_i$ pentru $i \in [k]$ ecuaţiile din ea.

Fie U substituţia definită prin

$$U(x_i) = t_i \text{ pentru orice } i \in [k].$$

Definiţia este corectă deoarece variabilele x_i sunt distincte. Deoarece variabilele x_i nu apar în termenii t_i deducem că $U(t_i) = t_i$, prin urmare $U(t_i) = U(x_i)$, deci U este un unificator. Vom dovedi că este cel mai general.

Observăm că pentru orice substituţie s compunerea $U; s$ este tot un unificator. Vom arăta că orice alt unificator este de această formă. Fie u un alt unificator, adica $u(x_i) = u(t_i)$ pentru orice $i \in [k]$. Observăm că $U; u = u$, căci $u(U(x_i)) = u(t_i) = u(x_i)$ pentru orice $i \in [k]$ şi $u(U(y)) = u(y)$ pentru orice altă variabilă y.

Deci U este cel mai general unificator, deoarece orice alt unificator poate fi exprimat ca o compunere a lui U cu o substituţie. In plus observăm că el este idempotent deoarece $U; U = U$.

În cele ce urmează vom scrie CGU ca o prescurtare pentru Cel mai General Unificator.

In semantica operaţională a limbajelor declarative rescrierea joacă un rol primordial. Pentru a rescrie un element a cu ajutorul unei reguli $l \doteq_s r$ se pune problema identificării unui subtermen al lui a cu imaginea lui l printr-o substituţie. Această problemă de potrivire(matching în engleză) se poate rezolva tot cu ajutorul algoritmului de unificare. Mai precis se caută a se unifica l cu un subtermen al lui a în care toate variabilele sunt considerate constante.

Chapter 3

TEOREMELE LUI HERBRAND

3.1 Introducere

Reamintim următoarele definiţii specifice logicii ecuaţionale multisortate.

Definiţia 3.1.1 *Fie* Σ*-algebra* $T_\Sigma(X)$ *şi* $G = \{l_1 \doteq_{s_1} r_1, \ldots, l_n \doteq_{s_n} r_n\}$ *o mulţime de egalităţi formale între elemente din* $T_\Sigma(X)$*. Atunci* $\gamma = (\forall X) l \doteq_s r$ *if* G *se numeşte clauză. Notăm cu* Γ *o mulţime de clauze.*

Notaţii 3.1.2 *Fie* $\mathcal{A}$ *o* Σ*-algebră şi* γ *o clauză ca mai sus. Notăm* $\mathcal{A} \models \gamma$ *dacă pentru orice morfism* $h : T_\Sigma(X) \to \mathcal{A}$ *astfel încât* $h(l_i) = h(r_i)$ *pentru orice* $i \in [n]$ *avem* $h(l) = h(r)$*. Similar* $\mathcal{A} \models \Gamma$ *dacă* $\mathcal{A} \models \gamma$ *pentru orice* $\gamma \in \Gamma$*. În acest caz spunem că* $\mathcal{A}$ *este o* Γ*-algebră.*

Notaţii 3.1.3 *Notăm cu* $T_\Sigma = T_\Sigma(\emptyset)$ Σ*-algebra iniţială (adică obiectul iniţial în categoria* Σ*-algebrelor) şi cu* $T_{\Sigma,\Gamma}$ Γ*-algebra iniţială (adică obiectul iniţial în categoria* Γ*-algebrelor, subcategorie a categoriei* Σ*-algebrelor).*

Introducem o nouă definiţie şi o nouă notaţie pentru programarea logică ecuaţională.

Definiţia 3.1.4 *Fie* $\mathcal{A}$ *o* Σ*-algebră şi* $G = \{l_1 \doteq_{s_1} r_1, \ldots, l_n \doteq_{s_n} r_n\}$ *o mulţime de egalităţi formale între elemente din* $T_\Sigma(X)$*.* $\mathcal{A} \models (\exists X)G$ *dacă există morfismul* $h : T_\Sigma(X) \to \mathcal{A}$ *astfel încât* $h(l_i) = h(r_i)$ *pentru orice* $i \in [n]$ *.*

Notaţii 3.1.5 *Fie* Γ *o mulţime de clauze şi* $G = \{l_1 \doteq_{s_1} r_1, \ldots, l_n \doteq_{s_n} r_n\}$ *o mulţime de egalităţi formale între elemente din* $T_\Sigma(X)$*. Notăm* $\Gamma \models (\exists X)G$ *dacă* $\mathcal{A} \models (\exists X)G$ *pentru orice* Γ*-algebră* $\mathcal{A}$*.*

Programarea logică ecuaţională îşi pune următoarea problemă $(\exists X)G$.

Teoremele lui Herbrand se referă la posibilitatea rezolvării acestor ecuaţii în toate Γ-algebrele.

Morfismele de algebre se extind natural pentru perechi de elemente şi pentru mulţimi de perechi de elemente. Dacă $h : \mathcal{A} \longrightarrow \mathcal{B}$ atunci pentru orice $a, b \in \mathcal{A}$ prin definiţie $h((a,b)) = (h(a), h(b))$ sau $h_s(a \doteq_s b) = (h_s(a) \doteq_s h_s(b))$. De asemenea în locul notaţiei $h_s(a) = h_s(b)$ vom mai utiliza $h_s((a,b)) \subseteq \Delta_B$.

3.2 Teoremele lui Herbrand

În literatura de specialitate se găsesc două teoreme ale lui Herbrand. Teorema care urmează combină cele două teoreme într-una singură.

Teorema 3.2.1 *Următoarele afirmaţii sunt echivalente:*

1. $\Gamma \models (\exists X)G$ *;*
2. $T_{\Sigma,\Gamma} \models (\exists X)G$ *;*
3. *Există* $\psi : T_\Sigma(X) \to T_\Sigma$ *astfel încât* $\Gamma \models (\forall\emptyset)\psi(G)$*, unde* $\psi(G) = \{\psi(l_1) =_{s_1} \psi(r_1), ..., \psi(l_n) =_{s_n} \psi(r_n)\}$ *.*

Demonstraţie: (1) $\Rightarrow$ (2) este evidentă, deoarece $T_{\Sigma,\Gamma}$ este o Γ-algebră.

(2) $\Rightarrow$ (3) Conform ipotezei, există $h : T_\Sigma(X) \to T_{\Sigma,\Gamma}$ astfel încât $h(l_i) = h(r_i)$ pentru orice $i \in [n]$. Pentru că $T_\Sigma(X)$ este algebră liberă şi deci proiectivă, există $\psi : T_\Sigma(X) \to T_\Sigma$ astfel încât $\psi;\widehat{\ } = h$.

$$\begin{tikzcd}
T_\Sigma(X) \arrow[rr, "\exists\psi"] \arrow[dr, "h"'] & & T_\Sigma \arrow[dl, "\widehat{\ }"] \\
& T_{\Sigma,\Gamma} &
\end{tikzcd}$$

Figure 3.1: Proiectivitatea Σ-algebrei $T_\Sigma(X)$

Avem deci că $\widehat{\psi(l_i)} = \widehat{\psi(r_i)}$ pentru orice $i \in [n]$. Deoarece $T_{\Sigma,\Gamma}$ se obţine prin factorizarea algebrei T_Σ la congruenţa semantică deducem că $\psi(l_i) \equiv_\Gamma \psi(r_i)$ pentru orice $i \in [n]$.

Rezultă că pentru orice morfism $f : T_\Sigma \longrightarrow \mathcal{A} \models \Gamma$ că $f(\psi(l_i)) = f(\psi(r_i))$ pentru orice $i \in [n]$.

Prin urmare $\Gamma \models (\forall\emptyset)\psi(G)$.

(3) $\Rightarrow$ (1) Fie M o Γ-algebră. Avem de arătat că $M \models (\exists X)G$.

Arătăm că $\psi;\alpha_M : T_\Sigma(X) \to M$ este morfismul care verifică proprietatea din definiţie. Deoarece $M \models \Gamma$, din ipoteză rezultă că $M \models (\forall\emptyset)\psi(G)$. Utilizând în definiţie morfismul $\alpha_M : T_\Sigma \to M$ deducem că $\alpha_M(\psi(l_i)) = \alpha_M(\psi(r_i))$, pentru orice $i \in [n]$. Rezultă că $(\psi;\alpha_M)(l_i) = (\psi;\alpha_M)(r_i)$, pentru orice $i \in [n]$.

Deci $M \models (\exists X)G$ şi demonstraţia se încheie. $\Box$

Prima echivalenţă a teoremei arată că problema programării logice se reduce la rezolvarea ei în Γ-algebra iniţială $0_{\Sigma,\Gamma}$, pas deosebit de mare deoarece în loc de o clasă de algebre lucrăm numai cu o algebră.

A treia afirmaţie din teoremă ne arată că rezolvarea problemei se poate face utilizând numai algebre libere. Ea face legătura cu programarea ecuaţională şi cu conceptul de soluţie.

A treia afirmaţie mai arată că pentru existenţa soluţiei este necesar ca suporturile din Σ-algebra iniţială T_Σ, corespunzătoare sorturilor variabilelor cuantificate existenţial, sa fie nevide.

Exerciţiu. Dată signatura Σ, să se determine mulţimea sorturilor pentru care suportul corespunzător din T_Σ este nevid.

Din a treia afirmaţie a teoremei lui Herbrand se mai vede că soluţia $\psi : T_\Sigma(X) \to T_\Sigma$ se caută într-o subalgebră T_Σ a algebrei $T_\Sigma(X)$ în care este pusă problema $(\exists X)G$. Chiar dacă în T_Σ nu mai apare nici o variabilă, acest fapt este neglijat în practică. Să presupunem că avem de rezolvat ecuaţia $x = f(y)$ unde evident $X = \{x, y\}$. Soluţia este dată chiar de această ecuaţie, adică în $T_\Sigma(\{y\})$ chiar dacă suportul corespunzător lui y este vid în T_Σ. Prin urmare ecuaţia $x = f(y)$ va avea soluţie numai în algebrele în care suportul corespunzător sortului lui y este nevid. În continuare ne vor interesa şi astfel de soluţii.

Chapter 4

CORECTITUDINEA REGULILOR PROGRAMĂRII LOGICE

4.1 Preliminarii

Amintim că pentru o Σ-algebra $\mathcal{A}$, $Sen(\mathcal{A}\ ) = \{a \doteq_s b \mid s \in S, a, b \in A_s\}$ (*propoziţiile lui* $\mathcal{A}$).

În continuare vom considera Γ o mulţime de ecuaţii condiţionate, iar $G \subseteq Sen(\mathcal{A})$.

Spunem că $\Gamma \models (\forall \mathcal{A})G$ dacă, pentru orice Σ-morfism $h : \mathcal{A} \to \mathcal{M} \models \Gamma$, avem $h_s(a) = h_s(b)$, pentru orice $a \doteq_s b \in G$. Pentru uşurinţă, vom mai nota şi cu $h(G) \subseteq \Delta_M$ faptul că $h_s(a) = h_s(b)$, pentru orice $a \doteq_s b \in G$, unde $h : \mathcal{A} \to \mathcal{M}$.

Congruenţa semantică $\equiv_\Gamma^\mathcal{A}$, relativ la o Σ-algebra $\mathcal{A}$, este definită prin:

$$a \equiv_\Gamma^\mathcal{A} b \;\Leftrightarrow\; \Gamma \models a \doteq_s b \;\Leftrightarrow\; h_s(a) = h_s(b), \text{ pentru orice } h : \mathcal{A} \to \mathcal{M} \models \Gamma.$$

Observaţie. $\Gamma \models (\forall \mathcal{A})G$ dacă şi numai dacă $G \subseteq \equiv_\Gamma^\mathcal{A}$.

Propoziţie 4.1.1 *Dacă* $h : \mathcal{A} \to \mathcal{B}$ *este un* Σ*-morfism, atunci* $h(\equiv_\Gamma^\mathcal{A}) \subseteq \equiv_\Gamma^\mathcal{B}$.

Demonstraţie Fie $a \equiv_\Gamma^\mathcal{A} b$. Vrem să arătăm că $h(a) \equiv_\Gamma^\mathcal{B} h(b)$.

Fie $f : \mathcal{B} \to \mathcal{M} \models \Gamma$. Cum $h; f : \mathcal{A} \to \mathcal{M} \models \Gamma$ şi $a \equiv_\Gamma^\mathcal{A} b$, rezultă că $(h; f)(a) = (h; f)(b)$, echivalent cu $f(h(a)) = f(h(b))$. Cum f a fost ales arbitrar, rezultă că $h(a) \equiv_\Gamma^\mathcal{B} h(b)$. $\square$

Corolar 4.1.1 *Dacă* $\Gamma \models (\forall \mathcal{A})G$ *şi* $h : \mathcal{A} \to \mathcal{B}$ *un* Σ*-morfism, atunci* $\Gamma \models (\forall \mathcal{B})h(G)$.

Demonstraţie Din ipoteză, conform observaţiei, deducem că $G \subseteq \equiv_\Gamma^\mathcal{A}$. Aplicând Propoziţia 4.1.1, deducem $h(G) \subseteq \equiv_\Gamma^\mathcal{B}$, deci, conform observaţiei, $\Gamma \models (\forall \mathcal{B})h(G)$. $\square$

Fie $\mathcal{A}$ o Σ-algebră şi z o nouă variabilă de sort s astfel încât $z \notin A_s$. Considerăm algebra liber generată de $A \cup \{z\}$, $T_\Sigma(A \cup \{z\})$, pe care o notăm cu $\mathcal{A}[z]$. Un element c din $\mathcal{A}[z]$ se numeşte *context* dacă numărul apariţiilor lui z în c este 1. Pentru $d \in A_s$, vom nota cu $(z \leftarrow d) : \mathcal{A}[z] \to \mathcal{A}$ unicul Σ-morfism cu proprietatea $(z \leftarrow d)(z) = d$ şi $(z \leftarrow d)(a) = a$, pentru orice $a \in A$. Pentru orice t din $\mathcal{A}[z]$ şi $a \in A_s$, vom prefera să scriem $t[a]$, în loc de $(z \leftarrow a)(t)$.

Datorită faptului că regulile programării logice lucrează pe mulţimi de egalităţi formale, vom defini noţiunea de context extins. Fie $\mathcal{A}$ o Σ-algebră, $c \in \mathcal{A}[z]_s$ un context şi $v \in A_s$. O egalitate formală de forma $c \doteq_s v$ sau $v \doteq_s c$ se numeşte *context extins*. Un context extins $c \doteq_s v$ (respectiv $v \doteq_s c$) va fi notat cu C. Să observăm că $(c \doteq_s v)[a] = (z \leftarrow a)(c \doteq_s v) = (c[a] \doteq_s c[v]) = (c[a] \doteq_s v)$.

Orice morfism $h : \mathcal{A} \to \mathcal{B}$ se poate extinde, în mod unic, la un morfism $h^z : \mathcal{A}[z] \to \mathcal{B}[z]$ prin $h^z(z) = z$ şi $h^z(a) = h(a)$, pentru orice $a \in A$. Pentru orice $a \in A$, avem $(z \leftarrow a); h = h^z; (z \leftarrow h(a))$. Pentru un context $c \in \mathcal{A}[z]$ deducem că $h(c[a]) = h^z(c)[h(a)]$, unde $h^z(c)$ este context.

Pentru C un context extins, se observă că $h^z(C)$ este un context extins şi că

$$h(C[a]) = h^z(C)[h(a)].$$

4.2 Soluţii şi reguli de deducţie

Problema programării logice este $(\exists\mathcal{A})G$.

Definiţie 4.2.1 *Un Σ-morfism $s : \mathcal{A} \to \mathcal{B}$ se numeşte **soluţie** pentru $(\exists\mathcal{A})G$ dacă $\Gamma \models (\forall\mathcal{B})s(G)$.*

Propoziţie 4.2.1 *Compunerea unei soluţii cu orice Σ-morfism este soluţie.*

Demonstraţie Fie $s : \mathcal{A} \to \mathcal{B}$ o soluţie pentru $(\exists\mathcal{A})G$ şi $h : \mathcal{B} \to \mathcal{C}$ un morfism. Vrem să arătăm că $s;h : \mathcal{A} \to \mathcal{C}$ este soluţie pentru $(\exists\mathcal{A})G$, adică $\Gamma \models (\forall\mathcal{C})(s;h)(G)$.

Deoarece $s : \mathcal{A} \to \mathcal{B}$ este soluţie pentru $(\exists\mathcal{A})G$, atunci $\Gamma \models (\forall\mathcal{B})s(G)$. Corolarul 4.1.1 implică $\Gamma \models (\forall\mathcal{C})h(s(G))$.

Deci $\Gamma \models (\forall\mathcal{C})(s;h)(G)$ şi astfel $s;h$ este soluţie pentru $(\exists\mathcal{A})G$. □

Din propoziţia precedentă rezultă că dacă avem o soluţie, atunci aceasta nu este unică. Acest fapt ne îndeamnă să căutăm o soluţie cât mai generală.

În general, soluţiile sunt construite în mai multe etape, apărând în final ca o compunere de morfisme. Morfismele care apar în procesul de calcul şi care, sperăm, ca în final să furnizeze o soluţie sunt numite *morfisme calculate*.

În continuare vom prezenta regulile de deducţie folosite în programarea logică. Aceste reguli ne permit să trecem de la o mulţime G de egalităţi formale la o altă mulţime G' de egalităţi formale, obţinând şi un morfism calculat. Aplicarea acestor reguli va înceta în momentul în care ajungem la o mulţime vidă de egalităţi formale. În acest punct, putem compune toate morfismele calculate găsite, în ordinea apariţiei lor, şi astfel vom obţine o soluţie pentru problema iniţială. Această afirmaţie va fi probată mai târziu în ipoteza că toate regulile de deducţie utilizate sunt corecte.

Menţionăm în continuare mai multe reguli de deducţie utile în programarea logică.

Regula morfismului: *Dacă $G \subseteq Sen(T_\Sigma(X))$ şi $\theta : T_\Sigma(X) \to T_\Sigma(Y)$, atunci*

$$G \to_m \theta(G),$$

cu morfismul calculat θ.

Regula reflexiei extinse: *Dacă $G \subseteq Sen(T_\Sigma(X))$ şi $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ astfel încât $\theta_s(l) = \theta_s(r)$, atunci*

$$G \cup \{l \doteq_s r\} \to_{re} \theta(G),$$

cu morfismul calculat θ.

Regula reflexiei: *Dacă $G \subseteq Sen(T_\Sigma(X))$ şi $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ astfel încât $\theta = CGU\{l, r\}$, atunci*

$$G \cup \{l \doteq_s r\} \to_r \theta(G),$$

cu morfismul calculat θ.

Regula pararescrierii: *Fie $G \subseteq Sen(T_\Sigma(X))$, $(\forall Y)l \doteq_s r \ if \ H \in \Gamma$ şi morfismul $\theta : T_\Sigma(Y) \to T_\Sigma(X)$. Dacă C este un context extins cu variabila distinsă z de sort s, atunci*

$$G \cup \{C[\theta_s(l)]\} \to_{pr} G \cup \theta(H) \cup \{C[\theta_s(r)]\}.$$

Menţionăm că pentru pararescriere, morfismul calculat este morfismul identitate.

Regula paramodulaţiei extinse: *Fie $(\forall Y)l \doteq_s r \ if \ H \in \Gamma$. Considerăm X astfel încât $X \cap Y = \emptyset$, $G \subseteq Sen(T_\Sigma(X))$ şi morfismul $\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ astfel încât $\theta_s(l) = \theta_s(a)$, unde $a \in T_\Sigma(X)_s$. Dacă C este un context extins cu variabila distinsă z de sort s, atunci*

$$G \cup \{C[a]\} \to_{pe} \theta(G \cup H \cup \{C[r]\}),$$

cu morfismul calculat $\theta_{/X}$, restricţia lui θ la $T_\Sigma(X)$.

Regula paramodulaţiei: *Fie $(\forall Y)l \doteq_s r\ if\ H \in \Gamma$. Considerăm X astfel încât $X \cap Y = \emptyset$, $G \subseteq Sen(T_\Sigma(X))$ şi morfismul $\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ astfel încât $\theta = CGU\{l, a\}$, unde $a \in T_\Sigma(X)_s$. Dacă C este un context extins cu variabila distinsă z de sort s, atunci*

$$G \cup \{C[a]\} \to_p \theta(G \cup H \cup \{C[r]\}),$$

cu morfismul calculat $\theta_{/X}$, restricţia lui θ la $T_\Sigma(X)$.

Comentariu Dorinţa exprimată mai sus de a obţine o soluţie cât mai generală face ca regulile de deducţie utilizate de semantica operaţională a programării logice să fie mai restrictive. Mai precis reflexia este reflexia extinsă cu condiţia suplimentară ca θ să fie cel mai general unificator pentru l şi r, iar paramodulaţia este paramodulaţia extinsă în care θ este cel mai general unificator pentru l şi a. Conform uzanţelor presupunem $X \cap Y = \emptyset$, fapt posibil datorită cuantificării universale a clauzei care ne permite să alegem variabile noi în locul celor din Y ori de câte ori este necesar.

4.3 Legături între regulile de deducţie

În continuare vom prezenta legăturile între regulile de deducţie pentru programarea logică. Aceste legături sunt importante, ajutându-ne, de exemplu, să demonstrăm mai uşor unele proprietăţi ale regulilor de deducţie.

În primul rând, este evident că regula reflexiei şi regula paramodulaţiei sunt cazuri particulare ale regulilor reflexiei extinse şi, respectiv, paramodulaţiei extinse (caz particular în care cerem ca morfismul implicat în regulă să fie un cel mai general unificator, nu doar un Σ-morfism).

Observăm că

$$G \cup \{l \doteq_s r\} \to_m \theta(G) \cup \{\theta_s(l) \doteq \theta_s(r)\},$$

cu morfismul calculat θ, ceea ce arată că regula morfismului şi eliminarea egalităţilor evidente permit eliminarea regulii reflexiei extinse dintre regulile de lucru.

Propoziţie 4.3.1 *Pararescrierea este un caz particular de paramodulaţie extinsă.*

Demonstraţie Considerăm pararescrierea $G \cup \{C[h_s(l)]\} \to_{pr} G \cup h(H) \cup \{C[h_s(r)]\}$, unde $(\forall Y)l \doteq_s r\ if\ H \in \Gamma$ şi $h : T_\Sigma(Y) \to T_\Sigma(X)$ este un Σ-morfism (presupunem $X \cap Y = \emptyset$).

Luăm $a = h_s(l)$. Să considerăm Σ-morfismul $\theta : T_\Sigma(X \cup Y) \to T_\Sigma(X)$ definit prin:

1. $\theta(y) = h(y)$, pentru orice $y \in Y$,
2. $\theta(x) = x$, pentru orice $x \in X$.

Observăm că $\theta(t) = t$, pentru orice $t \in T_\Sigma(X)$ şi $\theta(u) = h(u)$, pentru orice $u \in T_\Sigma(Y)$.

Deoarece G este o mulţime de egalităţi formale din $Sen(T_\Sigma(X))$, rezultă că $\theta_s(u \doteq_s v) = (\theta_s(u) \doteq_s \theta_s(v)) = (u \doteq_s v)$, pentru orice $u \doteq_s v \in G$. În concluzie, putem scrie $\theta(G) = G$.

Similar, H este o mulţime de egalităţi formale din $Sen(T_\Sigma(Y))$ şi astfel avem $\theta_s(u \doteq_s v) = (\theta_s(u) \doteq_s \theta_s(v)) = (h_s(u) \doteq_s h_s(v))$, pentru orice $u \doteq_s v \in H$. În concluzie, putem scrie $\theta(H) = h(H)$.

De asemenea, contextul extins C este o egalitate formală din $Sen(T_\Sigma(X \cup \{z\}))$ şi astfel avem $\theta^z(C) = C$.

Observăm că $\theta_s(r) = h_s(r)$ şi $\theta_s(a) = \theta_s(h_s(l)) = h_s(l) = \theta_s(l)$, deoarece $l, r \in T_\Sigma(Y)$.

Putem aplica regula paramodulaţiei extinse pentru $(\forall Y)l \doteq_s r\ if\ H \in \Gamma$ şi $\theta : T_\Sigma(X \cup Y) \to T_\Sigma(X)$ astfel obţinând:

$$G \cup \{C[h_s(l)]\} = G \cup \{C[a]\} \to_{pe} \theta(G \cup H \cup \{C[r]\}) = \theta(G) \cup \theta(H) \cup \theta(C[r]) =$$

$$= G \cup h(H) \cup \theta^z(C)[\theta_s(r)] = G \cup h(H) \cup C[h_s(r)].$$

Morfismul calculat $\theta_{/X}$ este identitatea lui $T_\Sigma(X)$. $\Box$

Propoziţie 4.3.2 *Dacă pentru orice clauză $(\forall Y)l \doteq_s r\ if\ H$ din Γ, orice variabilă din Y apare în l, atunci pararescrierea este un caz particular de paramodulaţie în care substituţia calculată este o identitate.*

Demonstraţie Păstrăm notaţiile şi demonstraţia din propoziţia precedentă. Vom proba, în plus, că θ este cel mai general unificator pentru l şi a.

Fie $u : T_\Sigma(X \cup Y) \to \mathcal{B}$ un unificator pentru l şi a. Deoarece $u_s(l) = u_s(a) = u_s(h_s(l))$ şi orice variabilă din Y apare în l, deducem că $u(y) = u(h(y))$, pentru orice $y \in Y$.

Notăm cu $u_{|X}$ restricţia lui u la X şi observăm că $u_{|X}(x) = u(x)$, pentru orice $x \in X$.

Observăm că $\theta; u_{|X} = u$: pentru orice $x \in X$, $u_{|X}(\theta(x)) = u_{|X}(x) = u(x)$, şi pentru orice $y \in Y$, $u_{|X}(\theta(y)) = u_{|X}(h(y)) = u(h(y)) = u(y)$.

Deci θ este cel mai general unificator pentru l şi a deoarece $u = \theta; u_{|X}$. $\Box$

Lemă 4.3.1 *Dacă $(\forall Y)t \doteq_s t \in \Gamma$, atunci $G \to_p (x \leftarrow t)(G)$, unde x este o variabilă care apare în G şi nu apare în t.*

Demonstraţie Alegem o apariţie a lui x în G şi scriem $G = G' \cup C[x]$, unde C este un context extins. Aplicând regula paramodulaţiei pentru $(\forall Y)t \doteq_s t \in \Gamma$, $a = x$, $\theta = CGU\{a, l\} = CGU\{x, t\} = x \leftarrow t$, obţinem:

$$G = G' \cup C[x] \to_p (x \leftarrow t)(G' \cup C[t]) = (x \leftarrow t)(G).$$

Ultima egalitate este adevărată deoarece x nu apare în t. $\Box$

Ipoteza x apare în G nu este esenţială deoarece, dacă x nu apare în G, $(x \leftarrow t)(G) = G$ şi prin urmare $(x \leftarrow t)(G)$ se obţine din G în 0 paşi.

Lemă 4.3.2 (Lema substituţiei) *Dacă sunt îndeplinite următoarele condiţii: G este o mulţime finită, $(\forall x)x \doteq x \in \Gamma$ pentru orice variabilă x, $(\forall x_1\ \forall x_2 \ldots \forall x_n)f(x_1, x_2, \ldots, x_n) \doteq f(x_1, x_2, \ldots, x_n) \in \Gamma$ pentru orice simbol de operaţie f, atunci regula morfismului poate fi realizată prin regula paramodulaţiei.*

Demonstraţie Primele două afirmaţii care urmează dovedesc că axiomele lemei substituţiei, mai sărace decât cele ale lemei 4.3.1, sunt suficiente pentru a demonstra concluzia lemei 4.3.1:

1. **Substituţia unei variabile x cu o variabilă y poate fi realizată prin regula paramodulaţiei în prezenţa axiomei $(\forall y)y \doteq y$.**

În cazul în care x apare în G şi $x \neq y$ se aplica Lema 4.3.1. În rest, evident.

2. **Arătăm că substituţia unei variabile cu un termen poate fi realizată prin paramodulaţie în prezenţa axiomelor $(\forall x)x \doteq x$ şi $(\forall x_1\ \forall x_2 \ldots \forall x_n)f(x_1, x_2, \ldots, x_n) \doteq f(x_1, x_2, \ldots, x_n)$.**

Vom demonstra acest lucru prin inducţie după structura termenului t.

Primul pas al inducţiei este chiar (1).

Presupunem că $t = f(t_1, t_2, \ldots, t_n)$.

Dacă x nu apare în G, atunci nu avem nimic de demonstrat. Presupunem că x apare în G şi folosind $(\forall x_1\ \forall x_2 \ldots \forall x_n)f(x_1, x_2, \ldots, x_n) \doteq f(x_1, x_2, \ldots, x_n) \in \Gamma$, unde variabilele $x_1, \ldots, x_n$ sunt noi, şi Lema 4.3.1 deducem

$$G \to_p (x \leftarrow f(x_1, x_2, \ldots, x_n))(G).$$

În continuare se aplică ipoteza de inducţie pentru orice $1 \le i \le n$, substituind fiecare x_i cu t_i.

Mai observăm că

$$x \leftarrow f(x_1, x_2, \ldots, x_n)\ ;\ x_1 \leftarrow t_1\ ;\ x_2 \leftarrow t_2\ ;\ \ldots\ ;\ x_n \leftarrow t_n = x \leftarrow f(t_1, t_2, \ldots, t_n),$$

deoarece variabilele $x_1, x_2, \ldots, x_n$ sunt noi.

3. **Arătăm că regula morfismului poate fi realizată prin regula paramodulaţiei.**

Fie $h : T_\Sigma(X) \to T_\Sigma(Y)$. Cum $var(G) = \{x_1, x_2, \ldots, x_n\} \subseteq X$, a realiza regula morfismului revine la a înlocui fiecare variabilă x_i cu $h(x_i)$. Putem realiza acest lucru conform punctelor (1) şi (2):

- întâi înlocuim fiecare variabilă x_i cu o variabilă nouă z_i, pentru orice $1 \le i \le n$:

$$G \to_p (x_1 \leftarrow z_1)(G) \to_p \ldots \to_p (x_n \leftarrow z_n)(\ldots(x_1 \leftarrow z_1)(G)\ldots) = G'$$

- acum înlocuim pentru fiecare $1 \le i \le n$, variabila z_i cu $h(x_i)$:

$$G' \overset{*}{\to}_p (z_1 \leftarrow h(x_1))(G) \overset{*}{\to}_p \ldots \overset{*}{\to}_p (z_n \leftarrow h(x_n))(\ldots(z_1 \leftarrow h(x_1))(G)\ldots) = h(G). \ \Box$$

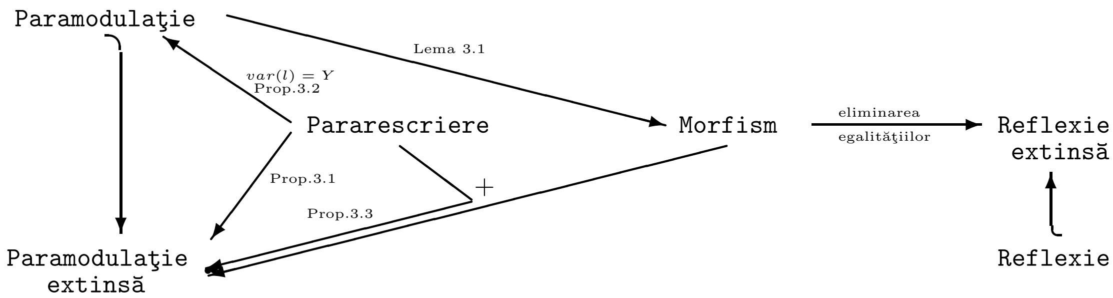

Figure 4.1: Legăturile dintre regulile de deducţie

Observaţie. În demonstraţia anterioară, la pasul (3), este extrem de important să schimbăm toate variabilele x_i cu variabile noi. Altfel am putea obţine rezultate nedorite, ca în exemplul de mai jos:

Dacă $h : T_\Sigma(\{x, y\}) \to T_\Sigma(\{x, y, z\})$, $h(x) = z$, $h(y) = x$ şi $G = \{x \doteq y\}$, atunci:

$$h(G) = (h(x) \doteq h(y)) = (z \doteq x),$$
$$(x \leftarrow h(x))((y \leftarrow h(y))(G)) = (x \leftarrow h(x))(x \doteq x) = (z \doteq z).$$

Deci $h(G) \neq (x \leftarrow h(x))((y \leftarrow h(y))(G))$.

Propoziţie 4.3.3 *Regula paramodulaţiei extinse se poate obţine din regula morfismului şi regula pararescrierii.*

Demonstraţie Fie $(\forall Y) l \doteq_s r \ if \ H \in \Gamma$, $\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ astfel încât $\theta_s(l) = \theta_s(a)$, unde $a \in T_\Sigma(X)_s$. Fie C un context extins şi z o variabilă nouă. Aplicând regula morfismului pentru morfismul θ, obţinem:

$$G \cup \{C[a]\} \to_m \theta(G) \cup \{\theta(C[a])\}.$$

Menţionăm următoarele egalităţi:

$$\theta(C[a]) = \theta^z(C)[\theta(a)] = \theta^z(C)[\theta(l)].$$

Acum putem aplica regula pararescrierii şi obţinem:

$$\theta(G) \cup \{\theta^z(C)[\theta(l)]\} \to_{pr} \theta(G) \cup \theta(H) \cup \{\theta^z(C)[\theta(r)]\} = \theta(G \cup H \cup \{C[r]\}).$$

Deci $G \cup \{C[a]\} \to_m \theta(G) \cup \{\theta^z(C)[\theta(l)]\} \to_{pr} \theta(G \cup H \cup \{C[r]\})$. $\square$

Putem sintetiza legăturile găsite între regulile de deducţie pentru programarea logică prin Figura 1.

4.4 Corectitudinea regulilor de deducţie

Definiţie 4.4.1 *Fie R o regulă de deducţie. Să presupunem că aplicând regula R obţinem $G \to_R G'$ cu morfismul calculat θ. Spunem că regula R este o* ***regulă corectă*** *dacă este îndeplinită următoarea condiţie: dacă s este o soluţie pentru G', atunci $\theta; s$ este soluţie pentru G.*

Dacă se aplică numai reguli corecte ajungându-se, în final, la mulţimea vidă de ecuaţii (sau la o mulţime formată doar din egalităţi adevărate), atunci compunerea tuturor morfismelor calculate este o soluţie a problemei iniţiale. Această afirmaţie rezultă din faptul că morfismul identitate este soluţie pentru orice mulţime de egalităţi adevărate, inclusiv mulţimea vidă.

În continuare vom arătă că regulile de deducţie considerate în secţiunile anterioare sunt corecte.

Deoarece $\models (\forall X) l \doteq_s l$, orice soluţie pentru G este soluţie şi pentru $G \cup \{l \doteq_s l\}$. Prin urmare *eliminarea egalităţilor adevărate* este o regulă corectă.

Propoziţie 4.4.1 *Regula morfismului este corectă.*

Demonstraţie Presupunem că $G \to_m \theta(G)$, unde $\theta : T_\Sigma(X) \to T_\Sigma(Y)$. Fie $s : T_\Sigma(Y) \to T_\Sigma(Z)$ o soluţie pentru $\theta(G)$, adică $\Gamma \models (\forall Z) s(\theta(G))$. Trebuie să arătăm că $\theta; s$ este soluţie pentru G, adică $\Gamma \models (\forall Z)(\theta; s)(G)$, ceea ce este evident. $\square$

Propoziţie 4.4.2 *Regula reflexiei extinse este corectă.*

Demonstratie Ştim că regula reflexiei extinse se poate obţine din regula morfismului. Cum regula morfismului este corectă, rezultă că şi regula reflexiei extinse este corectă. □

Corolar 4.4.1 *Regula reflexiei este corectă.*

Propoziţie 4.4.3 *Regula pararescrierii este corectă.*

Demonstraţie Considerăm pararescrierea $G \cup C[\theta_s(l)] \to_{pr} G \cup \theta(H) \cup \{C[\theta_s(r)]\}$, unde $(\forall Y) l \doteq_s r \; if \; H \in \Gamma$ şi $\theta : T_\Sigma(Y) \to T_\Sigma(X)$ un Σ-morfism.

Fie $S : T_\Sigma(X) \to \mathcal{B}$ o soluţie pentru $(\exists X) G \cup \theta(H) \cup \{C[\theta_s(r)]\}$, adică

$$\Gamma \models (\forall \mathcal{B}) S(G \cup \theta(H) \cup \{C[\theta_s(r)]\}). \tag{1}$$

Avem de arătat că $S : T_\Sigma(X) \to \mathcal{B}$ este soluţie pentru $(\exists X)(G \cup C[\theta_s(l)])$, adică

$$\Gamma \models (\forall \mathcal{B}) S(G \cup \{C[\theta_s(l)]\}).$$

Fie $h : \mathcal{B} \to \mathcal{M} \models \Gamma$. Din (1) deducem $(S;h)(G) \cup (\theta;S;h)(H) \cup (S;h)(C[\theta_s(r)]) \subseteq \Delta_M$. Prin urmare deducem:

$$(S;h)(G) \subseteq \Delta_M, \tag{2}$$

$$(\theta;S;h)(H) \subseteq \Delta_M, \tag{3}$$

$$(S;h)(C[\theta_s(r)]) \subseteq \Delta_M. \tag{4}$$

Deoarece $\mathcal{M} \models (\forall Y) l \doteq_s r \; if \; H$, folosind morfismul $\theta;S;h : T_\Sigma(Y) \to \mathcal{M}$ şi relaţia (3), deducem că

$$(\theta;S;h)_s(l) = (\theta;S;h)_s(r). \tag{5}$$

Probăm că

$$(z \leftarrow \theta(l)); S; h = (z \leftarrow \theta_s(r)); S; h. \qquad (6)$$

Observăm că cei doi membri sunt Σ-morfisme de la $T_\Sigma(X \cup \{z\})$ la $\mathcal{M}$.

Pentru orice $x \in X$ avem $(z \leftarrow \theta_s(l))(x) = x = (z \leftarrow \theta_s(r))(x)$. Pe de altă parte $(z \leftarrow \theta_s(l))(z) = \theta_s(l)$ şi $(z \leftarrow \theta_s(r))(z) = \theta_s(r)$. Folosind (5) deducem că

$$((z \leftarrow \theta_s(l)); S; h)_s(z) = (\theta;S;h)_s(l) = (\theta;S;h)_s(r) = ((z \leftarrow \theta_s(r); S; h)_s(z).$$

Prin urmare egalitatea (6) este probată.

Observăm că $h(S(C[\theta_s(l)])) = ((z \leftarrow \theta_s(l)); S; h)(C) \overset{(6)}{=} ((z \leftarrow \theta_s(r)); S; h)(C) = (S;h)(C[\theta_s(r)])$.

Din (4) deducem că $h(S(C[\theta_s(l)])) \subseteq \Delta_M$. Folosind (2) deducem $h(S(G \cup \{C[\theta_s(l)]\})) \subseteq \Delta_M$. □

Propoziţie 4.4.4 *Regula paramodulaţiei extinse este corectă.*

Demonstraţie Din Propoziţia 4.3.3 ştim că orice paramodulaţie extinsă se poate obţine din regula morfismului şi regula pararescrierii. Din Propoziţiile 4.4.1 şi 4.4.3 ştim că regulile morfismului şi pararescrierii sunt corecte, de unde rezulta că şi regula paramodulaţiei extinse este corectă. □

Corolar 4.4.2 *Regula paramodulaţiei este corectă.*

Chapter 5

RESCRIERI

5.1 Reguli de Deducţie pentru Rescriere

Reamintim alte reguli de deducţie pentru **Re**scriere în algebra $\mathcal{A}$ denumite **R**eflexivitate, **T**ranzitivitate şi **C**ompatibilitate cu operaţiile din Σ:

R	$a \doteq_s a$
T	$a \doteq_s d$ şi $d \doteq_s c$ implică $a \doteq_s c$
CΣ	$a_i \doteq_{s_i} c_i$ pentru $i \in [n]$ implică $A_\sigma(a_1, a_2, \ldots, a_n) \doteq_s A_\sigma(c_1, c_2, \ldots, c_n)$ pentru orice $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$.

Aceste reguli împreună cu $\mathbf{Sub}_\Gamma$ permit o demonstraţie mai uşoară a rezultatelor teoretice.

Pornind de la regulile de substituţie respectiv rescriere vom introduce încă două reguli de deducţie în algebra $\mathcal{A}$, substituţia şi respectiv rescrierea în subtermeni.

$\mathbf{SSub}_\Gamma$ Pentru orice $(\forall X)\, l \doteq_s r$ **if** $H \in \Gamma$ şi orice morfism $h\colon T_\Sigma(X) \to \mathcal{A}$
$(\forall u \doteq_{s'} v \in H) h_{s'}(u) \doteq_{s'} h_{s'}(v)$ implică $(\forall c \in \mathcal{A}[z]_{s'}$ context$)$ $c[h_s(l)] \doteq_{s'} c[h_s(r)]$

$\mathbf{SRew}_\Gamma$ Pentru orice $(\forall X)\, l \doteq_s r$ **if** $H \in \Gamma$ şi orice morfism $h\colon T_\Sigma(X) \to \mathcal{A}$
$(\forall u \doteq_{s'} v \in H)(\exists d \in A_{s'}) h_{s'}(u) \doteq_{s'} d$ şi $h_{s'}(v) \doteq_{s'} d$ implică $(\forall c \in \mathcal{A}[z]_{s'}$ context$)$ $c[h_s(l)] \doteq_{s'} c[h_s(r)]$

Remarcaţi că $\mathbf{SRew}_\Gamma$, *rescrierea într-un subtermen*, este o regulă de deducţie mai puternică decât $\mathbf{Rew}_\Gamma$, *rescrierea*, care poate fi obţinută din $\mathbf{SRew}_\Gamma$ pentru $c = z$. Analog $\mathbf{SSub}_\Gamma \Longrightarrow \mathbf{Sub}_\Gamma$.

Se observă că $\mathbf{Rew}_\Gamma$ şi **R** implică $\mathbf{Sub}_\Gamma$. Indicaţie: se ia $d = h(v)$.

5.2 Rescriere

Dată o mulţime Γ de ecuaţii condiţionate putem defini Γ–rescrierea, sau mai pe scurt rescrierea, ca fiind cea mai mică relaţie închisă la **R**, **T**, **C**Σ, $\mathbf{Rew}_\Gamma$. Din păcate unele demonstraţii sunt greu de făcut lucrând cu o astfel de definiţie, fapt pentru care vom da în cele ce urmează o construcţie efectivă pentru Γ–rescriere. Construcţia este făcută în trei etape, mulţimea Γ intervenind numai în a treia etapă. În prima etapă este definită închiderea la contexte ale unei relatii. În a doua etapă este construită închiderea unei relaţii în mulţimea preordinilor compatibile cu operaţiile

5.2.1 Închiderea la contexte

Definiţia 5.2.1 O relaţie Q pe A se numeţe **închisă la contexte** dacă pentru orice context c şi pentru orice pereche de elemente a, d din A, $a\, Q\, d$ implică $c[a]\, Q\, c[d]$.

Observaţia 5.2.2 *Fie Q o relaţie pe A închisă la contexte. Dacă Q este închisă la* $\mathbf{Sub}_\Gamma$*, respectiv* $\mathbf{Rew}_\Gamma$*, atunci Q este închisă la* $\mathbf{SSub}_\Gamma$*, respectiv* $\mathbf{SRew}_\Gamma$*.*

Definiţia 5.2.3 O relaţie $\rho \subset A \times A$ se numeşte compatibilă pe componente cu operaţiile algebrei $\mathcal{A}$ dacă

$$a\, \rho_{s_i}\, d \quad \text{implică} \quad A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n)\, \rho_s\, A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n)$$

pentru orice $\sigma \in \Sigma_{s_1 \ldots s_n, s}$, orice $i \in [n]$, unde $a_j \in A_{s_j}$ pentru orice $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$ şi $a, d \in A_{s_i}$.

Propoziţie 5.2.4 *O relaţie este închisă la contexte dacă şi numai dacă este compatibilă pe componente cu operaţiile algebrei.*

Demonstraţie: *Presupunem Q închisă la contexte. Pentru a demonstra compatibilitatea pe componente cu operaţia σ aplicăm ipoteza pentru un context de forma $\sigma(a_1, \ldots, a_{i-1}, z, a_{i+1}, \ldots, a_n)$.*

Reciproca se arată prin inducţie structurală în $\mathcal{A}[z]$.

Pasul 0: $c = z$. Pentru orice $(a, d) \in Q$, $c[a] = a$, $c[d] = d$, deci $(c[a], c[d]) \in Q$.

Pentru $c = \sigma(a_1, \ldots, a_{i-1}, c', a_{i+1}, \ldots, a_n)$ unde $c' \in \mathcal{A}[z]$ este un context

$$c[a] = (z \leftarrow a)(c) = A_\sigma(a_1, \ldots, a_{i-1}, (z \leftarrow a)(c'), a_{i+1}, \ldots, a_n) = A_\sigma(a_1, \ldots, a_{i-1}, c'[a], a_{i+1}, \ldots, a_n).$$

La fel $c[d] = A_\sigma(a_1, \ldots, a_{i-1}, c'[d], a_{i+1}, \ldots, a_n)$.
Din ipoteza de inducţie $c'[a]\, Q\, c'[d]$ şi ţinând cont de compatibilitatea pe componente obţinem $c[a]\, Q\, c[d]$.

Definiţia 5.2.5 Dacă Q este o relaţie pe A vom nota

$$\longrightarrow_Q \; = \; \{(c[a], c[d]) \; : \; (a, d) \in Q_s, \; c \in \mathcal{A}[z] \text{ este context unde variabila } z \text{ are sortul } s\}.$$

Propoziţie 5.2.6 *$\longrightarrow_Q$ este cea mai mică relaţie inchisă la contexte care include Q.*

Demonstraţie: *Pentru a dovedi că $\longrightarrow_Q$ este inchisă la contexte, vom prefera să arătăm că este compatibilă pe componente cu opreraţiile. Fie $(c[a], c[d])$ în $\longrightarrow_Q$ unde $(a, d) \in Q$, σ un simbol de operaţie şi a_i nişte elemente din $\mathcal{A}$. Folosind contextul $c' = \sigma(a_1, \ldots, a_{i-1}, c, a_{i+1}, \ldots, a_n)$ deducem că $(c'[a], c'[d])$ este în $\longrightarrow_Q$. Dar calculând ca mai sus*

$$c'[a] = A_\sigma(a_1, \ldots, a_{i-1}, c[a], a_{i+1}, \ldots, a_n) \quad \text{şi} \quad c'[d] = A_\sigma(a_1, \ldots, a_{i-1}, c[d], a_{i+1}, \ldots, a_n)$$

prin urmare $(A_\sigma(a_1, \ldots, a_{i-1}, c[a], a_{i+1}, \ldots, a_n), A_\sigma(a_1, \ldots, a_{i-1}, c[d], a_{i+1}, \ldots, a_n))$ este în $\longrightarrow_Q$.

Incluziunea $Q \subseteq \longrightarrow_Q$ se demonstrează folosind contextul z.

Dacă R este închisă la contexte, atunci $Q \subseteq R$ implică $\longrightarrow_Q \subseteq R$.

5.2.2 Închiderea la preordini compatibile cu operaţiile

Lemă 5.2.7 (Compatibilitatea pe argumente) *Fie $\rho \subset A \times A$ o relaţie tranzitivă şi reflexivă în algebra $\mathcal{A}$. Dacă ρ este compatibilă pe componente cu operaţiile algebrei $\mathcal{A}$, atunci ρ e compatibilă cu operaţiile, adică închisă la* $\mathbf{C}\Sigma$.

Demonstraţie: *Fie $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ şi $a_i, b_i \in A_{s_i}$ astfel încât $a_i \rho_{s_i} b_i$ pentru orice $i \in [n]$.*

Arătăm că $A_\sigma(a_1, \ldots, a_n)\, \rho_s\, A_\sigma(b_1, \ldots, b_n)$.

Dacă $n = 0$ avem $A_\sigma\, \rho_s\, A_\sigma$ din reflexivitate.

Dacă $n = 1$ avem din ipoteză $A_\sigma(a_1)\, \rho_s\, A_\sigma(b_1)$.

Dacă $n \geq 2$ aplicând succesiv ipoteza obţinem

$$A_\sigma(a_1, a_2, \ldots, a_n)\, \rho_s\, A_\sigma(b_1, a_2, \ldots, a_n)\, \rho_s\, A_\sigma(b_1, b_2, a_3, \ldots, a_n)\, \rho_s \ldots \rho_s\, A_\sigma(b_1, b_2, \ldots, b_n)$$

Din tranzitivitatea relaţiei ρ deducem $A_\sigma(a_1, a_2, \ldots, a_n)\, \rho_s\, A_\sigma(b_1, b_2, \ldots, b_n)$

Propoziţie 5.2.8 *$\xrightarrow{*}_Q$ este cea mai mică preordine compatibilă cu operaţiile care include Q.*

Demonstraţie: *Deoarece relaţia $\xrightarrow{*}_Q$ este reflexivă şi tranzitivă este suficient să demonstrăm compatibilitatea cu operaţiile numai pentru câte o componentă, adică să arătăm pentru fiecare $\sigma \in \Sigma_{s_1 \ldots s_n, s'}$ că $a \xrightarrow{*}_Q d$ implică*

$$A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) \xrightarrow{*}_Q A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n).$$

Probăm prin inducţie după numărul paşilor din $a \xrightarrow{*}_Q d$. Presupunem deci $a \longrightarrow_Q u$ şi $u \xrightarrow{*}_Q d$ cu un număr mai mic de paşi ceea ce prin ipoteza de inducţie implică

$$A_\sigma(a_1, \ldots, a_{i-1}, u, a_{i+1}, \ldots, a_n) \xrightarrow{*}_Q A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n).$$

Deoarece $\longrightarrow_Q$ este compatibilă pe componente cu operaţiile, din $a \longrightarrow_Q u$ rezultă că

$$A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) \longrightarrow_Q A_\sigma(a_1, \ldots, a_{i-1}, u, a_{i+1}, \ldots, a_n).$$

Prin tranzitivitate obţinem $A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) \xrightarrow{*}_Q A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n)$.

Evident $Q \subseteq \longrightarrow_Q \subseteq \xrightarrow{*}_Q$.

Fie R o preordine compatibilă cu operaţiile care include Q. Compatibilitatea cu operaţiile implică $\longrightarrow_Q \subseteq R$, de unde deducem $\xrightarrow{*}_Q \subseteq R$ deoarece R este reflexivă şi tranzitivă.

Notăm $u \downarrow_Q v$ dacă există $a \in \mathcal{A}$ astfel încât $u \xrightarrow{*}_Q a$ şi $v \xrightarrow{*}_Q a$.

5.2.3 Γ–rescriere

Fixăm Γ

Definim prin inducţie şirul crescător de mulţimi de perechi de elemente din $\mathcal{A}$.

$Q_0 = \emptyset$

$Q_{n+1} = \{(h(l), h(r))\ :\ (\forall Y)l =_s r\ \textbf{if}\ H \in \Gamma,\ h : T_\Sigma(Y) \to \mathcal{A},\ \text{şi}\ (\forall(u, v) \in H)h_s(u) \downarrow_{Q_n} h_s(v)\}$

Prin definiţie Q este reuniunea şirului crescător definit mai sus.

În cazul în care Q este definită ca mai sus, în loc de $\stackrel{*}{\longrightarrow}_Q$ vom prefera să scriem $\stackrel{*}{\Longrightarrow}_\Gamma$.
Relaţia $\stackrel{*}{\Longrightarrow}_\Gamma$ este denumită Γ–rescriere sau mai scurt rescriere.

Propoziţie 5.2.9 $\stackrel{*}{\Longrightarrow}_\Gamma$ *este închisă la* $\mathbf{SRew}_\Gamma$.

Demonstraţie: Fie $(\forall Y)l =_s r$ **if** $H \in \Gamma$, $h : T_\Sigma(Y) \to \mathcal{A}$ morfism cu proprietatea $h_s(u) \downarrow_Q h_s(v)$ pentru orice $(u, v) \in H$ şi un context $c \in \mathcal{A}[z]_{s'}$. Trebuie arătat că $c[h(l)] \stackrel{*}{\Longrightarrow}_\Gamma c[h(r)]$.

Deoarece H este finită şi numărul pasilor utilizat în $h_s(u) \downarrow_Q h_s(v)$ unde $(u, v) \in H$ este finit există un n natural astfel încât $h_s(u) \downarrow_{Q_n} h_s(v)$ pentru orice $(u, v) \in H$. Prin urmare $(h(l), h(r)) \in Q_{n+1}$. Deoarece $(h(l), h(r)) \in Q$ deducem $c[h(l)] \stackrel{*}{\Longrightarrow}_\Gamma c[h(r)]$. $\square$

Propoziţie 5.2.10 $\stackrel{*}{\Longrightarrow}_\Gamma$ *este cea mai mică relaţie închisă la* **R**, **T**, $\mathbf{C\Sigma}$ *şi* $\mathbf{Rew}_\Gamma$.

Demonstraţie: *Evident* $\stackrel{*}{\Longrightarrow}_\Gamma$ *este închisă la* **R**, **T** *şi* $\mathbf{C\Sigma}$ *şi, fiind închisă la* $\mathbf{SRew}_\Gamma$, *este închisă şi la* $\mathbf{Rew}_\Gamma$.

Fie W o relaţie închisă la **R**, **T**, $\mathbf{C\Sigma}$ *şi* $\mathbf{Rew}_\Gamma$. *Demonstrăm prin inducţie după n că $Q_n \subseteq W$.*
Dacă $n = 0$ avem $Q_0 = \emptyset \subseteq W$.
Pentru $n \geq 1$ fie $(h(l), h(r)) \in Q_n$, unde $(\forall Y)l =_s r$ **if** *$H \in \Gamma$, $h : T_\Sigma(Y) \to \mathcal{A}$ şi $(\forall(u, v) \in H)h_s(u) \downarrow_{Q_{n-1}} h_s(v)$. Din ipoteza de inducţie $Q_{n-1} \subseteq W$. Cum W este închisă la $C\Sigma$ rezultă că W este închisă la contexte, deci $\longrightarrow_{Q_{n-1}} \subseteq W$. Folosind faptul că W este închisă la* **R** *şi* **T** *rezultă că $\stackrel{*}{\longrightarrow}_{Q_{n-1}} \subseteq W$. Din $(\forall(u, v) \in H)h_s(u) \downarrow_{Q_{n-1}} h_s(v)$ obţinem $(\forall(u, v) \in H)h_s(u) \downarrow_W h_s(v)$. Folosind închiderea lui W la* $\mathbf{Rew}_\Gamma$ *rezultă că $(h(l), h(r)) \in W$, deci $Q_n \subseteq W$.*

Prin urmare $Q \subseteq W$ şi folosind propoziţia 5.2.8 deducem că $\stackrel{*}{\longrightarrow}_Q \subseteq W$.

5.3 Corectitudinea rescrierii

Toate regulile de deducţie de mai sus sunt corecte pentru $\equiv_\Gamma$. Vom demontra acest fapt numai pentru cele mai importante.

Lemă 5.3.1 *Regulile* **R**, **T** *şi* $\mathbf{C\Sigma}$ *sus sunt corecte pentru* $\equiv_\Gamma$.

Demonstraţie: *Relaţia $\equiv_\Gamma$ este o congruenţă, prin urmare este reflexivă, tranzitivă şi compatibilă cu operaţiile ($\forall\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$, $a_i \equiv_\Gamma b_i$ pentru orice $i \in [n]$ implică $A_\sigma(a_1, a_2, \ldots, a_n) \equiv_\Gamma A_\sigma(b_1, b_2, \ldots, b_n)$). De aici rezultă că $\equiv_\Gamma$ este închisă la regulile de deducţie* **R**, **T** *şi* $\mathbf{C\Sigma}$, *adică ele sunt corecte pentru* $\equiv_\Gamma$.

Într-o lecţie anterioară s-a demonstrat că $\mathbf{Sub}_\Gamma$ şi $\mathbf{Rew}_\Gamma$ sunt corecte pentru $\equiv_\Gamma$. Ţinând cont de observaţiile de mai sus putem deduce:

Corolar 5.3.2 $\stackrel{*}{\Rightarrow}_\Gamma \subset \equiv_\Gamma$. *(Corectitudinea rescrierii pentru* $\equiv_\Gamma$*)*

În general rescrierea nu este completă pentru $\equiv_\Gamma$.

5.4 Întâlnirea prin rescriere. Corectitudine şi completitudine

Fie $\downarrow_\Gamma$ relaţia de întâlnire ataşată lui $\stackrel{*}{\Rightarrow}_\Gamma$. Prin definiţie $a \downarrow_\Gamma b$ dacă şi numai dacă există c astfel încât $a \stackrel{*}{\Rightarrow}_\Gamma c$ şi $b \stackrel{*}{\Rightarrow}_\Gamma c$. Observăm că $\downarrow_\Gamma \subseteq \equiv_\Gamma$. (Corectitudinea întâlnirii pentru $\equiv_\Gamma$)

Propoziţie 5.4.1 *$\downarrow_\Gamma$ este închisă la $\mathbf{Sub}_\Gamma$.*

Demonstraţie: Fie $(\forall X)l =_s r$ **if** H în Γ şi $h : T_\Sigma(X) \longrightarrow \mathcal{A}$ un morfism cu proprietatea că $h_t(u) \downarrow_\Gamma h_t(v)$ pentru orice $u =_t v \in H$. Prin urmare pentru orice $u =_t v \in H$ există c_{uv} cu proprietăţile $h(u) \stackrel{*}{\Rightarrow}_\Gamma c_{uv}$ şi $h(v) \stackrel{*}{\Rightarrow}_\Gamma c_{uv}$.

Cu $\mathbf{Rew}_\Gamma$ deducem $h(l) \stackrel{*}{\Rightarrow}_\Gamma h(r)$ deci $h(l) \downarrow_\Gamma h(r)$. $\Box$

Reamintim că dacă $\sim$ este o congruenţă închisă la substituţii, atunci $\mathcal{A}/\sim \models \Gamma$.

Definiţia 5.4.2 O relaţie $\succ$ pe o mulţime A se numeşte **confluentă** dacă

$$(\forall a, b, c \in A)\{[a \succ b \text{ şi } a \succ c] \Rightarrow (\exists d \in A)[b \succ d \text{ şi } c \succ d]\}. \Box$$

Propoziţie 5.4.3 *Dacă $\succ$ este o preordine confluentă pe mulţimea A, atunci relaţia $\downarrow$ definită prin*

$$a \downarrow b \Longleftrightarrow (\exists c \in A) a \succ c \text{ şi } b \succ c$$

este cea mai mică echivalenţă pe A care include $\succ$.

Demonstraţie:

- Reflexivitatea rezultă luând $c := a$.
- Simetria este evidentă.
- Probăm tranzitivitatea. Fie $a \downarrow b$ şi $b \downarrow c$. Observăm că există $d, e \in A$ astfel încât $a \succ d$, $b \succ d$, $b \succ e$ şi $c \succ e$. Din confluenţă rezultă existenţa lui $f \in A$ astfel încât $d \succ f$ şi $e \succ f$. Rezultă că $a \succ f$ şi $c \succ f$, deci $a \downarrow c$.
- Pentru a dovedi că $\succ \subseteq \downarrow$, presupunând că $a \succ b$ şi observând că $b \succ b$ deducem $a \downarrow b$
- Fie $\equiv$ o relaţie de echivalenţă pe A care include $\succ$. Probăm că $\equiv$ include $\downarrow$. Presupunând că $a \downarrow b$ rezultă existenţa lui $c \in A$ astfel încât $a \succ c$ şi $b \succ c$. Deducem $a \equiv c$ şi $b \equiv c$, deci $a \equiv b$. $\Box$

Observaţia 5.4.4 *Dacă $\succ$ este o preordine confluentă şi compatibilă pe o Σ-algebră multisortată, atunci $\downarrow$ este o congruenţă.*

Demonstraţie: Fie $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $a_i \downarrow b_i$ pentru orice $i \in [n]$. Pentru orice $i \in [n]$, există c_i astfel încât $a_i \succ c_i$ şi $b_i \succ c_i$, prin urmare

$$A_\sigma(a_1, a_2, \dots, a_n) \succ A_\sigma(c_1, c_2, \dots, c_n) \text{ şi } A_\sigma(b_1, b_2, \dots, b_n) \succ A_\sigma(c_1, c_2, \dots, c_n).$$

Deci $A_\sigma(a_1, a_2, \dots, a_n) \downarrow A_\sigma(b_1, b_2, \dots, b_n)$. $\Box$

Teorema 5.4.5 *Dacă $\stackrel{*}{\Rightarrow}_\Gamma$ este confluentă, atunci $\downarrow_\Gamma$ este completă.*

Demonstraţie: Confluenţa lui $\stackrel{*}{\Rightarrow}_\Gamma$ este necesară pentru ca $\downarrow_\Gamma$ să devină congruenţă. Deoarece $\downarrow_\Gamma$ este închisă la $\mathbf{Sub}_\Gamma$ deducem că $\mathcal{A}/\downarrow_\Gamma \models \Gamma$.

Fie $a \equiv_\Gamma b$. Utilizând morfismul de factorizare $f : \mathcal{A} \to \mathcal{A}/\downarrow_\Gamma \models \Gamma$ deducem $f(a) = f(b)$ deci $a \downarrow_\Gamma b$. $\Box$

Chapter 6

COMPLETITUDINEA PARAMODULATIEI

6.1 Prolog

Reamintim că Δ înseamnă o mulţime de egalităţi adevărate.

Teoremă 6.1.1 *Dacă* $a \downarrow_\Gamma d$*, atunci* $\{a \doteq_s d\} \xrightarrow{*}_{pr} \Delta$.

Demonstratie Presupunem $a \downarrow_\Gamma d$. Atunci există v astfel încât $a \stackrel{*}{\Rightarrow}_\Gamma v$ şi $d \stackrel{*}{\Rightarrow}_\Gamma v$. Ţinând cont de definiţia lui $\stackrel{*}{\Rightarrow}_\Gamma$, putem scrie $a \xrightarrow{*}_Q v$ şi $d \xrightarrow{*}_Q v$. Deoarece Q este reuniunea şirului crescător $\{Q_n\}_{n \in N}$, rezultă că există un număr natural n cu proprietatea $a \xrightarrow{*}_{Q_n} v$ şi $d \xrightarrow{*}_{Q_n} v$, deci $a \downarrow_{Q_n} d$.

Arătăm prin inducţie după n că $\{a \doteq_s d\} \xrightarrow{*}_{pr} \Delta$. Cazul $n = 0$ este evident. Presupunem că pentru orice x, y dacă $x \downarrow_{Q_n} y$, atunci $\{x \doteq_s y\} \xrightarrow{*}_{pr} \Delta$. Presupunem că $a \downarrow_{Q_{n+1}} d$.

Facem o nouă inducţie după numărul paşilor $\rightarrow_{Q_{n+1}}$ folosiţi. Dacă numărul paşilor este 0, atunci $a = d$, concluzia fiind evidentă.

În cazul contrar, presupunem, de exemplu, că $a \rightarrow_{Q_{n+1}} w$ şi $w \downarrow_{Q_{n+1}} d$ cu un pas mai puţin. Din ipoteza de inducţie putem scrie $\{w \doteq_s d\} \xrightarrow{*}_{pr} \Delta$.

Deoarece $a \rightarrow_{Q_{n+1}} w$ există $(\forall Y) l \doteq_{s'} r \ if \ H \in \Gamma$, morfismul $h : T_\Sigma(Y) \rightarrow T_\Sigma(X)$ astfel încât $h(u) \downarrow_{Q_n} h(v)$, pentru orice $u \doteq v \in H$, şi contextul c în $T_\Sigma(X \cup \{z\})$ astfel încât $a = c[h_{s'}(l)]$ şi $w = c[h_{s'}(r)]$. Observăm că $\{c[h_{s'}(l)] \doteq_s d\} \rightarrow_{pr} h(H) \cup \{c[h_{s'}(r)] \doteq_s d\} = h(H) \cup \{w \doteq_s d\}$.

Prin urmare, deoarece $\{w \doteq_s d\} \xrightarrow{*}_{pr} \Delta$, deducem că $\{a \doteq_s d\} \xrightarrow{*}_{pr} h(H) \cup \Delta$. Deoarece $h(u) \downarrow_{Q_n} h(v)$, pentru orice $u \doteq v \in H$, din ipoteza de inducţie deducem $h(H) \xrightarrow{*}_{pr} \Delta$, deci $\{a \doteq_s d\} \xrightarrow{*}_{pr} \Delta$. $\Box$

Corolar 6.1.1 *Dacă G este o mulţime finită astfel încât* $G \subseteq \downarrow_\Gamma$*, atunci* $G \xrightarrow{*}_{pr} \Delta$.

6.2 Completitudinea

Observăm că identitatea lui $T_\Sigma(Y)$ este soluţie pentru $(\exists Y)\Delta$ deoarece $\Gamma \models (\forall Y)\Delta$. Prin urmare, dacă $G \xrightarrow{*}_p \Delta$ cu morfismul calculat σ, atunci σ este o solutie pentru $(\exists X)G$. Prin urmare, putem opri rezolvarea în momentul ajungerii la o mulţime de egalităţi adevărate.

Presupunem că mulţimea Γ de ecuaţii condiţionate satisface următoarele condiţii: $(\forall x)\ x \doteq x \in \Gamma$, pentru orice variabilă x, $(\forall x_1\ \forall x_2 \ldots \forall x_n) f(x_1, x_2, \ldots, x_n) \doteq f(x_1, x_2, \ldots, x_n) \in \Gamma$, pentru orice simbol de operaţie f, şi, pentru orice axiomă $(\forall Y) l \doteq_s r \ if \ H \in \Gamma$, orice variabilă din Y apare în l.

Teoremă 6.2.1 (Teorema de completitudine) *În condiţiile de mai sus, dacă* $\downarrow_\Gamma$ *este completă, atunci orice soluţie poate fi obţinută numai cu regula paramodulaţiei.*

Demonstraţie Fie $\sigma : T_\Sigma(X) \rightarrow T_\Sigma(Y)$ o soluţie pentru $(\exists X)G$, adică $\Gamma \models (\forall Y)\sigma(G)$. Prin urmare, $\sigma(G)$ este o submulţime a congruenţei semantice, adică $\sigma(G) \subseteq \equiv_\Gamma$.

Deoarece $\downarrow_\Gamma$ este completă, adică $\downarrow_\Gamma = \equiv_\Gamma$, deducem că $\sigma(G) \subseteq \downarrow_\Gamma$. Conform Prologului obţinem $\sigma(G) \xrightarrow{*}_{pr} \Delta$.

Deoarece pentru orice axiomă $(\forall Y) l \doteq_s r \ if \ H \in \Gamma$, orice variabilă din Y apare în l, deducem, din Propoziţia 4.3.2, că orice pararescriere este un caz particular de paramodulaţie în care substituţia calculată este o identitate. În concluzie, putem scrie $\sigma(G) \xrightarrow{*}_p \Delta$ cu substituţia calculată identitatea.

Din Lema substituţiei deducem că $G \xrightarrow{*}_p \sigma(G)$ cu substituţia calculată σ.

Deci $G \xrightarrow{*}_p \Delta$ cu substituţia calculată σ. $\square$

Chapter 7

COMPLETITUDINEA NARROWINGULUI

7.1 Forme normale

Reamintim că $\stackrel{*}{\Rightarrow}_\Gamma$ este relaţia de rescriere în A.

Definiţia 7.1.1 Elementul $n \in A$ se numeşte **o formă normală** pentru dacă

$$(\forall b \in A)(n \stackrel{*}{\Rightarrow}_\Gamma b \text{ implică } n = b).$$

Fie N mulţimea elementelor din A care sunt forme normale. Presupunem **axioma Formei Normale unice**

$$\textbf{FN}!\ \ (\forall a \in A)(\exists!\, fn(a) \in N) a \stackrel{*}{\Rightarrow}_\Gamma fn(a).$$

Observaţia 7.1.2 *Dacă $a \stackrel{*}{\Rightarrow}_\Gamma d$, atunci $fn(a) = fn(d)$.*

Demonstraţie: Din ipoteză şi $d \stackrel{*}{\Rightarrow}_\Gamma fn(d)$ deducem $a \stackrel{*}{\Rightarrow}_\Gamma fn(d)$, deci din unicitatea formei normale a lui a deducem $fn(a) = fn(d)$.

Observaţia 7.1.3 *Axioma* **FN!** *implică $\stackrel{*}{\Rightarrow}_\Gamma$ este confluentă.*

Demonstraţie: Presupunem $a \stackrel{*}{\Rightarrow}_\Gamma d$ şi $a \stackrel{*}{\Rightarrow}_\Gamma c$. Deducem $fn(a) = fn(d) = fn(c)$, deci $d \stackrel{*}{\Rightarrow}_\Gamma fn(d)$ şi $c \stackrel{*}{\Rightarrow}_\Gamma fn(d)$.

Observaţia 7.1.4 *Funcţia $fn\colon A \longrightarrow N$ este surjectivă şi*

$$a \downarrow_\Gamma d \Leftrightarrow fn(a) = fn(d).$$

Demonstraţie: Deoarece pentru orice element n în formă normală $n = fn(n)$ rezultă surjectivitatea funcţiei fn.

Presupunem $fn(a) = fn(d)$. Deoarece $a \stackrel{*}{\Rightarrow}_\Gamma fn(a)$ şi $d \stackrel{*}{\Rightarrow}_\Gamma fn(a)$ deducem $a \downarrow_\Gamma d$.

Presupunem $a \downarrow_\Gamma d$. Fie $c \in A$ astfel încât $a \stackrel{*}{\Rightarrow}_\Gamma c$ şi $d \stackrel{*}{\Rightarrow}_\Gamma c$. Deducem $fn(a) = fn(c)$ şi $fn(d) = fn(c)$, deci $fn(a) = fn(d)$. $\Box$

Pentru cazul algebrelor libere mai menţionăm că orice subexpresie a unei forme normal este tot o formă normală.

7.2 Introducere

Se lucrează în algebre libere. Vom nota cu $T_\Sigma(X)$ algebra din care începem să lucrăm.

Fie $(\forall Y) l \doteq_s r$ **if** $H \in \Gamma$ o clauză. Mulţimea de variabile Y va fi disjunctă de X. Vom presupune că $T_\Sigma(X)$ şi $T_\Sigma(Y)$ sunt subalgebre în $T_\Sigma(X \cup Y)$ şi notăm cu i_X şi i_Y morfismele incluziune.

În continuare vom lucra cu un caz particular de paramodulaţie denumit narrowing sau îngustare.

Narrowing(Îngustare): Fie $(\forall Y)l \doteq_s r$ **if** $H \in \Gamma$ şi $\theta = CGU(a,l) : T_\Sigma(X \cup Y) \to \mathcal{B}$ unde $a \in T_\Sigma(X)$ **nu este o variabilă**. Dacă G este o mulţime de egalităţi formale şi C este un context extins din $T_\Sigma(X \cup \{z\})$, atunci

$$G \cup \{C[a]\} \quad \longrightarrow_n \quad \theta(G \cup H \cup \{C[r]\})$$

cu morfismul calculat $i_X; \theta$.$\Box$

Menţionăm că ipoteza care apare mai jos şi anume că membrul stâng al concluziei unei axiome nu este o variabilă este o ipoteza naturală deoarece în caz contrar dacă condiţiile axiomei sunt verificate, atunci orice termen ar putea fi rescris.

Propoziţie 7.2.1 *Dacă pentru orice clauză $(\forall Y)l \doteq_s r$ **if** H din Γ l nu este variabilă şi orice variabilă din Y apare în l, atunci pararescrierea este un caz particular de îngustare în care substituţia calculată este o identitate.*

Demonstraţie: Este suficient să reluăm demonstraţiile propoziţiilor 4.3.1 şi 4.3.2 şi din egalitatea $a = h_s(l)$ din faptul că l nu este o variabilă rezultă că nici a nu este variabilă.

7.3 Lema de ridicare

Definiţia 7.3.1 O substituţie se numeşte normală dacă duce orice variabilă într-un element în formă normală.

Propoziţie 7.3.2 *Presupunem că mulţimile de variabile X şi Y sunt disjuncte şi că $T_\Sigma(X)$ şi $T_\Sigma(Y)$ sunt subalgebre în $T_\Sigma(X \cup Y)$.*

Fie $l \in T_\Sigma(Y)$ astfel încât orice variabilă din Y apare în l şi $a \in T_\Sigma(X)$. Fie $\rho : X \cup Y \to T_\Sigma Z$ o substituţie a cărei restricţie la X este normală şi $\rho(a) = \rho(l)$.

Dacă $\psi = CGU(a,l) : X \cup Y \longrightarrow T_\Sigma V$ şi θ este unica substituţie pentru care $\rho = \psi; \theta$, atunci θ este normală.

Demonstraţie: Existenţa lui $CGU(a,l)$ rezultă din faptul că ρ este unificator pentru a şi l. În plus, fără a restrânge generalitatea, putea să presupunem că $V \subseteq X \cup Y$ şi că $\psi(v) = v$ pentru orice $v \in V$.

Fie $v \in V$. Vom studia două cazuri.

1. Dacă $v \in X$, atunci $\theta(v) = \theta(\psi(v)) = \rho(v)$ este normală prin ipoteză.

2. Presupunem $v \in Y$. Deoarece $a \in T_\Sigma(X)$ şi $v \notin X$ rezultă că variabila v nu apare în a. Deoarece Y este mulţimea variabilelor care apar în l, v apare în l. Dar $\psi(v) = v$ implică apariţia lui v în $\psi(l) = \psi(a)$. Deoarece variabila v nu apare în a rezultă că v a fost introdus în $\psi(a)$ prin substituţia ψ, deci există $x \in X$ o variabilă care apare în a, astfel încât v apare în $\psi(x)$. Prin urmare $\theta(v)$ este subtermen în $\theta(\psi(x)) = \rho(x)$. Deoarece $\rho(x)$ este prin ipoteză o forma normală, rezultă că orice subtermen al său este o formă normală. In particular $\theta(v)$ este o formă normală. $\Box$

Dacă $\theta : X \to Z$ şi $\varphi : Y \to Z$ sunt două funcţii cu domeniile disjuncte notăm cu $< \theta, \varphi >: X \cup Y \to Z$ unica funcţie care pe X acţionează ca θ şi pe Y ca φ.

Propoziţie 7.3.3 *Pentru orice $(\forall Y)l = r$ if H din Γ presupunem că orice variabilă din Y apare în l. Fie G o mulţime de egalităţi din $T_\Sigma(X)$.*

Dacă $\theta : T_\Sigma(X) \to T_\Sigma(Z)$ este normală şi

$$\theta(G) \longrightarrow_{pr} Q,$$

atunci $\theta = \varphi\theta'$ cu θ' normală, există R cu $\theta'(R) = Q$ şi

$$G \longrightarrow_n R \text{ cu substituţia calculată } \varphi.$$

Demonstraţie: Fie $(\forall Y)l \doteq r$ if $H \in \Gamma$ regula şi $\eta : T_\Sigma(Y) \longrightarrow T_\Sigma(Z)$ substituţia utilizate în pararescrieie. Vom presupune că variabilele din Y sunt noi, adică Y este disjunct de X.

Datorită normalităţii lui θ pararescrierea nu se poate face într-un subtermen de forma $\theta(x)$ unde x este o variabilă, aşa că presupunem că ea se face în $\theta(a) = \eta(l)$ unde a este un subtermen în G care nu este variabilă. Prin urmare:

$$G = G' \cup \{C[a]\} \quad \text{şi} \quad Q = \theta(G') \cup \eta(H) \cup \{\theta^z(C)[\eta(r)]\}$$

unde z este o variabilă nouă, C este un context extins din $T_\Sigma(X \cup \{z\})$ şi $\theta(a) = \eta(l)$.

Fie $\psi = CGU(a,l) : X \cup Y \to V$ şi $\theta' : V \to Z$ unica substituţie cu proprietatea $\psi\theta' =< \theta, \eta >$. Deoarece restricţia θ a lui $< \theta, \eta >$ la X este normală conform ipotezei, şi $< \theta, \eta > (a) =< \theta, \eta > (l)$, aplicând propoziţia 7.3.2 rezultă normalitatea lui θ'.

Notând cu $\varphi : T_\Sigma(X) \to T_\Sigma(V)$ restricţia lui ψ la $T_\Sigma(X)$ deducem că $\theta = \varphi\theta'$.

Rezultă că

$$G \longrightarrow_n \psi(G' \cup H \cup \{C[r]\}) \text{ cu morfismul calculat } \varphi.$$

Notând $R = \psi(G' \cup H \cup \{C[r]\})$ mai observăm că:

$$\theta'(R) = (\psi;\theta')(G' \cup H \cup \{C[r]\}) =< \theta, \eta > (G' \cup H \cup \{C[r]\}) = \theta(G') \cup \eta(H) \cup \{\theta^z(C)[\eta(r)]\} = Q. \ \Box$$

$$\begin{tikzcd}
G \arrow[d, "n"'] & X \arrow[r, "\theta"] \arrow[d, "\varphi"] & Z \arrow[d, "1_Z"'] & \theta(G) \arrow[d, "pr"'] \\
R \arrow[d, "*", "n"'] & V' \arrow[r, "\theta'"] \arrow[d, "\sigma"] & Z \arrow[d, "1_Z"'] & Q = \theta'(R) \arrow[d, "*", "pr"'] \\
G' & V \arrow[r, "\epsilon"] & Z & S
\end{tikzcd}$$

Propoziţie 7.3.4 *Pentru orice* $(\forall Y)l \doteq r$ *if* H *din* Γ *presupunem că orice variabilă din* Y *apare în* l. *Fie* G *o mulţime finită de egalităţi din* $T_\Sigma(X)$.

Dacă $\theta : T_\Sigma(X) \to T_\Sigma(Z)$ *este normală şi*

$$\theta(G) \xrightarrow{*}_{pr} S,$$

atunci

$$G \xrightarrow{*}_n G' \text{ cu morfismul calculat } \sigma$$

pentru care există o substituţie normală ϵ *cu proprietăţile* $\epsilon(G') = S$ *şi* $\theta = \sigma\epsilon$.

Demonstraţie: Prin inducţie după numărul paşilor. Vom pune în evidenţă prima pararescriere

$$\theta(G) \longrightarrow_{pr} Q \xrightarrow{*}_{pr} S.$$

Conform propoziţiei precedente $\theta = \varphi\theta'$ cu θ' normală, există R cu $\theta'(R) = Q$ şi

$$G \longrightarrow_n R \text{ cu morfismul calculat } \varphi.$$

Folosind ipoteza de inducţie din $\theta'(R) \xrightarrow{*}_{pr} S$ deducem

$$R \xrightarrow{*}_n G' \text{ cu morfismul calculat } \sigma$$

pentru care există o substituţie normală ϵ cu proprietăţile $\epsilon(G') = S$ şi $\theta' = \sigma\epsilon$. Din cele de mai sus rezultă că

$$G \xrightarrow{*}_n G' \text{ cu morfismul calculat } \varphi\sigma$$

şi $(\varphi\sigma)\epsilon = \varphi\theta' = \theta$. $\Box$

7.4 Epilog

Propoziţie 7.4.1 *Fie $G \subseteq T_\Sigma(X) \times T_\Sigma(X)$ finită şi morfismul $h : T_\Sigma(X) \to T_\Sigma(Y)$ cu $h(G) = \Delta$. Atunci $G \xrightarrow{*}_r \emptyset$ cu substituţia calculată s' pentru care există morfismul f cu proprietatea $s'; f = h$.*

Demonstraţie: Reamintim definiţia reflexiei:

"Dacă $\theta : \mathcal{A} \to \mathcal{B}$ este cel mai general unificator pentru l si r, atunci $G \cup \{l =_t r\} \quad \longrightarrow_r \quad \theta(G)$ cu mofismul calculat θ."

Vom demonstra prin inducţie după numărul elementelor mulţimii G.

Fie $G = G' \cup \{l =_s r\}$. Din ipoteză $h(l) = h(r)$. Fie $u : T_\Sigma(X) \to T_\Sigma(Z)$ cel mai general unificator pentru l şi r. Atunci există un unic $v : T_\Sigma(Z) \to T_\Sigma(Y)$ astfel încât $u; v = h$.

Observăm conform definiţiei de mai sus că $G \longrightarrow_r u(G')$ cu morfismul calculat u.

Cum $v(u(G')) = h(G') \stackrel{ip}{=} \Delta$, aplicând ipoteza de inducţie pentru $u(G')$ deducem că $u(G) \xrightarrow{*}_r \emptyset$ cu substituţia calculată w pentru care există f astfel încât $w; f = v$.

Atunci $G \xrightarrow{*}_r \emptyset$ cu substituţia calculată $u; w$. În plus $h = u; v = (u; w); f$. $\Box$

7.5 Completitudine

Ipoteze. Pentru orice $(\forall Y) l \doteq r$ if H din Γ presupunem că orice variabilă din Y apare în l. Rescrierea are proprietatea formei normale unice (**FN!**).

Reamintim că proprietatea **FN!** implică confluenţa rescrierii şi completitudinea relaţiei de întâlnire prin rescriere.

Fie $s : X \to Z$ o soluţie pentru $(\exists X)G$. Cu ipoteza **FN!** pentru Γ soluţia s se poate normaliza obţinând soluţia normală $s' : X \to Z$ definită prin $s'(x) = fn(s(x))$ pentru orice $x \in X$. Observăm că $s(x) \stackrel{*}{\Rightarrow}_\Gamma s'(x)$ pentru orice $x \in X$. Prin inducţie structurală se arată uşor că $s(r) \stackrel{*}{\Rightarrow}_\Gamma s'(r)$ pentru orice $r \in T_\Sigma(X)$. Pentru orice $u =_t v \in G$ observăm că

$$s(u) \stackrel{*}{\Rightarrow}_\Gamma s'(u) \quad \text{şi} \quad s(v) \stackrel{*}{\Rightarrow}_\Gamma s'(v).$$

Probăm că s' este soluţie pentru $(\exists X)G$ adică $\Gamma \models (\forall Z) s'(G)$. Fie $u \doteq_t v \in G$. Deoarece s este soluţie pentru $(\exists X)G$ deducem $\Gamma \models (\forall Z) s(u) \doteq_t s(v)$, deci $s(u) \equiv_\Gamma s(v)$. Dar $s(u) \equiv_\Gamma s'(u)$ şi $s(v) \equiv_\Gamma s'(v)$ deci $s'(u) \equiv_\Gamma s'(v)$ adică $\Gamma \models (\forall Z) s'(u) =_t s'(v)$ (am folosit $\stackrel{*}{\Rightarrow}_\Gamma \subseteq \equiv_\Gamma$ şi $\equiv_\Gamma$ tranzitivă). Rezultă că $\Gamma \models (\forall Z) s'(G)$, deci s' este soluţie.

Propoziţie 7.5.1 *Orice soluţie normală se obţine prin particularizarea unei soluţii obţinute cu narrowing şi reflexie.*

Demonstraţie: Fie $s' : T_\Sigma(X) \to T_\Sigma(Z)$ o soluţie normală. Utilizând completitudinea relaţiei de întâlnire prin rescriere din $\Gamma \models (\forall Z) s'(G)$ rezultă că $s'(G) \subseteq \downarrow_\Gamma$, prin urmare conform prologului $s'(G) \xrightarrow{*}_{pr} \Delta$. Din lema de ridicare rezultă existenţa substituţiei σ cu

$$G \xrightarrow{*}_n G' \text{cu morfismul calculat } \sigma$$

şi a substituţiei normale ϵ cu proprietăţile $\epsilon(G') = \Delta$ şi $s' = \sigma\epsilon$.

Deoarece G' este o mulţime de egalităţi unificabile prin ϵ, deducem conform epilogului

$$G' \longrightarrow_r \emptyset \text{ cu morfismul calculat } \theta$$

şi există o substituţie ζ cu proprietatea $\epsilon = \theta\zeta$. Deoarece $s' = \sigma\theta\zeta$ deducem că orice soluţie normală s' poate fi obţinută prin particularizarea unei soluţii $\sigma\theta$ găsite cu narrowing şi reflexie.

Chapter 8

REZOLUŢIE À LA PROLOG

Programarea logică relaţională, ilustrată în viaţa de toate zilele de limbajul Prolog, este bazată pe rezoluţie.

8.1 Rezoluţia

Axiomele, clauze Horn, au forma $(\forall Y)\pi(v)$ **if** H unde
1) $\pi(v)$ este un atom, adică π este un predicat şi v este un vector de termeni în concordanţă cu aritatea lui π iar
2) H este o mulţime de atomi.

Ţelul este o mulţime de atomi. Punând în evidenţă atomul asupra căruia va acţiona rezoluţia pentru o axiomă ca mai sus ţelul devine $\{\pi(s)\} \cup T$. Ca mai sus presupunem că variabilele din Y sunt disjuncte de variabilele din ţel.

Fie $\theta = CGU(v, s)$. Prin rezoluţie, cu substituţia calculată θ ajungem la ţelul $\theta(H \cup T)$.

8.2 Rezoluţie = Narrowing = Paramodulaţie

Trecerea de la varianta relaţională la varianta ecuaţională se face prin
1) adăugarea la signatură a sortului b, transformarea predicatelor în simboluri de operaţii având rezultatul de sort b
2) înlocuirea fiecărui atom $\pi(v)$ cu egalitatea $\pi(v) \doteq_b t$ unde t este o constantă de sort b reprezentând adevărul.

Varianta ecuaţională a unei mulţimi de atomi C va fi notată cu $C^e = \{\pi(v) \doteq_b t \,:\, \pi(v) \in C\}$.

Propoziţie 8.2.1 *În varianta ecuaţională, rezoluţia se poate realiza prin narrowing şi eliminarea egalităţilor reale.*

Demonstraţie: Fie G o mulţime de atomi şi $(\forall Y)\,\pi(v)\,\mathrm{if\,H}$ o clauză Horn. Considerăm $G = G' \cup \{\pi(s)\}$ şi $\theta = CGU(v, s)$. Prin rezoluţie obţinem $\theta(G' \cup H)$.

În varianta ecuaţională $G^e = G'^e \cup \{\pi(s) \doteq_b t\}$ şi $(\forall Y)\,\pi(v) \doteq_b t\,\mathrm{if\,H^e}$.
Alegem $a = \pi(s)$, $l = \pi(v)$ şi observăm că $\theta = CGU(s, v) = CGU(\pi(s), \pi(v))$.
Cum a nu este variabilă rezultă că putem aplica narrowing-ul:

$$\begin{array}{lll} G'^e \cup \{\pi(s) \doteq_b t\} & \longrightarrow_n & \theta(G'^e \cup H^e \cup \{t \doteq_b t\}) = \\ & & \theta((G' \cup H)^e \cup \{t \doteq_b t\}) = \\ & & [\theta(G' \cup H)]^e \cup \{t \doteq_b t\} \end{array}$$

În urma eliminării egalităţii adevărate $t \doteq_b t$ obţinem varianta ecuaţională a rezultatului rezoluţiei, $\theta(G' \cup H)$.

Corolar 8.2.1 *Fie G o mulţime de atomi. Orice soluţie pentru $(\exists X)G$ obţinută cu rezoluţia poate fi obţinută prin narrowing şi eliminarea egalităţilor adevărate ca soluţie pentru $(\exists X)G^e$*

Propoziţie 8.2.2 *Fie Γ o mulţime de clauze Horn şi G o mulţime de atomi. Aplicarea narrowing-ului folosind Γ^e în varianta ecuaţională G^e se poate realiza prin rezoluţie folosind Γ în G.*

Demonstraţie: Fie $G = G' \cup \{P(s)\}$ şi $(\forall Y)\,\pi(v) \doteq_b t\,\mathrm{if\,H^e}$ o clauză astfel încât să se poată aplica narrowing-ul lui $P(s) \doteq_b t$. Atunci $l = \pi(v)$ şi există $\theta = CGU(l, a)$, unde a trebuie ales.

Singura variantă posibilă pentru a este $a = P(s)$. Observăm că a nu este variabilă.
Cum există $\theta = CGU(a, l)$ rezultă că $P = \pi$ şi $\theta = CGU(v, s)$. Prin urmare

$$G^e \rightarrow_n \theta(G'^e \cup \{t \doteq_b t\} \cup H^e) = \theta(G' \cup H)^e \cup \{t \doteq_b t\}.$$

Aplicând rezoluţia obţinem

$$G' \cup \{P(s)\} \longrightarrow \theta(G' \cup H) \text{ cu morfismul calculat } \theta$$

În concluzie din $G^e \longrightarrow_n G_1$ cu Γ^e şi morfismul calculat θ, deducem că G se duce prin rezoluţie cu Γ şi morfismul calculat θ în F cu $G_1 = F^e \cup \{t \doteq_b t\}$. $\square$

Corolar 8.2.2 *Fie G o mulţime de atomi. Orice soluţie pentru $(\exists X)G^e$ obţinută prin narrowing şi eliminarea egalităţilor adevărate poate fi obţinută cu rezoluţia ca soluţie pentru $(\exists X)G$.*

Mai observăm că în varianta ecuaţională a unui program Prolog reflexia nu poate fi aplicată deoarece unificarea nu poate fi făcută în egalitatea $\pi(v) = t$.

Concluzia este că rezoluţia este completă, fapt ce rezultă din teoremele de completitudine demonstrate anterior.

8.3 Exercitiul 1

Se păstrează notaţiile din capitolele precedente. Se dă următorul fragment de program EQLOG:

```
sort nat < nlist < list
op 0 : -> nat
op s : nat -> nat
op nil : -> list
op _ _ : list list -> list [assoc]
op cap : nlist -> nat
op cdr : nlist -> list
var E : nat
var L : list
eq cap(E L) = E              ***> 1
eq cdr(E L) = L              ***> 2
op # : list -> nat
eq #(nil) = 0                ***> 3
eq #(E L) = s(#(L))          ***> 4
```

Se cere să se găsească soluţie pentru următoarea interogare:

$$\exists L\{\#(L) = s(s(0)), cap(L) = 0\}$$

Rezolvare. Avem 3 variante:

- să unificăm membrul stâng din ecuaţia 1 cu $cap(L)$;
- să unificăm membrul stâng din ecuaţia 3 cu $\#(L)$;
- să unificăm membrul stâng din ecuaţia 4 cu $\#(L)$.

Vom alege ultima alternativă. Deoarece ecuaţia 4 are variabile care apar în scop, redenumim variabilele şi obţinem: $\#(E\ L1) = s(\#(L1))$.

Identificăm cadrul de aplicare a paramodulaţiei:

- contextul c este $z = s(s(0))$;
- cel mai general unificator pentru $\#(L)$ şi $\#(E\ L1)$ este $L := EL1$;
- ecuaţia care se utilizează nu este condiţionată deci H este vid.

Noul scop este $\{cap(E\ L1) = 0, s(\#(L1)) = s(s(0))\}$.

Subtermenul unde se aplică paramodulaţia este $\#(L1)$, iar ecuaţia folosită este 4. Din nou vom face o redenumire a variabilelor, ecuaţia 4 devine $\#(E1\ L2) = s(\#L2)$.

După noul pas de paramodulaţie, scopul devine: $\{cap(E\ E1\ L2) = 0, s(s(\#(L2))) = s(s(0))\}$. Se observă că este posibil să facem din nou paramodulaţie cu ecuaţia 4 şi putem intra astfel în ciclu infinit.

Vom unifica $cap(E\ E1\ L2)$ cu membrul stâng al ecuaţiei 1, în care redenumim variabilele; cel mai general unificator calculat este $E2 := E, L1 := E1\ L2$. Contextul este $z = 0$.

Scopul devine $\{E = 0, s(s(\#(L2))) = s(s(0))\}$.

Unificăm $\#(L2)$ cu membrul stâng al ecuaţiei 3; contextul este $z = 0$, cel mai general unificator, $L2 := nil$.

Noul scop este $\{E = 0, s(s(0)) = s(s(0))\}$.

Prin aplicarea reflexiei, scopul devine $\emptyset$ şi se adaugă la soluţie $E := 0$.

Soluţia $L = 0\ E1\ nil$ se obţine astfel :

```
L = E L1
  = E E1 L2
  = E E1 nil
  = 0 E1 nil
```

8.4 Exercitiul 2

Se dă următorul fragment de program:

```
0 <= x                          ***> 1
s x <= s y :- x <= y            ***> 2
```

Se cere soluţia pentru:

1. $w <= s\ 0$ (toate soluţiile);
2. $s\ s\ 0 <= w$;
3. $s\ s\ 0 <= w = true$, pentru programul echivalent în EQLOG.

Rezolvare:

1. Prima variantă este de a utiliza prima clauză, cel mai general unificator este $w := 0, x := s\ 0$. Scopul devine $0 <= s\ 0$ care este adevărat din prima clauză; soluţia este $w = 0$.

   A doua variantă este de a utiliza a doua clauză, cel mai general unificator este $w := s\ x, y := 0$. Scopul devine $x <= 0$. Cu prima clauză în care redenumim variabilele obţinem $0 <= 0$, care este adevărat cu cel mai general unificator $x := 0, x' := 0$. Soluţia este $w = s\ 0$.

2. Folosind clauza 2, unificatorul cel mai general este $x := s\ 0, w := s\ y$, iar scopul devine $s\ 0 <= y$. Folosim clauza 2, cu variabilele redenumite deoarece apar în scop; unificatorul este $x' := 0, y := s\ y'$, iar scopul devine $0 <= y'$ care este adevărat conform primei clauze, unificând $x := y'$. Soluţia este $w = s\ s\ y'$.

3. Programul echivalent în EQLOG se obţine înlocuind $\Pi(x_1, ..., x_n)$ cu $\Pi(x_1, ..., x_n) = true$.

   Scopul este $\{s\ s\ 0 <= w = true\}$. Identificăm cadrul de aplicare a narrowing-ului cu ecuaţia a doua:

   - contextul extins este $z = true$;
   - a este $s\ s\ 0 <= w$;
   - l este $s\ x <= s\ y$;
   - cel mai general unificator pentru a şi l este $w := s\ y, x := s\ 0$.

   Scopul devine $\{true = true, s\ 0 <= y = true\}$. Folosim din nou a doua ecuaţie, dar redenumim variabilele:

   - contextul extins este $z = true$;
   - a este $s\ 0 <= y$;
   - l este $s\ x' <= s\ y'$;
   - cel mai general unificator pentru a şi l este $y := s\ y', x' := 0$.

   Scopul devine $\{true = true, true = true, 0 <= y' = true\}$. Folosim prima ecuaţie în care redenumim variabilele:

   - contextul extins este $z = true$;
   - a este $0 <= y'$;
   - l este $0 <= x''$;
   - cel mai general unificator pentru a şi l este $x'' := y'$.

   Scopul devine {true = true, true = true, true = true} şi prin reflexie devine $\emptyset$. Soluţia este $w = s\ s\ y'$.