PROGRAMARE LOGICĂ ECUAŢIONALĂ
ediţia a III-a

Virgil Emil Căzănescu

October 4, 2008

1 SIGNATURI ŞI ALGEBRE MULTISORTATE

Virgil Emil Căzănescu

March 6, 2008

1 Signaturi multisortate

În programare, mai mult decât în orice altă activitate, datele utilizate sunt de mai multe feluri, sau **sorturi** aşa cum vom spune în continuare. Mai mult, de cele mai multe ori, în diferitele construcţii sintactice, într-un anumit loc al acestora nu poate fi plasată decât o dată de un anumit fel(sort). Aceasta ar fi explicaţia faptului că algebrele multisortate constituie una dintre cele mai utile unelte pentru informatica teoretică.

Algebrele la rândul lor nu sunt toate la fel. Felul algebrelor este dat de signatura lor. O signatură are două componenete una pentru date şi una pentru operaţii.

Componenta pentru date este pur şi simplu o mulţime S ale cărei elemente $s \in S$ se numesc sorturi.

Fiecare operaţie este caracterizată de modul acesteia de acţiune. Operaţia acţionează pe un anumit număr fix de date de sorturi precizate şi are rezultatul de un sort dat. Ca exemplu pentru operaţia cu numele o notăm cu

$$o : s_1 s_2 \dots s_n \longrightarrow s$$

faptul ca ea are n argumente de sorturi s_1, s_2, ..., s_n iar rezultatul acesteia este de sort s.

Toate aceste informaţii privind felul algebrei sunt adunate în conceptul de signatură. Cu S^* notăm mulţimea şirurilor finite formate cu elemente din S.

Definiţia 1.1 O signatură

$$(S, \{\Sigma_{s_1 s_2 \dots s_n, s}\}_{s_1 s_2 \dots s_n \in S^*, s \in S})$$

este formată dintr-o muţime S şi o familie de mulţimi

$$\{\Sigma_{s_1 s_2 \dots s_n, s}\}_{s_1 s_2 \dots s_n \in S^*, s \in S}.$$

Pentru fiecare $s_1 s_2 \dots s_n \in S^*$ şi $s \in S$ mulţimea $\Sigma_{s_1 s_2 \dots s_n, s}$ conţine numele operaţiilor cu n argumente de sorturi $s_1 s_2 \dots s_n$ şi rezultat de sort s.

Menţionăm că mulţimile $\Sigma_{s_1 s_2 \dots s_n, s}$ pot avea elemente comune, ceea ce permite modelarea supraîncărcării operatorilor, adică permisiunea ca mai multe operaţii să aibă acelaş nume.

Când nu există pericol de confusie vom scrie pur şi simplu (S, Σ) sau Σ în loc de $(S, \{\Sigma_{s_1 s_2 \dots s_n, s}\}_{s_1 s_2 \dots s_n \in S^*, s \in S})$.

2 Algebre multisortate

Algebrele sunt formate în mare din date şi operaţii. Datele sunt de mai multe sorturi, adică pentru fiecare sort s algebra conţine o mulţime a datelor de sort s. Familia acestor mulţimi, numită şi suportul algebrei, constitue o mulţime sortată.

2.1 Mulţimi şi funcţii multisortate

Fixăm mulţimea S a sorturilor.

Definiţia 2.1 O familie de mulţimi $M = \{M_s\}_{s \in S}$ indexată de S se numeşte mulţime S-sortată.

Observăm că aceeaşi literă este folosită atât pentru întreaga mulţime cât şi pentru toate componentele acesteia.

Conceptele uzuale cu mulţimi se extind pe componente de la mulţimile uzuale la mulţimile S-sortate aşa cum se vede din exemplele de mai jos

$$\{M_s\}_{s \in S} \subseteq \{N_s\}_{s \in S} \text{ dacă şi numai dacă } M_s \subseteq N_s \ \forall s \in S,$$

$$\{M_s\}_{s\in S} \cup \{N_s\}_{s\in S} = \{M_s \cup N_s\}_{s\in S},$$
$$\{M_s\}_{s\in S} \cap \{N_s\}_{s\in S} = \{M_s \cap N_s\}_{s\in S},$$
$$\{M_s\}_{s\in S} \times \{N_s\}_{s\in S} = \{M_s \times N_s\}_{s\in S}.$$

O funcţie între două mulţimi sortate duce un element din prima mulţime într-un element de acelaşi sort din a două mulţime.

Definiţia 2.2 O funcţie S-sortată

$$f : M \longrightarrow N$$

este o familie de funcţii $f = \{f_s\}_{s\in S}$ unde componenta de sort s este o funcţie uzuală $f_s : M_s \longrightarrow N_s$.

Ca şi în cazul mulţimilor S-sortate, operaţiile cu funcţiile S-sortate se fac pe componente. Dacă $f : M \longrightarrow N$ şi $g : N \longrightarrow P$ sunt funcţii S-sortate atunci compunerea lor $f; g : M \longrightarrow P$ este definită prin

$$(f; g)_s = f_s; g_s.$$

Compunerea funcţiilor S-sortate este asociativă.

Pentru orice mulţime S-sortată M funcţia ei identitate $1_M : M \longrightarrow M$ este definită prin $(1_M)_s = 1_{M_s}$ pentru orice $s \in S$. Funcţia identitate are efect neutru la compunere.

2.2 Algebre multisortate

Definiţia 2.3 O Σ-algebră $\mathcal{A} = (\{A_s\}_{s\in S}, \{A_\sigma\}_{\sigma\in\Sigma})$ este formată dintr-o mulţime S-sortată $A = \{A_s\}_{s\in S}$ şi o familie de operaţii $\{A_\sigma\}_{\sigma\in\Sigma}$. Pentru claritate, dacă $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$, adică $\sigma : s_1 s_2 \ldots s_n \longrightarrow s$, atunci

$$A_\sigma : A_{s_1} \times A_{s_2} \times \ldots A_{s_n} \longrightarrow A_s.$$

Dacă nu există pericol de confuzie în loc de $(\{A_s\}_{s\in S}, \{A_\sigma\}_{\sigma\in\Sigma})$ vom scrie mai simplu (A_s, A_σ).

Din definiţia de mai sus rezultă că dacă $\sigma \in \Sigma_{\lambda,s}$ unde λ este şirul vid din S^*, atunci $A_\sigma \in A_s$. Deci operaţiile fără argumente, numite şi **constante**, sunt elemente ale algebrei de sort corespunzător sortului rezultat.

Vom continua prin a defini pentru algebrele multisortate cele mai uzuale concepte specifice algebrei: morfisme, subalgebre, algebre libere, congruenţe.

3 Morfisme de algebre multisortate

Un morfism între două algebre multisortate, asemănător oricărui morfisrm de structuri algebrice, este o funcţie multisortată între suporturile celor două algebre care verifică o condiţie suplimentară. Pentru a scrie această condiţie pentru cazul algebrelor multisortate să plecăm de la conceptul uzual de morfism pentru o structură algebrică bazată pe o operaţie binară. $h : (A, *) \longrightarrow (B, \&)$ este morfism dacă

$$(\forall a \in A)(\forall b \in A)h(a * b) = h(a)\&h(b).$$

Să analizăm egalitatea de mai sus. Pentru un număr de elemente arbitrare din prima algebră egal cu numărul de argumente ale operaţiei evaluăm cei doi membri

- membrul stâng:

1) se aplică operaţia din prima algebră elementelor din prima algebra
2) se aplica morfismul h rezultatului obţinut

- membrul drept:

1) se aplică morfismul h elementelor din prima algebra obţinându-se nişte elemente din a doua algebră
2) se aplică operaţia din a doua algebră acestor elemente

- se cere ca rezultatul evalării celor doi membri să fie egali.

Să facem acelaşi lucru pentru două algebre multisortate $\mathcal{A} = (A_s, A_\sigma)$, $\mathcal{B} = (B_s, B_\sigma)$ şi o funcţie S-sortată $h : A \longrightarrow B$. Condiţia de mai sus trebuie pusă pentru fiecare operaţie cu numele $\sigma : s_1 s_2 \ldots s_n \longrightarrow s$ şi oricare ar fi elementele $a_1 \in A_{s_1}$, $a_2 \in A_{s_2} \ldots a_n \in A_{s_n}$

- membrul stâng:

1) se aplică operaţia din prima algebră elementelor din prima algebra: $A_\sigma(a_1, a_2, \ldots, a_n)$
2) se aplica morfismul h rezultatului obţinut $h_s(A_\sigma(a_1, a_2, \ldots, a_n))$

- membrul drept:

1) se aplică morfismul h elementelor din prima algebra obţinându-se nişte elemente din a doua algebră: $h_{s_1}(a_1), h_{s_2}(a_2), \ldots, h_{s_n}(a_n)$

2) se aplică operaţia din a doua algebră acestor elemente: $B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \ldots, h_{s_n}(a_n))$

- se cere ca rezultatul evalării celor doi membri să fie egali.

$$h_s(A_\sigma(a_1, a_2, \ldots, a_n)) = B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \ldots, h_{s_n}(a_n)).$$

Definiţia 3.1 Funcţia S-sortată $h : A \longrightarrow B$ este un morfism de Σ-algebre multisortate $h : \mathcal{A} \longrightarrow \mathcal{B}$ dacă pentru orice $s_1 s_2 \ldots s_n \in S^*$, pentru orice $s \in S$, pentru orice $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$, pentru orice $a_1 \in A_{s_1}$, $a_2 \in A_{s_2}$,...,$a_n \in A_{s_n}$

$$h_s(A_\sigma(a_1, a_2, \ldots, a_n)) = B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \ldots, h_{s_n}(a_n)).$$

Este util să remarcăm că există câte o condiţie pentru fiecare nume de operaţie. În cazul operaţiilor fără argumente, aşa zisele constante, condiţia de morfism este pentru orice $\sigma \in \Sigma_{\lambda,s}$ egalitatea $h_s(A_\sigma) = B_\sigma$. Cu alte cuvinte morfismele trebuie să păstreze constantele. Pe cazuri particulare observăm că orice morfism de monoizi duce elementul neutru în elementul neutru şi că orice morfism de semiinele duce elementul neutru la adunare, respectiv la înmulţire tot în elementul neutru la adunare respectiv la înmulţire.

Observăm că funcţia identitate 1_A este morfism de Σ-algebre de la $\mathcal{A}$ la $\mathcal{A}$.

Propoziţie 3.2 *Compunerea ca funcţii S-sortate a două morfisme de Σ-algebre este un morfism de Σ-algebre.*

Demonstraţie: Fie $h : \mathcal{A} \longrightarrow \mathcal{B}$ şi $g : \mathcal{B} \longrightarrow \mathcal{C}$ două morfisme de Σ-algebre. Probăm că $h; g : \mathcal{A} \longrightarrow \mathcal{C}$ este morfism de Σ-algebre.

Fie $s_1 s_2 \ldots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ şi $a_1 \in A_{s_1}$, $a_2 \in A_{s_2}$, ..., $a_n \in A_{s_n}$. Observăm că

$$(h; g)_s(A_\sigma(a_1, a_2, \ldots, a_n)) = g_s(h_s(A_\sigma(a_1, a_2, \ldots, a_n))) = g_s(B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \ldots, h_{s_n}(a_n))) =$$

$$= C_\sigma(g_{s_1}(h_{s_1}(a_1)), g_{s_2}(h_{s_2}(a_2)), \ldots, g_{s_n}(h_{s_n}(a_n))) = C_\sigma((h; g)_{s_1}(a_1), (h; g)_{s_2}(a_2), \ldots, (h; g)_{s_n}(a_n)). \square$$

Compunerea morfismelor de Σ-algebre este asociativă.

Morfismul identitate are efect neutru la compunere.

3.1 Izomorfisme de algebre multisortate

Definiţia 3.3 Morfismul de Σ-algebre $h : \mathcal{A} \longrightarrow \mathcal{B}$ se numeşte **izomorfism** dacă există morfismul $g : \mathcal{B} \longrightarrow \mathcal{A}$ cu proprietăţile $h; g = 1_\mathcal{A}$ şi $g; h = 1_\mathcal{B}$.

Dacă există, morfismul g din definiţia de mai sus este unic. Întradevăr dacă $f : \mathcal{B} \longrightarrow \mathcal{A}$ este un alt morfism cu proprietăţile $h; f = 1_\mathcal{A}$ şi $f; h = 1_\mathcal{B}$. Observăm că

$$g = g; 1_\mathcal{A} = g; (h; f) = (g; h); f = 1_\mathcal{B}; f = f.$$

Datorită unicităţii sale, conform uzanţelor morfismul g, denumit şi inversul lui h, este notat în continuare cu h^{-1}.

Observăm că morfismele identitate sunt izomorfisme. In plus $(1_\mathcal{A})^{-1} = 1_\mathcal{A}$.

Propoziţie 3.4 *Un morfism este izomorfism dacă şi numai dacă are toate componentele bijective*

Demonstraţie: Fie $h : \mathcal{A} \longrightarrow \mathcal{B}$ un morfism de Σ-algebre.

Presupunem că h este izomorfism, prin urmare există morfismul $h^{-1} : \mathcal{B} \longrightarrow \mathcal{A}$ cu proprietăţile $h; h^{-1} = 1_\mathcal{A}$ şi $h^{-1}; h = 1_\mathcal{B}$. Rezultă că pentru orice sort $s \in S$ au loc egalităţile $h_s; h_s^{-1} = 1_{A_s}$ şi $h_s^{-1}; h_s = 1_{B_s}$, adică funcţia h_s este inversabilă pentru orice $s \in S$, deci toate componentele h_s ale lui h sunt bijecţii.

Reciproc, presupunem că toate componentele h_s ale lui h sunt bijecţii. Prin urmare pentru orice $s \in S$ există funcţia $h_s^{-1} : B_s \longrightarrow A_s$ cu proprietăţile $h_s; h_s^{-1} = 1_{A_s}$ şi $h_s^{-1}; h_s = 1_{B_s}$. De aici notând $h^{-1} = \{h_s^{-1}\}_{s \in S}$ rezultă că $h; h^{-1} = 1_A$ şi $h^{-1}; h = 1_B$.

Pentru a încheia demonstraţia mai trebuie arătat că funcţia S-sortată $h^{-1} : B \longrightarrow A$ este un morfism de Σ-algebre $h^{-1} : \mathcal{B} \longrightarrow \mathcal{A}$.

Fie $s_1 s_2 \ldots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ şi $b_1 \in B_{s_1}$, $b_2 \in B_{s_2}$, ..., $b_n \in B_{s_n}$. Deoarece h este morfism deducem

$$h_s(A_\sigma(h_{s_1}^{-1}(b_1), h_{s_2}^{-1}(b_2), \ldots, h_{s_n}^{-1}(b_n))) = B_\sigma(h_{s_1}(h_{s_1}^{-1}(b_1)), h_{s_2}(h_{s_2}^{-1}(b_2)), \ldots, h_{s_n}(h_{s_n}^{-1}(b_n))) = B_\sigma(b_1, b_2, \ldots, b_n).$$

Aplicând funcţia h_s^{-1} ambilor membri deducem

$$A_\sigma(h_{s_1}^{-1}(b_1), h_{s_2}^{-1}(b_2), \ldots, h_{s_n}^{-1}(b_n)) = h_s^{-1}(B_\sigma(b_1, b_2, \ldots, b_n))$$

deci $h^{-1} : \mathcal{B} \longrightarrow \mathcal{A}$ este morfism de Σ-algebre.

Propoziţie 3.5 *Compunerea a două izomorfisme este un izomorfism. Î*n *plus*

$$(f; g)^{-1} = g^{-1}; f^{-1}$$

2 ALGEBRE LIBERE - APLICAŢII

Virgil Emil Căzănescu

March 11, 2008

1 Expresii

1.1 Ce este o expresie?

Conceptul de *expresie* aşa cum este el folosit în învăţământul preuniversitar nu are o definiţie şi un înţeles precis. Vom da un exemplu care să ilustreze acest fapt. La întrebarea "este $x * y * z$ o expresie?" răspunsul depinde de contextul în care a fost pusă întrebarea. Dacă operaţia $*$ a fost declarată asociativă, atunci $x * y * z$ este o expresie. În caz contrar ea nu este o expresie deoarece include o ambiguitate putând fi interpretată ca $x * (y * z)$ sau $(x * y) * z$ ambele fiind expresii. În continuare noţiunea de expresie va fi definită în ipoteza că *operaţiile cu care lucrăm nu au nici o proprietate suplimentară.*

Mai menţionăm că cele două expresii de mai sus mai pot fi scrise în scrierea poloneză $*x * yz$ şi $* * xyz$ sau în scrierea poloneză inversă $xyz**$ şi $xy*z*$. Ne interesează *o definiţie a conceptului de expresie care să fie independentă de forma de scriere a acesteia.*

Definiţia 1.1 Σ-algebra $\mathcal{A} = (A_s, A_\sigma)$ se numeşte liber generată de $V \subseteq A$ dacă pentru orice Σ-algebră D şi pentru orice funcţie sortată $f : V \to D$, există un unic morfism de Σ-algebre $f^\# : A \to D$ care extinde f.

Σ-algebra $\mathcal{A}$ se numeşte liberă dacă există $V \subseteq A$ astfel încât $\mathcal{A}$ este liber generată de V.

Definiţia 1.2 Se numeşte *expresie* un element dintr-o algebră liberă.

Bineînţeles că acest concept este încă dependent de signatura cu care lucrăm. În plus noţiunea naivă de expresie ne dă intuiţia necesară pentru înţelegerea conceptului de algebră liberă: algebrele libere nu sunt altceva decât algebre de expresii.

Independenţa de modul de scriere al expresiilor corespunde unicităţii abstracţie de un izomorfism al algebrei libere pentru care este fixată mulţimea generatorilor.

1.2 Evaluarea expresiilor

Un alt concept deosebit de util atât în matematică cât şi în informatică este cel de *evaluare a unei expresii*. Deşi este clar că pentru a evalua o expresie este necesar să dăm valori variabilelor care apar în ea, mai puţin evident este faptul că trebuie precizat şi unde dăm valori acestor variabile. Pentru a ilustra acest fapt menţionăm că expresia $x?(y \top z)$ nu poate fi evaluată numai dând valori variabilelor x, y şi z într-o mulţime dacă mulţimea nu este înzestrată cu două operaţii binare corespunzătoare simbolurilor de operaţii binare $?$ şi $\top$. În concluzie pentru a evalua o expresie este necesar să dăm

1. o algebră în care se fac calculele şi care are aceeaşi signatură cu cea a expresiei
2. valori variabilelor din expresie.

Menţionăm că a da valori variabilelor din mulţimea X în algebra $\mathcal{D}$ este echivalent cu a da o funcţie $v : X \longrightarrow \mathcal{D}$. Pentru orice variabilă x din X valoarea dată lui x este $v(x)$.

Vom nota cu $T_\Sigma(X)$ algebra liber generată de mulţimea X de variabile. Incluziunea $X \subseteq T_\Sigma(X)$ este echivalentă cu faptul intuitiv că orice variabilă este o expresie. Pentru orice algebră $\mathcal{D}$ şi pentru orice funcţie $v : X \longrightarrow \mathcal{D}$ există, conform definiţiei algebrelor libere, un unic morfism $v^\# : T_\Sigma(X) \longrightarrow \mathcal{D}$ a cărui restricţie la X coincide cu v.

Fixând algebra $\mathcal{D}$ vom constata că există o bijecţie naturală între $Alg_\Sigma(T_\Sigma(X), \mathcal{D})$ mulţimea morfismelor de algebre de la $T_\Sigma(X)$ la $\mathcal{D}$ şi $Set_S(X, \mathcal{D})$ mulţimea funcţiilor S-sortate de la X la $\mathcal{D}$. Fie

$$r : Alg_\Sigma(T_\Sigma(X), \mathcal{D}) \longrightarrow Set_S(X, \mathcal{D})$$

funcţia restricţie, adică $r(h) : X \longrightarrow \mathcal{D}$ este restricţia morfismului h la X. Proprietatea de mai sus a algebrei libere spune că

$$(\forall v \in Set_S(X, \mathcal{D}))(\exists! v^{\#} \in Alg_\Sigma(T_\Sigma(X), \mathcal{D}))r(v^{\#}) = v$$

adică r este bijecţie. Existenţa acestei bijecţii ne permite să identificăm elementele celor două mulţimi ne mai făcând distincţie între un morfism $v^{\#}$ de la $T_\Sigma(X)$ la $\mathcal{D}$ şi v, restricţia lui la X.

Dacă mai sus scriam că a da valori variabilelor din mulţimea X în algebra $\mathcal{D}$ este echivalent cu a da o funcţie $v : X \longrightarrow \mathcal{D}$ acum spunem că:

A da valori variabilelor din mulţimea X în algebra $\mathcal{D}$ este echivalent cu a da un morfism $v : T_\Sigma(X) \longrightarrow \mathcal{D}$.

O altă consecinţa a celor de mai sus este:

Pentru a defini un morfism de la algebra liber generată de X la algebra $\mathcal{D}$ este suficient să dăm o funcţie de la X la $\mathcal{D}$.

Definiţia 1.3 Dacă $e \in T_\Sigma(X)$ este o expresie cu variabile din X şi $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ morfismul prin care se dau valori în $\mathcal{D}$ variabilelor, atunci $h(e)$ este rezultatul evaluării expresiei e pentru valorile variabilelor date de funcţia $h : X \longrightarrow \mathcal{D}$. □

Pentru a ne convinge că această definiţie modelează corect realitatea vom relua exemplul de mai sus privind expresia $x?(y \top z)$. Să evaluăm această expresie în mulţimea numerelor naturale unde ? este înmulţirea şi $\top$ este adunarea.

Pentru valorile $x = 2$, $y = 3$ şi $z = 1$ intuitiv obţinem $2 * (3 + 1) = 8$ iar cu definiţia de mai sus unde

$$h : (T_\Sigma(\{x, y, z\}), ?, \top) \longrightarrow (N, *, +)$$

este morfismul definit prin $h(x) = 2$, $h(y) = 3$ şi $h(z) = 1$ obţinem

$$h(x?(y \top z)) = h(x) * h(y \top z) = h(x) * (h(y) + h(z)) = 2 * (3 + 1) = 8.$$

1.3 Semantica instrucţiunii de atribuire

Fie X mulţimea variabilelor utilizate în program. O instrucţiune de atribuire este de forma $x := e$ unde x este o variabilă şi e este o expresie, adică $e \in T_\Sigma(X)$.

Fie $\mathcal{D}$ Σ-algebra datelor cu care se fac calculele. Ne interesează partiţia memoriei în care sunt memorate datele utilizate în timpul execuţiei programului, date depozitate în celule ale memoriei care corespund variabilelor din X. Prin urmare starea memoriei este caracterizată în fiecare moment de o funcţie $s : X \longrightarrow D$. Dacă x este o variabilă s(x) este valoarea din celula de memorie corespunzătoare lui x. Fie S mulţimea stărilor memoriei, adică mulţimea funcţiilor de la mulţimea variabilelor X la mulţimea datelor D.

O funcţie parţială de la A la B este o funcţie definită numai pe o parte a lui A cu valori în B.

Semantica unei instrucţiuni, sau mai general a unui program, este o funcţie parţială F de la mulţimea S a stărilor la ea insăşi. Funcţia F este definită pentru starea s a memoriei dacă şi numai dacă execuţia instrucţiunii începută în starea s a memoriei se termină. Mai mult $F(s)$ este starea memoriei în momentul terminării execuţiei.

Vom defini $S(x := e)$ semantica atribuirii $x := e$. Fie $s : X \longrightarrow D$ starea memoriei la începutul execuţiei atribuirii şi $s^{\#} : T_\Sigma(X) \longrightarrow \mathcal{D}$ unica extindere la un morfism a lui s. Observăm că $s^{\#}(e)$ este rezultatul evaluării expresiei e în starea s a memorie. Prin urmare, prin definiţie

$$S(x := e)(s)(y) = \begin{cases} s^{\#}(e) & \text{dacă } y = x \\ s(y) & \text{dacă } y \neq x. \end{cases}$$

2 Unicitatea abstracţie de un izomorfism a algebrelor libere

Teorema 2.1 *Două algebre liber generate de aceeaşi mulţime sunt izomorfe.*

Demonstraţie: Fie $\mathcal{A} = (A_s, A_\sigma)$ şi $\mathcal{B} = (B_s, B_\sigma)$ două Σ-algebre liber generate de X. Notăm cu $i_A : X \longrightarrow A$ şi $i_B : X \longrightarrow B$ funcţiile incluziune ale lui X în A, respectiv în B. Demonstraţia are patru paşi.

1. Deoarece algebra $\mathcal{A}$ este liber generată de X există un morfism $f : \mathcal{A} \longrightarrow \mathcal{B}$ cu proprietatea $i_A; f = i_B$.

Pasul 2 este asemănător cu primul, doar că se inversează rolul algebrelor $\mathcal{A}$ şi $\mathcal{B}$. La fel vor fi şi paşii 3 şi 4.

2. Deoarece algebra $\mathcal{B}$ este liber generată de X există un morfism $g : \mathcal{B} \longrightarrow \mathcal{A}$ cu proprietatea $i_B; f = i_A$.

3. Deoarece $i_A; (f; g) = (i_A; f); g = i_B; g = i_A = i_A; 1_\mathcal{A}$ şi deoarece $\mathcal{A}$ este algebră liber generată de X deducem $f; g = 1_\mathcal{A}$.

4. Deoarece $i_B; (g; f) = (i_B; g); f = i_A; f = i_B = i_B; 1_\mathcal{B}$ şi deoarece $\mathcal{B}$ este algebră liber generată de X deducem $g; f = 1_\mathcal{B}$.

Deci f şi g sunt izomorfisme inverse unul altuia.

Propoziţie 2.2 *Orice algebră izomorfă cu o algebră liberă este algebră liberă.*

2.1 Algebre iniţiale

Definiţia 2.3 O Σ-algebră $\mathcal{I}$ se numeţe iniţială dacă pentru orice Σ-algebră $\mathcal{A}$ există un unic morfism

$$\alpha_{\mathcal{A}} : \mathcal{I} \longrightarrow \mathcal{A}.$$

Observăm că o algebră este iniţială dacă şi numai dacă este liber generată de mulţimea vidă.
Din teoremele de mai sus rezultă că:

Σ-algebra iniţială este unică abstracţie făcând de un izomorfism.

Acest fapt are aplicaţii importante în informatică.
Un tip de date se numeşte **abstract** dacă este unic determinat abstracţie făcând de un izomorfism. Se vede prin urmare că dând o signatură am dat implicit, prin algebra iniţială corespunzătoare signaturii, un tip abstract de dare.
Vom da un exemplu cunoscut din algebra de liceu. Se ştie că numerele întregi formează un inel iniţial. Vă invităm să reflectaţi asupra următoarei definiţii a idei de număr întreg.

Se numeşte număr întreg un element al inelului iniţial.

3 Tipuri abstracte de date - introducere

Un tip de date se numeşte **abstract** dacă este unic determinat abstracţie făcând de un izomorfism. Abstract înseamnă de fapt că nu ne interesează cum sunt scrise sau memorate datele.
Una dintre metodele prin care se poate defini un tip abstract de date este cel al algebrei iniţiale. Mai clar : este suficient să dăm o signatură şi eventual nişte ecuaţii, condiţionate sau nu, deoarece algebra iniţială, a cărei existenţă este garantată de teoremele care vor fi prezentate mai târziu, este unic determinată abstracţie de un izomorfism, prin urmare este un tip abstract de date.
Tipul numerelor naturale este definit abstract ca fiind semiinelul iniţial.
Tipul numerelor întregi este definit abstract ca fiind inelul iniţial.

Definiţiile de mai sus, deşi corecte sunt ineficiente, deoarece nu face posibilă execuţia de calcule. Prin urmare dorim alte definiţii echivalente dar prin care calculatorul să fie învăţat să facă calcule. Vom exemplifica pentru numere naturale fără intra în prea multe detalii.

3.1 Tipul abstract al numerelor naturale

Considerăm signatura cu un singur sort *nat*, o singură constantă de sort nat şi o singură operaţie unară cu argument şi rezultat de sort nat:

sort nat .
op 0 : $\longrightarrow$ nat .
op s : nat $\longrightarrow$ nat .

Elementele algebrei iniţiale sunt

$$0,\ s(0),\ s(s(0)),\ s(s(s(0))),\ s(s(s(s(0)))),\ ...$$

şi ele reprezintă numerele naturale 0 1 2 3 4 ...

Propoziţie 3.1 *Fie $\mathcal{N} = (N, 0_N, s_N)$ algebra definită prin: N este mulţimea numerelor naturale, 0_N este numărul natural zero şi $s_N(n) = n + 1$ pentru orice număr natural n. Algebra $\mathcal{N}$ este iniţială.*

Demonstraţie: Fie $\mathcal{A} = (A, 0_A, s_A)$ o altă algebră pentru signatura de mai sus. Definim funţia $h : N \longrightarrow A$ prin inducţie

$h(0_N) = O_A$
$h(n + 1) = s_A(h(n))$ pentru orice număr natural n.

Prima egalitate de mai sus şi $h(s_N(n)) = s_A(h(n))$ pentru orice număr natural n dovedesc că $h : \mathcal{N} \longrightarrow \mathcal{A}$ este un morfism.

Probăm unicitatea. Fie $g : \mathcal{N} \longrightarrow \mathcal{A}$ un alt morfism. Arătăm prin inducţie că $g(n) = h(n)$ pentru orice n natural.
$g(0_N) = 0_A = h(0_N)$ şi
$g(n+1) = g(s_N(n)) = s_A(g(n)) = s_A(h(n)) = h(s_N(n)) = h(n+1)$. □

Propoziţia anterioară ne arată cum pot fi definite numerele naturale prim metoda algebrei iniţiale ca tip abstract de date. Ea dovedeşte corectitudinea definiţiei de mai sus.

Deocamdată prin signatura de mai sus calculatorul învaţă numerele naturale dar nu ştie încă să calculeze. Pentru început să-l învăţăm să adune. Dacă introducem în signatură o operaţie binară +

op _+_ : nat nat ⟶ nat

nu realizăm nimic altceva decât să adauge la mulţimea de mai sus a numerelor natural foarte mult gunoi. De exemplu, deoarece calculatorul nu ştie încă să adune, 0 + 0 este un nou element de care nu avem nevoie. Il învăţăm dându-i urmatoarele două reguli de rescriere precedate de o declarare de variabile

var X Y : nat .
eq X + 0 = 0 .
eq X + s(Y) = s(X+Y) .

Trebuie să remarcăm diferenţa esenţială dintre o regulă de rescriere şi o egalitate. O regulă de rescriere se aplică numai de la stânga la dreapta, deoarece simetria este unul dintre marii duşmani ai programării prin rescriere conducând la neterminarea programelor.

Ce părere aveţi despre comutativitate?

Să vedem cum efectuează maşina adunarea 2 + 2, adică:

s(s(0)) + s(s(0)).

Calculatorul nu poate aplica decât a doua regulă pentru X=s(s(0)) şi Y=s(0) ajungând la

s(s(s(0)) + s(0)).

Trebuie din nou aplicată a doua regulă de rescriere pentru X=s(s(0)) şi Y=0 ajungând la

s(s(s(s(0)) + 0)).

Acum se poate aplica numai prima regulă pentru X=s(s(0)) obţinând rezultatul s(s(s(s(0)))), adică 4. Calculatorul se opreşte deoarece nu se mai pot face rescrieri.

Corectitudinea acestei definiţii precum şi a celei care urmează va fi făcută mai târziu.

Calculatorul va şti să şi înmulţească dacă mai introducem o operaţie binară şi două reguli de rescriere

op _*_ : nat nat ⟶ nat .
eq X * 0 = 0 .
eq X * s(Y) = X*Y + X .

3 SUBALGEBRE

Virgil Emil Căzănescu

March 11, 2008

1 Părti stabile, Subalgebre

Ideea de parte stabilă este foarte simplă deoarece este un concept natural. Concret, o parte P a unei algebre este stabilă dacă rezultatul aplicării oricărei operaţii din algebră unor elemente din P este tot în P.

Definiţia 1.1 Fie $\mathcal{A} = (A_s, A_\sigma)$ o Σ-algebră şi $P_s \subseteq A_s$ pentru orice $s \in S$. Partea $P = \{P_s\}_{s\in S}$ a lui A se numeşte stabilă dacă pentru orice $s_1 s_2 \dots s_n \in S^*$, pentru orice $s \in S$, pentru orice $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$, pentru orice $a_1 \in P_{s_1}$, $a_2 \in P_{s_2}$,...,$a_n \in P_{s_n}$ elementul $A_\sigma(a_1, a_2, \dots, a_n)$ este în P_s.

Observăm că orice parte stabilă conţine toate constantele, adică $(\forall s \in S)(\forall \sigma \in \Sigma_{\lambda,s})A_s \in P_s$.

Vom arăta în cele ce urmează că părtile stabile ale oricărei algebre formează o familie Moore. Pentru aceasta se poate demonstra că orice intersecţie de părti stabile este o parte stabilă. Lăsăm acest fapt ca exerciţiu. Vom prefera o cale mai dificilă dar cu rezultat mult mai util în multe cazuri.

Fie $\mathcal{A} = (A_s, A_\sigma)$ o Σ-algebră şi $X \subseteq A$. Definim prin inducţie şirul de părţi ale lui A astfel
$X^0 = X$ şi
$X_s^{n+1} = X_s^n \cup \{A_\sigma(a) : \sigma \in \Sigma_{w,s}, a \in X_w^n\}$ pentru orice $n \in N$ şi orice $s \in S$.

Observăm că şirul $\{X^n\}_{n\in N}$ este crescător.

Definim partea $\overline{X}$ a lui A prin

$$\overline{X} = \bigcup_{n\in N} X^n.$$

Propoziţie 1.2 *$\overline{X}$ este partea stabilă generată de X.*

Demonstraţie: Cu alte cuvinte $\overline{X}$ este cea mai mică parte stabilă a lui $\mathcal{A}$ care include X, adică trebuie să demonstrăm că

1. $X \subseteq \overline{X}$
2. $\overline{X}$ este parte stabilă
3. dacă P este o parte stabilă care include X atunci P include $\overline{X}$

Prima incluziune este aprope evidentă deoarece $X = X^0 \subseteq \overline{X}$.

Probăm că $\overline{X}$ este parte stabilă. Fie $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $a_1 \in \overline{X}_{s_1}$, $a_2 \in \overline{X}_{s_2}$, ..., $a_n \in \overline{X}_{s_n}$. Pentru orice $1 \le i \le n$ din $a_i \in \overline{X}_{s_i}$, adică $a_i \in \bigcup_{n\in N} X_{s_i}^n$ există un număr natural k_i cu proprietatea $a_i \in X_{s_i}^{k_i}$. Fie k cel mai mare dintre numerele $k_1, k_2, \dots, k_n$. Deoarece şirul $\{X^n\}_n$ este crescător deducem că $a_i \in X_{s_i}^k$ pentru orice $1 \le i \le n$. Din definiţia şirului $\{X^n\}_n$ deducem $A_\sigma(a_1, a_2, \dots, a_n) \in X_s^{k+1}$. Deoarece $X^{k+1} \subseteq \overline{X}$ deducem că $A_\sigma(a_1, a_2, \dots, a_n) \in \overline{X}_s$ deci $\overline{X}$ este parte stabilă.

Fie P o parte stabilă a algebrei $\mathcal{A}$ cu proprietatea $X \subseteq P$. Probăm prin inducţie că $X^n \subseteq P$ pentru orice n natural.

Dacă $n = 0$ atunci $X^0 = X \subseteq P$.

Presupunem $X^n \subseteq P$ şi demonstrăm că $X^{n+1} \subseteq P$. Fie $a \in X_s^{n+1}$. Dacă $a \in X_s^n$ din ipoteza de inducţie deducem $a \in P_s$. Altfel există $s_1 s_2 \dots s_k \in S^*$, $s \in S$, şi $a_i \in X_{s_i}^n$ pentru orice $1 \le i \le k$ cu proprietatea $a = A_\sigma(a_1, a_2, \dots, a_k)$. Din ipoteza de inducţie deducem $a_i \in P_{s_i}$ pentru orice $1 \le i \le k$. Deoarece P este parte stabilă deducem că $A_\sigma(a_1, a_2, \dots, a_k) \in P_s$, deci $a \in P_s$.

Deoarece $X^n \subseteq P$ pentru orice n natural rezultă că $\bigcup_{n\in N} X^n \subseteq P$, deci $\overline{X} \subseteq P$. □

Definiţia 1.3 Fie $\mathcal{A}$ o Σ-algebră şi $X \subseteq A$. Dacă $\overline{X} = A$ spunem că X generează $\mathcal{A}$ sau că $\mathcal{A}$ este generată de X.

Operatorul de închidere asociat familiei Moore a părţilor stabile are următoarele proprietăţi:

1. dacă $X \subseteq Y$ sunt părţi ale algebrei, atunci $\overline{X} \subseteq \overline{Y}$,
2. $\overline{\overline{X}} = \overline{X}$ pentru orice parte X a algebrei.

1.1 Inducţie structurală

Această metodă de a face inducţie este folosită pentru a demonstra că elementele unei algebre au o anumită proprietate. Metoda se numeşte structurală deoarece se bazează pe structura algebrei.

Fie $\mathcal{A} = (A_s, A_\sigma)$ o Σ-algebră, X o mulţime de generatori ai algebrei $\mathcal{A}$ şi **P** o proprietate referitoare la elementele algebrei $\mathcal{A}$. Pentru a dovedi că toate elementele algebrei $\mathcal{A}$ au proprietatea **P** este suficient ca să dovedim că

1. orice element din X are proprietatea **P,**
2. pentru orice $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ dacă $a_1 \in A_{s_1}$, $a_2 \in A_{s_2}$, ..., $a_n \in A_{s_n}$ sunt elemente arbitrare cu proprietatea **P**, atunci $A_\sigma(a_1, a_2, \dots, a_n)$ are proprietatea **P.**

Ne vom convinge de corectitudinea inducţiei structurale. Fie B submulţimea lui A formată din toate elementele algebrei $\mathcal{A}$ care au propritatea **P.**
- Proprietatea 1 de mai sus ne asigură că $X \subseteq B$.
- Proprietatea 2 de mai sus ne asigură că B este parte stabilă.

Prin urmare $\overline{X} \subseteq B$. Dar $\overline{X} = A$ deoarece X generează algebra $\mathcal{A}$, prin urmare $B = A$, deci orice element din A are proprietatea **P.**

1.2 Subalgebre

Conceptul de subalgebră este foarte apropiat de cel de parte stabilă. Diferenţa principală constă în faptul că una este o algebră şi alta este o mulţime.

O subalgebră a algebrei $\mathcal{A} = (A_s, A_\sigma)$ este o altă algebră $\mathcal{B} = (B_s, B_\sigma)$ cu proprietăţile $B \subseteq A$ şi $B_\sigma(b_1, b_2, \dots, b_n) = A_\sigma(b_1, b_2, \dots, b_n)$ oricare ar fi $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $b_1 \in B_{s_1}$, $b_2 \in B_{s_2}$, ..., $b_n \in B_{s_n}$.

Observăm că dacă algebra $\mathcal{B}$ este subalgebră a algebrei $\mathcal{A}$, atunci B este o parte stabilă a algebrei $\mathcal{A}$.

Reciproc, dacă B este o parte stabilă a algebrei $\mathcal{A}$ putem defini subalgebra $\mathcal{B}$ de suport B prin $B_\sigma(b_1, b_2, \dots, b_n) = A_\sigma(b_1, b_2, \dots, b_n)$ oricare ar fi $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $b_1 \in B_{s_1}$, $b_2 \in B_{s_2}$, ..., $b_n \in B_{s_n}$.

2 Morfisme şi părţi stabile

O pereche de morfisme cu acelaşi domeniu şi acelaşi codomeniu se mai numeşte şi săgeată dublă.

Definiţia 2.1 Fie $f : \mathcal{A} \longrightarrow \mathcal{B}$ şi $g : \mathcal{A} \longrightarrow \mathcal{B}$ două morfisme. Numin **nucleu de săgeată dublă** al morfismelor f şi g submulţimea lui A notată **Ker(f,g)** şi definită pentru orice sort s prin

$$Ker(f, g)_s = \{a \in A_s : f_s(a) = g_s(a)\}.$$

Propoziţie 2.2 *Nucleul de săgeată dublă este o parte stabilă.*

Demonstraţie: Fie $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $a_1 \in Ker(f, g)_{s_1}$, $a_2 \in Ker(f, g)_{s_2}$, ..., $a_n \in Ker(f, g)_{s_n}$. Pentru orice $1 \leq i \leq n$ deducem că $f_{s_i}(a_i) = g_{s_i}(a_i)$. Prin urmare

$$f_s(A_\sigma(a_1, a_2, \dots, a_n)) = B_\sigma(f_{s_1}(a_1), f_{s_2}(a_2), \dots, f_{s_n}(a_n)) = B_\sigma(g_{s_1}(a_1), g_{s_2}(a_2), \dots, g_{s_n}(a_n)) = g_s(A_\sigma(a_1, a_2, \dots, a_n)),$$

deci $A_\sigma(a_1, a_2, \dots, a_n) \in Ker(f, g)_s$

Corolar 2.3 *Fie $f : \mathcal{A} \longrightarrow \mathcal{B}$ şi $g : \mathcal{A} \longrightarrow \mathcal{B}$ două morfisme şi X o submulţime a lui A. Dacă restricţiile lui f şi g la X coincid, atunci restrcţiile lui f şi g la $\overline{X}$ sunt egale.*

Corolar 2.4 *Fie $f : \mathcal{A} \longrightarrow \mathcal{B}$ şi $g : \mathcal{A} \longrightarrow \mathcal{B}$ două morfisme Dacă restricţiile lui f şi g la o mulţime de generatori ai algebrei $\mathcal{A}$ coincid, atunci $f = g$.*

Propoziţie 2.5 *Fie $h : \mathcal{A} \longrightarrow \mathcal{B}$ un morfism de Σ-algebre.*

1. *Dacă P este o parte stabilă a lui $\mathcal{A}$, atunci $h(P)$ este o parte stabilă a lui $\mathcal{B}$.*
2. *Dacă Q este o parte stabilă a lui $\mathcal{B}$, aunci, $h^{-1}(Q)$ este o parte stabilă a lui $\mathcal{A}$.*
3. *Dacă X este o parte a lui $\mathcal{A}$, atunci $h(\overline{X}) = \overline{h(X)}$.*

Demonstraţie:

1. Prima proprietate spune că imaginea directă a unei părţi stabile printr-un Σ-morfism este o parte stabilă.

Fie $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $b_1 \in h_{s_1}(P_{s_1})$, $b_2 \in h_{s_2}(P_{s_2})$, $\dots$, $b_n \in h_{s_n}(P_{s_n})$. Pentru orice $1 \leq i \leq n$ există $p_i \in P_{s_i}$ astfel încât $b_i = h_{s_i}(p_i)$. Deoarece P este o parte stabilă deducem $A_\sigma(p_1, p_2, \dots, p_n) \in P_s$. Observăm că

$$h_s(A_s(p_1, p_2, \dots, p_n)) = B_s(h_{s_1}(p_1), h_{s_2}(p_2), \dots, h_{s_n}(p_n)) = B_s(b_1, b_2, \dots, b_n).$$

Prin urmare $B_s(b_1, b_2, \dots, b_n) \in h_s(P_s)$, deci $h(P)$ este parte stabilă a algebrei $\mathcal{B}$.

2. A doua proprietate spune că imaginea inversă a unei părţi stabile printr-un Σ-morfism este o parte stabilă.

Fie $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $a_1 \in h_{s_1}^{-1}(Q_{s_1})$, $a_2 \in h_{s_2}^{-1}(Q_{s_2})$, $a_n \in h_{s_n}^{-1}(Q_{s_n})$. Deoarece Q este parte stabilă şi $h_{s_1}(a_1) \in Q_{s_1}$, $h_{s_2}(a_2) \in Q_{s_2}$, $\dots$ $h_{s_n}(a_n) \in Q_{s_n}$ deducem $B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \dots, h_{s_n}(a_n)) \in Q_s$. Deoarece

$$h_s(A_s(a_1, a_2, \dots, a_n)) = B_s(h_{s_1}(a_1), h_{s_2}(a_2), \dots, h_{s_n}(a_n))$$

deducem că $A_s(p_1, p_2, \dots, p_n) \in h_s^{-1}(Q_s)$, deci $h^{-1}(Q)$ este parte stabilă.

3. Din $X \subseteq \overline{X}$ deducem $h(X) \subseteq h(\overline{X})$. Deoarece membrul drept este o parte stabilă fapt ce rezultă din prima proprietate deducem că

$$\overline{h(X)} \subseteq h(\overline{X}).$$

Din $h(X) \subseteq \overline{h(X)}$ deducem $h^{-1}(h(X)) \subseteq h^{-1}(\overline{h(X)})$. Deoarece $X \subseteq h^{-1}(h(X))$ rezultă că $X \subseteq h^{-1}(\overline{h(X)})$. Deoarece membrul drept este conform proprietăţii 2 o parte stabilă deducem $\overline{X} \subseteq h^{-1}(\overline{h(X)})$. Deoarece imaginea directă este crescătare $h(\overline{X}) \subseteq h(h^{-1}(\overline{h(X)}))$. Deoarece $h(h^{-1}(\overline{h(X)})) \subseteq \overline{h(X)}$ deducem

$$h(\overline{X}) \subseteq \overline{h(X)}.$$

Din cele două incluziuni de mai sus rezultă concluzia.

4 ALGEBRE LIBERE

Virgil Emil Căzănescu

March 11, 2008

1 Algebre libere şi algebre Peano

Algebrele libere şi algebrele Peano sunt două concepte echivalente. Pentru a înţelege mai bine acest fapt vom exemplifica un fenomen asemănător din cazul mult mai simplu al monoizilor.

Fie M un monoid şi $B \subseteq M$. Reamintim două definiţii echivalente pentru faptul că monoidul M este liber generat de submulţimea sa B.

Definiţia 1.1 Pentru orice monoid N şi orice funcţie $f : B \longrightarrow N$ există un unic morfism $f^{\#} : M \longrightarrow N$ de monoizi a cărui restricţie la B este f.

Definiţia 1.2 Pentru orice $m \in M$ există şi sunt unice numărul natural n şi elementele $b_1 \in B$, $b_2 \in B$, ..., $b_n \in B$ cu proprietatea $m = b_1 b_2 \dots b_n$.

Să observăm diferenţa esenţială dintre cele două definiţii. Observăm că în definiţia 1.1 nu apar de loc elemente, apărând numai concepte din afara monoidului M. Definiţia 1.2 în schimb lucrează numai cu elemente din interiorul monoidului.

Comparând definiţia dată algebrelor libere cu cele două de mai sus constatăm că definiţia 1.1 este asemănătoare. Sunt atât de asemănătoare încât pot fi generalizate la cel mai înalt nivel de abstractizare, cel al teoriei categoriilor.

Conceptul asemănător celui din definiţia 1.2 este cel de algebră Peano, concept echivalent cu cel de algebră liberă.

Problema esenţială este de a demonstra existenţa algebrelor libere, fapt care nu este simplu. Reamintim că o algebră liberă este de fapt o algebră de expresii. Se poate demonstra că expresiile formează o algebră Peano şi apoi proba că algebrele Peano sunt libere. Deoarece expresiile pot fi scrise în mai multe moduri, fiecare dintre aceste scrieri poate conduce la o demonstraţie. Deoarece textul se adresează unor informaticieni vom prefera reprezentarea expresiilor ca arborii etichetaţi, local ordonaţi, în care frunzele sunt etichete cu variabile(generatori) sau nume de constante. Restul nodurilor sunt etichetate cu nume de operaţii ale căror argumente sunt date de subarborii având rădăcinile drept succesori ale nodului.

1.1 Existenţa algebrelor libere - varianta scurtă

Definiţia 1.3 Se numeşte **mulţime S-sortată de variabile** o mulţime S-sortată $X = \{X_s\}_{s \in S}$ cu componentele disjuncte două câte două.

Condiţia de mai sus rezultă din faptul că o variabilă nu poate fi de două sorturi diferite. Altfel spus fiecare variabilă îşi determină sortul. O definiţie echivalentă ar fi o funţie $f : X \longrightarrow S$. În acest caz $X_s = f^{-1}(s)$ pentru orice $s \in S$.

Varianta scurtă este scrisă pentru cei care au dificultăţi în a înţelege o demonstraţie corectă dar mai dificilă. Cei care preferă varianta scurtă pot sării secţiunea următoare. Ceilalţi sunt invitaţi să sară direct la secţiunea următoare.

Plecăm de la o signatură (S, Σ) şi o mulţime S-sortată de variabile X.

Fie $T_\Sigma(X)$ cea mai mică mulţime S-sortată cu următoarele proprietăţi:

1. $X_s \subseteq T_\Sigma(X)_s$ pentru orice $s \in S$,
2. $\Sigma_{\lambda,s} \subseteq T_\Sigma(X)_s$ pentru orice $s \in S$,
3. Pentru orice $n \geq 1$, pentru orice $s_1 s_2 \dots s_n \in S^*$, pentru orice $s \in S$, pentru orice $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$, dacă $t_i \in T_\Sigma(X)_{s_i}$ pentru orice $1 \leq i \leq n$, atunci $\sigma(t_1, t_2, \dots, t_n) \in T_\Sigma(X)_s$.

Mulţimea S-sortată $T_\Sigma(X)$ devine o Σ-algebră definind operaţiile notate T_σ după cum urmează:

1. $T_\sigma = \sigma$ dacă $\sigma \in \Sigma_{\lambda,s}$
2. $T_\sigma(t_1, t_2, \ldots t_n) = \sigma(t_1, t_2, \ldots, t_n)$ dacă $n \geq 1$, $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ şi $t_i \in T_\Sigma(X)_{s_i}$ pentru orice $1 \leq i \leq n$.

Vom "demonstra" în cele ce urmează că Σ-algebra $(T_\Sigma(X)_s, T_\sigma)$ este liber generată de X.

Fie $\mathcal{A} = (A_s, A_\sigma)$ o Σ-algebră şi $f : X \longrightarrow A$ o funcţie S-sortată.

Definim funcţia S-sortată $f^\# : T_\Sigma(X) \longrightarrow \mathcal{A}$ prin

1. $f^\#_s(x) = f_s(x)$ pentru orice $x \in X_s$ şi $s \in S$,
2. $f^\#_s(\sigma) = A_\sigma$ pentru orice $\sigma \in \Sigma_{\lambda,s}$,
3. $f^\#_s(\sigma(t_1, t_2, \ldots, t_n)) = A_\sigma(f^\#_{s_1}(t_1), f^\#_{s_2}(t_2), \ldots, f^\#_{s_n}(t_n))$ pentru orice $n \geq 1$, pentru orice $s_1 s_2 \ldots s_n \in S^*$, pentru orice $s \in S$, pentru orice $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ şi orice $t_i \in T_\Sigma(X)_{s_i}$ pentru orice $1 \leq i \leq n$.

Din prima parte a definiţiei de mai sus se vede că restricţia funcţiei $f^\#$ la X este chiar funcţia f.
Din celelalte două părţi ale definiţiei rezultă că $f^\#$ este un morfism de Σ-algebre de la $T_\Sigma(X)$ la $\mathcal{A}$.

Unicitatea: Dacă $h : T_\Sigma(X) \longrightarrow \mathcal{A}$ este un morfism a cărui restricţie la X este f, deoarece X este o mulţime de generatori ai lui $T_\Sigma(X)$ şi deoarece h şi $f^\#$ coincid pe X deducem că $h = f^\#$.

Unde-i greşeala?

2 Algebre Peano

Definiţia 2.1 O Σ algebră $\mathcal{A} = (A_s, A_\sigma)$ se numeţe Peano peste $X \subseteq A$ dacă

1. X generează algebra $\mathcal{A}$,
2. pentru orice $\sigma \in \Sigma_{w,s}$ şi orice $a \in A_w$, $A_\sigma(a) \notin X_s$ şi
3. $(\forall \sigma \in \Sigma_{w,s})(\forall a \in A_w)(\forall \sigma' \in \Sigma_{w',s})(\forall a' \in A_{w'})\ A_\sigma(a) = A_{\sigma'}(a') \Rightarrow w = w', \sigma = \sigma'$ şi $a = a'$.

Teorema 2.2 *Orice algebră Peano peste X este liber generată de X.*

Demonstraţie: Fie $\mathcal{A} = (A_s, A_\sigma)$ o algebră Peano peste $X \subseteq A$, $\mathcal{B}$ o altă algebră şi $h : X \longrightarrow B$ o funcţie.

Deoarece algebra $\mathcal{A}$ este generată de X rezultă că

$$A = \bigcup_{n \in N} X^n$$

unde $X^0 = X$ şi pentru orice $n \in N$ şi orice $s \in S$

$$X^{n+1}_s = X^n_s \cup \{A_\sigma(a) : \sigma \in \Sigma_{w,s}, a \in X^n_w\}.$$

Definim prin inducţie după $n \in N$ şirul de funcţii $h^n : X^n \longrightarrow B$ prin $h^0 = h$ şi

$$h^{n+1}_s(a) = \begin{cases} h^n_s(a) & \text{dacă } a \in X^n_s \\ B_\sigma(h^n_w(a')) & \text{dacă } a = A_\sigma(a') \notin X^n_s \text{ unde } \sigma \in \Sigma_{w,s} \text{ şi } a' \in X^n_w \end{cases}$$

Corectitudinea acestei definiţii rezultă din condiţia 3 din definiţia algebrei Peano.

Observăm că şirul funcţiilor $h^n : X^n \longrightarrow B$ este crescător, adică restricţia lui h^{n+1} la X^n este chiar h^n. Mai mult pentru orice $m \geq n$ restricţia lui h^m la X^n este chiar h^n.

Definim funcţia $g : A \longrightarrow B$ pentru orice $s \in S$şi orice $a \in A_s$ prin

$$g_s(a) = h^n_s(a) \text{ dacă } n \text{ este cel mai mic număr natural cu proprietatea } a \in X^n_s.$$

Observăm că $g_s(a) = h^m_s(a)$ pentru orice număr natural m cu proprietatea $a \in X^m_s$.

Probăm că $g : \mathcal{A} \longrightarrow \mathcal{B}$ este morfism de algebre. Fie $\sigma \in \Sigma_{w,s}$ şi $a \in A_w$.
Deoarece $A_\sigma(a) \notin X^0_s$ există n cel mai mic număr natural cu proprietatea $A_\sigma(a) \in X^{n+1}_s - X^n_s$. Rezultă că

$$g_s(A_\sigma(a)) = h^{n+1}_s(A_\sigma(a)).$$

Probăm că $a \in X_w^n$. Deoarece $A_\sigma(a) \in X_s^{n+1} - X_s^n$ există $\sigma' \in \Sigma_{w',s}$ şi $a' \in X_{w'}^n$ astfel încât $A_\sigma(a) = A_{\sigma'}(a')$. Rezultă că $w = w'$, $\sigma = \sigma'$ şi $a = a'$, deci $a \in X_w^n$. Prin urmare

$$g_w(a) = h_w^n(a) \quad \text{şi} \quad h_s^{n+1}(A_\sigma(a)) = B_\sigma(h_w^n(a))$$

deci

$$g_s(A_\sigma(a)) = B_\sigma(h_w^n(a)) = B_\sigma(g_w(a)).$$

Restricţia lui g la $X = X^0$ este $h^0 = h$.

Unicitatea lui g este consecinţa faptului că X generează $\mathcal{A}$. □

3 Algebra arborilor de derivare

O gramatică independentă de context posedă două mulţimi disjuncte, una a neterminalelor N şi una a terminalelor T precum şi multimea $P \subseteq N \times (N \cup T)^*$ a producţiilor. Chiar dacă conceptul de gramatică independentă de context verifică şi alte condiţii, noi ne restrângem doar la cele de mai sus deoarece acestea sunt utile în cele ce urmează.

Fiecărei gramaticii independente de context G i se poate ataşa o signatură:

- neterminalele devin sorturi,
- producţiile devin nume de operaţii,
- dacă $(n, t_0 n_1 t_1 \dots n_k t_k) \in P$ unde literele n sunt neterminale şi literele t sunt cuvinte cu litere terminale, atunci

$$(n, t_0 n_1 t_1 \dots n_k t_k) : n_1 n_2 \dots n_k \longrightarrow n$$

adică operaţia corespunzătoare lui $(n, t_0 n_1 t_1 \dots n_k t_k)$ are ca rezultat un element de sortul indicat de neterminalul din membrul stâng al producţiei, un număr de argumente egal cu numărul de neterminale din membrul drept al producţiei şi sorturile argumentelor sunt chiar neterminalele din membrul drept al producţiei.

O algebră a cărei signatură este cea ataşată gramaticii independente de context G se numeşte G-algebră.

Un arbore de derivare într-o gramatică independentă de context are următoarele proprietăţi:

1. are noduri etichetate cu terminale sau neterminale, rădăcina fiind etichetată cu un neterminal;
2. este local ordonat: succesorii fiecărui nod sunt într-o ordine totală;
3. pentru orice nod, dacă este etichetat cu un terminal atunci nu are succesor, iar dacă este etichetat cu un neterminal n, atunci acesta şi cuvântul format de etichetele succesorilor formează o producţie $(n, s_1 s_2 \dots s_{k-1} s_k) \in P$.

Arborii de derivare ai unei gramatici independente de context G pot fi organizaţi ca o G-algebră $\mathcal{A}$ după cum urmează:

- pentru orice neterminal n mulţimea A_n este formată din totalitatea arborilor de derivare care au rădăcina etichetată cu n.
- pentru producţia $p = (n, t_0 n_1 t_1 \dots n_k t_k)$ şi arborii $a_i \in A_{n_i}$ arborele $A_p(a_1, a_2, \dots, a_k)$ este format astfel: rădăcina este etichetată cu n, succesorii rădăcinii în număr egal cu numărul literelor din $t_0 n_1 t_1 \dots n_k t_k$ sunt chiar literele acestui cuvânt şi subarborele fiecărui nod etichetat cu n_i este chiar a_i.

Teorema 3.1 *Algebra arborilor de derivare ai unei gramatici independente de context este algebră Peano peste mulţimea vidă.*

Demonstraţie: Pentru a proba că $\mathcal{A}$ este generată de mulţimea vidă este suficient să demonstrăm că A este singura parte stabilă a lui $\mathcal{A}$. Fie P o parte stabilă a lui $\mathcal{A}$. Vom proba prin inducţie dupa adâncimea arborilor că orice arbore este în P. Reamintim că adâncimea unui arbore este lungimea celei mai lungi ramuri din arbore. Fie m un număr natural. Presupunem prin ipoteza de inducţie că orice arbore cu adâncimea strict mai mică decât m este în P. Probăm că orice arbore de adâncime m este în P. Fie a un arbore de adâncime m. Analizând rădăcina şi primul nivel al arborelui a deducem existenţa unei producţii $p = (n, t_0 n_1 t_1 \dots n_k t_k)$ cu proprietatea

$$a = A_p(a_1, a_2, \dots, a_k).$$

unde a_i sunt subarborii lui a care au vârfurile în succesorii rădăcinii lui a care sunt etichetaţi cu neterminale.

Observăm că arborii $a_1, a_2, \dots, a_k$ au adâncimea cu cel puţin o unitate mai mică decât adâncimea lui a, adică strict mai mică decât m. Prin ipoteza de inducţie rezultă că $a_i \in P_{s_i}$ pentru orice $1 \leq i \leq k$. Deoarece P este o parte stabilă deducem că $A_p(a_1, a_2, \dots, a_k) \in P_s$, deci $a \in P_s$.

A doua condiţie din definiţia algebrelor Peano

$$A_P(a_1, a_s, \ldots, a_k) \notin \emptyset$$

este evident adevărată.

Trecem la ultima condiţie. Fie $p = (n, t_0 n_1 t_1 \ldots n_k t_k)$ şi $q = (n, t'_0 s_1 t'_1 \ldots s_{k'} t'_{k'})$ două nume de operaţii(producţii) unde literele n şi s sunt neterminale şi literele t sunt cuvinte formate din terminale. Fie $a_i \in A_{n_i}$ pentru $1 \leq i \leq k$ şi $b_i \in A_{s_i}$ pentru $1 \leq i \leq k'$ astfel încât

$$A_p(a_1, a_2, \ldots, a_k) = A_q(b_1, b_2, \ldots, b_{k'})$$

Egalând cuvintele formate cu etichetele succesorilor rădăcinilor din cei doi arbori egali deducem

$$t_0 n_1 t_1 \ldots n_k t_k = t'_0 s_1 t'_1 \ldots s_{k'} t'_{k'}.$$

Reamintim că prin definiţia gramaticilor un element nu poate fi în acelaşi timp şi terminal şi neterminal. Deoarece numărul neterminalelor din cele două cuvintre trebuie să fie egal deducem egalitatea $k = k'$. Deoarece primele neterminale din cele două cuvinte trebuie să fie pe aceeaşi poziţie deducem că $t_0 = t'_0$ şi $n_1 = s_1$. Rezultă că

$$t_1 \ldots n_k t_k = t'_1 \ldots s_k t'_k.$$

Continuăm raţionamentul ca mai sus şi deducem $t_i = t'_i$ pentru $0 \leq i \leq k$ şi $n_i = s_i$ pentru $1 \leq i \leq k$. De aici deducem că $p = q$.

În final egalând subarborii cu rădăcinile aflate pe aceeaşi poziţie ale primului nivel rezultă că $a_i = b_i$ pentru orice $1 \leq i \leq k$. □

Scriu rândurile care urmează deoarece de mai multe ori câţiva dintre studenţii au contestat lipsa primului pas al inducţiei în prima parte a demonstraţiei de mai sus. El nu lipseşte ci este pur şi simplu inclus în demonstraţia de mai sus. Primul pas este cazul $m = 0$. Este evident că prin ipoteza de inducţie nu se presupune nimic deoarece nu există arbori de adâncime strict negativă. Arborele a de adâncime 0 nu are decât rădăcină etichetată să spunem cu n, prin urmare $p = (n, \lambda)$, deci $a = A_p$. În concluzie $A_p \in P_n$ deoarece P este parte stabilă, deci $a \in P_n$.

4 Existenţa algebrelor libere

Urmărim să arătăm că pentru orice signatură Σ şi pentru mulţime S-sortată de variabile X există o Σ-algebră liber generată de X pe care în cele ce urmează o vom nota cu $T_\Sigma(X)$. Construcţia algebrei libere care urmează este bazată pe scrierea poloneză a expresiilor, adică un şir de semne de operaţii şi variabile în care semnul de operaţie este plasat în faţa argumentelor sale care trebuie sa-l urmeze. Producţiile de forma $(s, \sigma s_1 s_2 \ldots s_n)$ unde $\sigma \in \Sigma_{s_1 s_2 \ldots s_n}$ spun că dacă e_i este expresie de sort s_i pentru orice $1 \leq i \leq n$, atunci $\sigma e_1 e_2 \ldots e_n$ este expresie de sort s. Producţiile de forma (s, x) unde $s \in S$ şi $x \in X_s$ spun că orice variabilă x de sort s este expresie de sort s.

Fie (S, Σ) o signatură şi X o mulţime S-sortată cu componentele disjuncte două câte două. Fără a micşora generalitatea vom presupune că Σ şi X sunt disjuncte.

Considerăm gramatica independentă de context definită prin
1. Mulţimea neterminalelor este S,
2. Mulţimea terminalelor este $\Sigma \cup X$ şi
3. Mulţimea producţiilor este $\{(s, \sigma w) | \sigma \in \Sigma_{w,s}\} \cup \{(s, x) | s \in S, x \in X_s\}$.

Notăm cu $\mathcal{A} = (A_s, A_{(s,\sigma w)}, A_{(s,x)})$ algebra arborilor ei de derivare. Ea este algebră Peano peste mulţimea vidă.

Notăm cu $\mathcal{B} = (A_s, B_\sigma)$ algebra de signatură Σ definită prin $B_\sigma = A_{(s,\sigma w)}$ pentru orice $\sigma \in \Sigma_{w,s}$ şi cu $i : X \longrightarrow A$ funcţia S-sortată definită pentru orice $s \in S$ şi $x \in X_s$ prin $i_s(x) = A_{(s,x)}$. Observăm că funcţia i are toate componentele injective.

Teorema 4.1 *Σ-algebra $\mathcal{B}$ este Peano peste $i(X)$.*

Demonstraţie:

1. Fie $C \subseteq A$ o parte stabilă a algebrei $\mathcal{B}$ care include $i(X)$. Observăm că C este o parte stabilă a algebrei $\mathcal{A}$. Deaorece $\mathcal{A}$ este Peano peste mulţimea vidă rezultă că $C = A$.

În concluzie algebra $\mathcal{B}$ este generată de $i(X)$.

2. Fie $\sigma \in \Sigma_{w,s}$ şi $a \in A_w$. Deoarece $A_{(s,\sigma w)}(a) \neq A_{(s',x)}$ pentru orice $s' \in S$ şi $x \in X_{s'}$ rezultă că $B_\sigma(a) \notin i(X)$.
3. Fie $\sigma \in \Sigma_{w,s}$, $a \in A_w$, $\sigma' \in \Sigma_{w',s}$ $a' \in A_{w'}$ cu $B_\sigma(a) = B_{\sigma'}(a')$. Din

$$A_{(s,\sigma w)}(a) = A_{(s',\sigma w')}(a')$$

deoarece $\mathcal{A}$ este algebră Peano deducem că $(s, \sigma w) = (s, \sigma' w')$ şi $a = a'$. Mai observăm că $\sigma = \sigma'$ şi $w = w'$. □

Putem spune că ne-am atins scopul deoarece Σ-algebra $\mathcal{B}$ este liber generată de mulţimea $i(X)$ care, deoarece funcţia i este injectivă, este în bijecţie cu X.

Propoziţie 4.2 *Orice algebră liberă este Peano*

Demonstraţie: Fie $\mathcal{L}$ o Σ-algebră liber generată de mulţimea X. Considerăm o Σ-algebră $\mathcal{P}$ Peano peste X. Deoarece şi $\mathcal{P}$ este liber generată de X rezultă existenţa unui izomorfism $i : \mathcal{P} \longrightarrow \mathcal{L}$ cu proprietatea $i(x) = x$ pentru orice $x \in X$. Deci $\mathcal{L}$ este algebră Peano peste X.

5 SEMANTICA ALGEBREI INIŢIALE

Virgil Emil Căzănescu

March 22, 2008

Metoda semanticii algebrei initiale este o simplificare a metodei mai clasice a semanticii denotaţionale.

Metoda semanticii algebrei inţiale se aplică pentru limbaje definite printr-o gramatică independentă de context $G = (N, T, P, a)$, unde N este mulţimea neterminalelor, T mţimea terminalelor, P mulţimea producţiilor şi a axioma gramaticii. Ea spune că **pentru a defini semantica limbajului gramaticii G este suficient să dăm o G-algebră $\mathcal{S} = (S_n, S_p)$** unde n este un neterminal şi p o producţie.

Pentru a înţelege afirmaţia de mai sus trebuie să intrăm puţin în amănunte. Fie $\mathcal{A}$ algebra arborilor de derivare. Deoarece $\mathcal{A}$ este algebră inţială există un unic morfism de G-algebre

$$M : \mathcal{A} \longrightarrow \mathcal{S}.$$

Dat un cuvânt c din limbajul gramaticii G există un arbore de derivare arb cu rădăcina etichetată cu a a cărui frontieră este c. Semantica cuvântului c este prin definiţie $M_a(arb)$.

Menţionăm că metoda este bine definită numai pentru gramaticile neambigue, fapt care asigură unicitatea arborelui arb. Acest aspect este mai degrabă legat de analiza sintactică.

Trecem la exemple care vor clarifica şi mai mult ideile de mai sus.

1 Semantica unui şir de cifre ca număr natural

Vom considera o gramatică G care generează şirurile finite de cifre zecimale, considerate ca terminale. Gramatica are două neterminale <cifra> şi <nat> ultima fiind şi axiomă a gramaticii. Descriem în continuare producţiile gramaticii cărora le dăm un nume

ci	<cifra>	$\longrightarrow$	i	pentru orice cifră zecimală i
n1	<nat>	$\longrightarrow$	<cifra>	
n2	<nat>	$\longrightarrow$	<nat> <cifra>	

Vom defini algebra semantică explicând de ce o definim astfel. Deoarece gramatica are două neterminale, signatura asociată are două sorturi, prin urmare algebra semantică trebuie să aibă ca suport două mulţimi. Deoarece semantica unei cifre, care nu este nimic altceva decât un semn, este numărul reprezentat de cifră. Prin urmare suportul corespunzător neterminalului <cifra> trebuie să conţină măcar numerele de la zero la nouă. Deci prin definiţie

$$S_{<cifra>} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

Deoarece semantica unui şir finit de cifre este numărul natural pe care acest şir îl reprezintă, suportul corespunzător neterminalului <nat> trebuie să conţină măcar mulţimea numerelor naturale. Deci prin definiţie

$$S_{<nat>} = \text{este mulţimea numerelor naturale.}$$

Să trecem la definirea operaţiilor algebrei semantice.

Deoarece producţia ci corespunzătoare cifrei i nu are nici un neterminal în membrul drept, operaţia corespunzătoare ei este o operaţie fără argumente, adică un element din $S_{<cifra>}$

$$S_{ci} \text{ este numărul natural reprezentat de cifra } i.$$

Producţia n1 spune că orice cifră este un şir dr cifre. Deoarece valoarea numărului reprezentat de un şir de lungime unu este egală cu valoarea numărului reprezentat de singura cifră din şir şi semantica lor trebuie să fie aceeaşi. Deci prin definiţie

$$S_{n1} \text{ este funcţia incluziune de la } S_{<cifra>} \text{ la } S_{<nat>}.$$

Producţia n2 ne spune că prin adăugarea la dreapta unui şir de cifre, să spunem s1, a unei cifre, să spunem c, se obţine un nou şir de cifre s2 = s1c. Operaţia corespunzătoare producţiei n2 trebuie să fie o funcţie

$$S_{n2} : S_{<nat>} \times S_{<cifra>} \longrightarrow S_{<nat>}$$

definită pentru numele n şi m prin

$$S_{n2}(n, m) = 10n + m$$

deoarece valoarea ca număr a şirului s2 este egală cu de zece ori valoarea ca număr a şirului s1 plus valoarea ca număr a cifrei.

Cu aceasta semantica este complet definită. Vom exemplifica pentru şirul 023. Arborele de derivare pentru acest şir este

$$A_{n2}(A_{n2}(A_{n1}(A_{c0}), A_{c2}), A_{c3}).$$

Să-i aplicăm morfismul semantic acestui arbore.

$M_{<nat>}(A_{n2}(A_{n2}(A_{n1}(A_{c0}), A_{c2}), A_{c3})) = S_{n2}(S_{n2}(S_{n1}(S_{c0}), S_{c2}), S_{c3}) = S_{n2}(S_{n2}(S_{n1}(0), 2), 3) = S_{n2}(S_{n2}(0, 2), 3) =$

$S_{n2}(10 \times 0 + 2, 3) = S_{n2}(2, 3) = 10 \times 2 + 3 = 23$

Ce sa întâmplă dacă în membrul drept al producţiei n2 scriem $< cifra >< nat >$?

2 Un calculator de buzunar

Vom da un alt exemplu, cel al unui minicalculator pe faţa căruia se află un mic ecran pe care se poate afişa un singur număr şi mai multe butoane:

ON pentru pornirea calculatorului
OFF pentru oprirea calculatorului
0 1 2 3 4 5 6 7 8 9 pentru cifrele zecimale
+ × pentru adunare şi înmulţire
M pentru unica celulă de memorie a calculatorului
( ) pentru parantezele necesare în scrierea expresiilor
IF , pentru o anumită instrucţiune
E pentru comanda de evaluarea a unei expresii

Vom considera o gramatică G care generează limbajul programelor care pot fi executate de minicalculator. Ea extinde gramatica din exemplul precedent.

Gramatica are cinci neterminale <cifra>, <nat>, <exp> pentru expresii, <inst> pentru anumite porţiuni de programe şi <program> care este şi axiomă a gramaticii. Descriem în continuare producţiile gramaticii cărora le dăm un nume

ci	<cifra>	$\longrightarrow$	i	pentru orice cifră zecimală i
n1	<nat>	$\longrightarrow$	<cifra>	
n2	<nat>	$\longrightarrow$	<nat> <cifra>	
r1	<exp>	$\longrightarrow$	<nat>	
r2	<exp>	$\longrightarrow$	M	
r3	<exp>	$\longrightarrow$	<exp> + <exp>	
r4	<exp>	$\longrightarrow$	<exp> × <exp>	
r5	<exp>	$\longrightarrow$	IF <exp>,<exp>,<exp>	
r6	<exp>	$\longrightarrow$	(<exp>)	
r7	<inst>	$\longrightarrow$	<exp> E OFF	
r8	<inst>	$\longrightarrow$	<exp> E <inst>	
r9	<program>	$\longrightarrow$	ON <inst>	

Pentru a da semantica este necesar să explicăm cum functionează calculatorul. Se porneşte calculatorul apăsând butonul ON, moment în care se iniţializează unica celulă de memorie cu zero. Se introduce o expresie care la apăsarea butonului E se evaluează. Rezultatul evaluarii este afişat pe ecran şi introdus în unica celulă de memorie în locul vechii valori. Se introduce altă expresie, se apasă butonul E şi aşa mai departe. La final se apasă butonul OFF pentru închiderea calculatorului. Mai remarcăm că se calculează numai cu numere naturale.

Vom defini algebra semantică. Deoarece signatura asociată are cinci sorturi, algebra semantică trebuie să aibă ca suport cinci mulţimi, primele două fiind cele de mai sus $S_{<cifra>}$ şi $S_{<nat>}$.

Expresiile sunt făcute pentru a fi evaluate, prin urmare prima idee ar fi ca semantica unei expresii să fie un număr. Dar oare ce număr se obţine prin evaluarea expresiei M+3. Trebuie să menţionăm că fiecare apariţie a lui M este înlocuită în timpul evaluării cu valoarea care se află în unica celulă de memorie. Prin urmare rezultatul evaluării lui M+3 depinde de conţinutul celulei de memorie. Prin urmare semantica expresiei M+3 este funcţia $f : N \longrightarrow N$ defină prin $f(x) = x + 3$ care asociază numărului x care se află în celula de memorie rezultatul $x + 3$ al evaluării acesteia. Prin urmare prin definiţie $S_{<exp>}$ este mulţimea funcţiilor de la mulţimea numerelor naturale la ea însăşi.

O instrucţiune comandă evaluarea unui şir finit de expresii. Semantica instrucţiunii este, prin definiţie, şirul numerelor care apar pe ecran în timpul execuţiei instrucţiunii, adică un şir finit şi nevid de numere naturale. Mulţimea acestor şiruri o notăm cu N^+. Rezultatul evaluării depinde evident de valoarea din memorie din momentul începerii execuţiei instrucţiunii, prin urmare

$$S_{<inst>} = \{f : N \longrightarrow N^+ \ / \ f \ este funcţie.\}$$

Un program diferă de o instrucţiune prin faptul că se introduce zero în celula de memorie înainte de execuţia instrucţiunii pe care o conţine, prin urmare şirul de numere care apar pe ecran în timpul execuţiei programului nu mai depinde de conţinutul memoriei, fiind un element din N^+, deci $S_{<program>} = N^+$.

Definiţiile pentru operaţii, fără a mai repeta pe cele de mai sus sunt:

$S_{r1}(n)(m) = n$
$S_{r2}(m) = m$
$S_{r3}(e1, e2)(m) = e1(m) + e2(m)$
$S_{r4}(e1, e2)(m) = e1(m) \times e2(m)$
$S_{r5}(e1, e2, e3)(m) =$ dacă $e1(m) = 0$ atunci $e2(m)$ altfel $e3(m)$
$S_{r6}(e) = e$
$S_{r7}(e)(m) = e(m)$
$S_{r8}(e, ı)(m) = e(m)ı(e(m))$
$S_{r9}(ı) = ı(0)$

unde am folosit următoarele notaţii

m este un număr natural reprezentând conţinutul celulei de memorie
n este un număr natural
e este o funcţie de la N la N reprezentând semantica unei expresii
ı este o funcţie de la N la N^+ reprezentând semantica unei instrucţiuni.

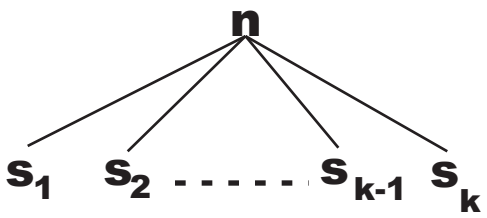
APLICAŢII ale metodei ALGEBREI INIŢIALE

Virgil Emil Căzănescu

February 14, 2010

1 Arbori de derivare

Un arbore de derivare într-o gramatică independentă de context are următoarele proprietăţi:
1. are noduri etichetate cu terminale sau neterminale;
2. este local ordonat: succesorii fiecărui nod sunt într-o ordine totală;
3. pentru orice nod, dacă este etichetat cu un terminal atunci nu are succesor, iar dacă este etichetat cu un neterminal n, atunci acesta şi cuvântul format de etichetele succesorilor formează o producţie $(n, s_1 s_2 \dots s_{k-1} s_k) \in P$.



1.1 O gramatică pentru expresii

Definim o gramatică independentă de context pentru expresiile construite cu variabilele x, y şi z, cu operaţiile binare de adunare şi înmulţire şi cu paranteze. Gramatica trebuie constrită respectand urmatoarele condiţii privind expresiile:

1. înmulţirile se fac înaintea adunărilor
2. adunările se fac de la stânga la dreapta
3. înmulţirile de la dreapta la stânga.

Mulţimea terminalelor este $\{x, y, z, (,), +, *\}$. Fie $N = \{V, F, T, E, P\}$ mulţimea neterminalelor. Semnificaţia lor este următoarea:

V reprezintă variabilele
F de la factor, reprezintă cele mai "mici" elemente folosite în construcţia expresiilor
T de la termen, reprezintă un produs de factori
E de la expresie, reprezintă o sumă de termeni
P de la program.

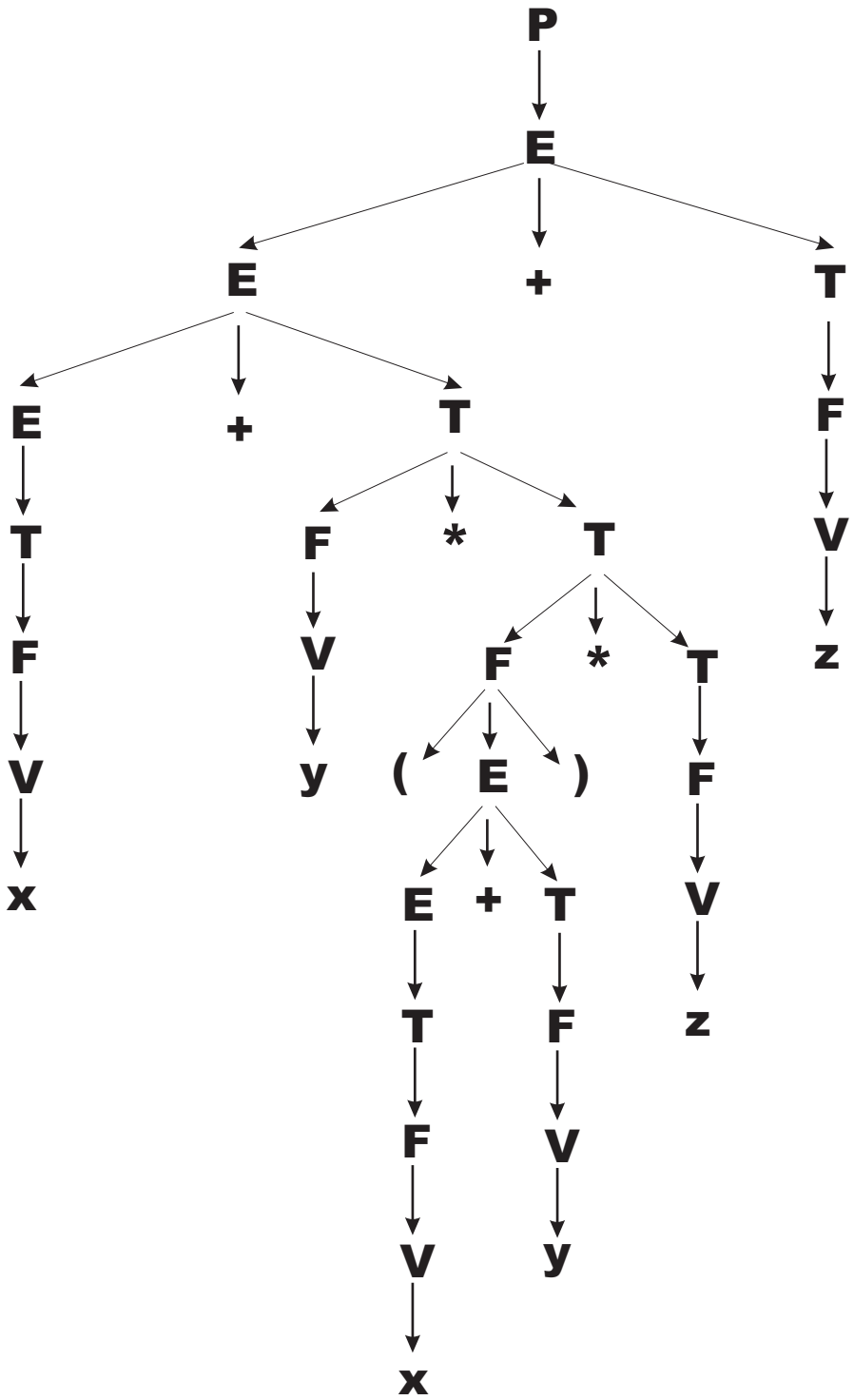
Continuăm cu mulţimea **regulilor(producţiilor)** gramaticii:

0. $P \to E$ **1.** $V \to x$ **2.** $V \to y$ **3.** $V \to z$ **4.** $F \to V$
5. $F \to (E)$ **6.** $T \to F$ **7.** $T \to F * T$ **8.** $E \to T$ **9.** $E \to E + T$

Observaţi producţiile 7 şi 9 pentru a vedea cum se precizează ordinea de execuţie a operaţiilor de acelaşi fel. Neterminalul din membrul stâng se află în dreapta, respectiv în stânga semnului operaţiei.

1.2 Un exemplu

Să se construiască arborele de derivare pentru expresia $x + y * (x + y) * z + z$.



Pentru a reface expresia, arborele se parcurge în inordine.

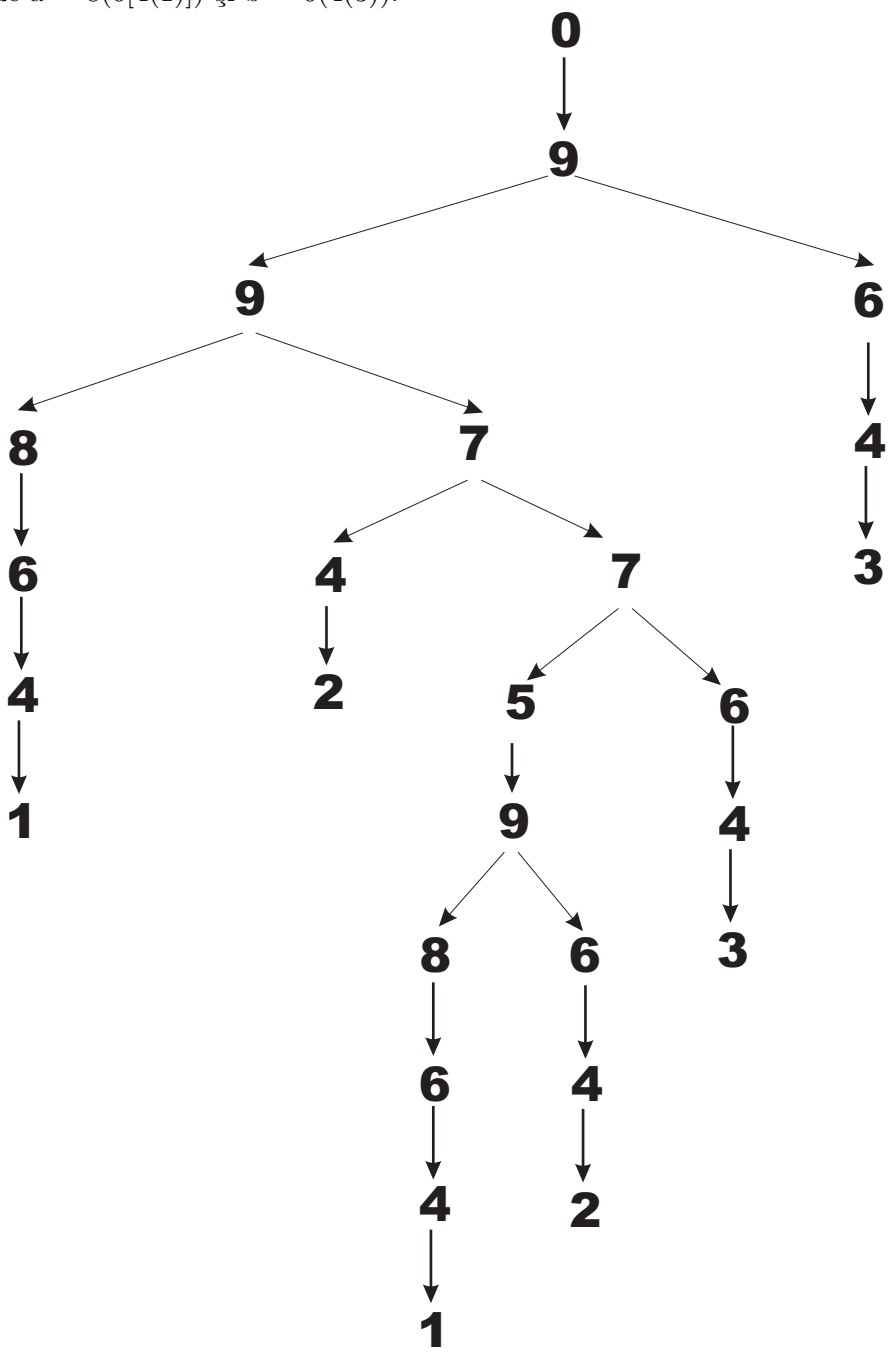
Dacă veţi incerca să construiţi alt arbore de derivare pentru aceeaşi expresie veţi constata ca acest lucru nu este posibil. Prin urmare modul de construcţie al expresiei va impune o anumită ordine în evaluarea acesteia.

Folosind numerele care denumesc producţiile ca nume ale operaţiilor din algebra arborilor de derivare

arborele de mai sus este

$$A = 0(9[9\{a,\ 7(4[2],\ 7[\ 5\{9(a,6[4(2)])\},b\ ])\ \},b])$$

unde a = 8(6[4(1)]) şi b = 6(4(3)).



2 Scrierea poloneză inversă

Semantica pe care o dăm unui limbaj depinde de scopul pe care-l urmărim. Pentru a ilustra această idee vom da două semantici pentru expresiile definite mai sus.

Prima semantică va da scrierea poloneză inversă pentru expresii. A doua semantică va fi dată în secţiunea următoare.

Algebra semantică va avea toate suporturile egale cu semigrupul liber generat de $\{x, y, z, +, *\}$.

Operaţiile sunt următoarele:

$0 = 4 = 5 = 6 = 8$ sunt aplicaţia identitate,

1 = x, 2 = y şi 3 = z,

$7(\alpha, \beta) = \alpha\beta+$ şi $9(\alpha, \beta) = \alpha\beta*$.

Aplicând morfismul semantic arborelui A obţinem xyxy+z**+z+.

3 Compilare

Un compilator care va produce un program pentru evaluarea expresiilor si tipărirea rezultatului evalării va fi modelat printr-un morfism.

Vom pune în evidenţă elementele care apar în programele obţinute prin compilare. **R** este un registru din memorie. **P** este un pointer către primul loc liber al stivei utilizate în timpul evaluării.

Presupunem că limbajul de asamblare include următoarele instrucţiuni:

inc P	operator de incrementare a poziţiei primului loc liber din stivă
dec P	operator de decrementare a poziţiei primului loc liber din stivă
Ad R	adună valoarea din vârful stivei prin conţinutul lui R şi depune rezultatul în R
Mu R	inmulţeşte valoarea din vârful stivei prin conţinutul lui R şi depune rezultatul în R
ld R	valoarea din vârful stivei este depusă în R
st R	valoarea din R este depusă în vârful stivei
print	tipăreşte valoarea din vârful stivei

Semantica unei expresii va fi programul care evaluează expresia si imprima rezultatul evaluării. Pentru definirea acesteia se va folosi metoda semanticii algebrei iniţiale.

Toate cele 5 suporturi ale algebrei semantice, corespunzătoare neterminalelor, sunt identice şi coincid cu mulţimea bucăţilor de program(şiruri de instrucţiuni separate prin ;). Astfel de bucăţi de programe vorfi notete cu litere greceşti.

Cele 10 operaţii ale algebrei semantice, corespunzătoare producţiilor sunt:

1_S	=	st x ; inc P
2_S	=	st y ; inc P
3_S	=	st z ; inc P
$4_S(\alpha)$	=	α
$5_S(\alpha)$	=	α
$6_S(\alpha)$	=	α
$7_S(\alpha,\beta)$	=	α ; β ; dec P ; ld R ; dec P ; Mu R ; st R ; inc P
$8_S(\alpha)$	=	α
$9_S(\alpha,\beta)$	=	α ; β ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P
$0_S(\alpha)$	=	α ; dec P ; print

Notăm cu $\mathcal{S}$ algebra semantică de mai sus, cu $\mathcal{A}$ algebra arborilor de derivare şi cu $C : \mathcal{A} \longrightarrow \mathcal{S}$ unicul morfism de algebre existent. Morfismul C este modelarea algebrică a compilatorului.

Conform metodei semanticii algebrei iniţiale rezultă că $C(\mathrm{A})$ este programul care evaluează şi tipăreşte rezultatul pentru expresia $x + y * (x + y) * z + z$.

3.1 Program

Prezentăm calculele care dovedesc afirmaţia de mai sus.

$$C(\mathrm{a}) = 8_S(6_S[4_S(1_S)]) = 1_S \quad \text{şi} \quad C(\mathrm{b}) = 6_S(4_S(3_S)) = 3_S$$

Notând c = 7[5{9(a,6[4(2)])},b]

$C(\mathrm{c}) = 7_S[\ 5_S\{9_S(1_S,6_S[4_S(2_S)])\},3_S\ ] = 7_S[\ 9_S(1_S,2_S),3_S\ ] =$
$9_S(1_S,2_S),3_S$; dec P ; ld R ; dec P ; Mu R ; st R ; inc P =
1_S ; 2_S ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; 3_S ; dec P ; ld R ; dec P ; Mu R ; st R ; inc P =
st x ; inc P; st y ; inc P ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; st z ; inc P; dec P ; ld R ; dec P ; Mu R ; st R ; inc P

Deoarece A = 0(9[9{a, 7(4[2], c) },b]) deducem

$C(\mathrm{A}) = 0_S(9_S[9_S\{1_S,\ 7_S(2_S,C(\mathrm{c}))\ \},3_S]) = 0_S(9_S[9_S\{1_S,\ 7_S(2_S,C(\mathrm{c}))\ \},3_S]) =$
$9_S[9_S\{1_S,\ 7_S(2_S,C(\mathrm{c}))\ \},3_S]$; dec P ; print =
$9_S\{1_S,\ 7_S(2_S,C(\mathrm{c}))\ \}$; 3_S ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; dec P ; print =
1_S ; $7_S(2_S,C(\mathrm{c}))$; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; 3_S ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; dec P ; print =
1_S ; 2_S ; $C(\mathrm{c})$; dec P ; ld R ; dec P ; Mu R ; st R ; inc P ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; 3_S ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; dec P ; print =

st x ; inc P ; st y ; inc P ; st x ; inc P ; st y ; inc P ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; st z ; inc P ; dec P ; ld R ; dec P ; Mu R ; st R ; inc P ; dec P ; ld R ; dec P ; Mu R ; st R ; inc P ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; st z ; inc P ; dec P ; ld R ; dec P ; Ad R ; st R ; inc P ; dec P ; print

3.2 Optimizare cod

Se obsrvă că programul de mai sus poate fi simplificat. Eliminând instrucţiunile “inc P ; dec P” al căror efect cumulat este nul obţinem programul

st x ; inc P ; st y ; inc P ; st x ; inc P; st y ; ld R ; dec P ; Ad R ; st R ; inc P ; st z ; ld R ; dec P ; Mu R ; st R ; ld R ; dec P ; Mu R ; st R ; ld R ; dec P ; Ad R ; st R ; inc P ; st z ; ld R ; dec P ; Ad R ; st R ; print

Mai observăm că grupul de instrucţiuni “st R ; ld R ; dec P” are acelaşi efect ca instrucţunea “dec P” ceea ce conduce la programul

st x ; inc P ; st y ; inc P ; st x ; inc P; st y ; ld R ; dec P ; Ad R ; st R ; inc P ; st z ; ld R ; dec P ; Mu R ; dec P ; Mu R ; dec P ; Ad R ; st R ; inc P ; st z ; ld R ; dec P ; Ad R ; st R ; print

7 ALGEBRE CÂT şi ALGEBRE PROIECTIVE

Virgil Emil Căzănescu

May 24, 2008

Dacă $f : A \longrightarrow B$ este o funcţie se numeşte **echivalenţa nucleară** a lui f, sau mai pe scurt nucleul lui f

$$Ker(f) = \{(a, b) \in A \times A : f(a) = f(b)\}.$$

Echivalenţa nucleară a unei funcţii este o relaţie de echivalenţă. În cazul multisortat nucleul este luat pe componente, adică pentru fiecare sort în parte.

1 Congruenţe

O relaţie de echivalenţă într-o Σ-algebră (A_s, A_σ) este de fapt o familie de relaţii de echivalenţe, câte una pentru fiecare mulţime A_s. O congruenţă este o relaţie de echivalenţă care este compatibilă cu operaţiile algebrei. De fapt compatibilatea trebuie cerută numai pentru operaţiile cu argumente căci pentru o constantă $A_\sigma \sim A_\sigma$ rezultă din reflexivitate.

Definiţia 1.1 O congruenţă $\sim$ în Σ-algebra (A_s, A_σ) este o relaţie de echivalenţă cu proprietatea pentru orice $s_1 s_2 \dots s_n \in S^*$, pentru orice $s \in S$, pentru orice $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$, dacă $a_i \sim b_i$ în A_{s_i} pentru orice $1 \leq i \leq n$, atunci

$$A_\sigma(a_1, a_2, \dots a_n) \sim A_\sigma(b_1, b_2, \dots, b_n).$$

Cea mai mică congruenţă este relaţia de egalitate. Cea mai mare congruenţă este relaţia totală.

Propoziţie 1.2 *Dacă $h : \mathcal{A} \longrightarrow \mathcal{B}$ este un Σ-morfism, atunci $Ker(h)$ este o congruenţă.*

Demonstraţie: *Fie $s_1 s_2 \dots s_n \in S^*$, $s \in S$ şi $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$. Presupunem că $a_i \; Ker(h_{s_i}) \; b_i$ în A_{s_i} pentru orice $1 \leq i \leq n$. Rezultă că $h_{s_i}(a_i) = h_{s_i}(b_i)$ pentru orice $1 \leq i \leq n$. Prin urmare*

$$h_s(A_\sigma(a_1, a_2, \dots, a_n)) = B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \dots, h_{s_n}(a_n)) = B_\sigma(h_{s_1}(b_1), h_{s_2}(b_2), \dots, h_{s_n}(b_n)) = h_s(A_\sigma(b_1, b_2, \dots, b_n))$$

prin urmare $A_\sigma(a_1, a_2, \dots, a_n) \; Ker(h_s) \; A_\sigma(b_1, b_2, \dots, b_n)$ deci $Ker(h)$ este congruenţă.

Propoziţie 1.3 *Orice intersecţie de congruenţe este tot o congruenţă.*

Demonstraţie: *Fie $\{\sim^k\}_{k \in K}$ o mulţime de congruenţe şi*

$$\sim \; = \bigcap_{k \in K} \sim^k .$$

Probăm că $\sim$ este congruenţă.

Fie $s_1 s_2 \dots s_n \in S^*$, fie $s \in S$, fie $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $a_i \sim b_i$ în A_{s_i} pentru orice $1 \leq i \leq n$.

Pentru orice $k \in K$ deoarece $\sim^k$ este congruenţă, fiindcă $a_i \sim^k b_i$ în A_{s_i} pentru orice $1 \leq i \leq n$ deducem

$$A_\sigma(a_1, a_2, \dots a_n) \sim^k A_\sigma(b_1, b_2, \dots, b_n).$$

Deci

$$A_\sigma(a_1, a_2, \dots a_n) \sim A_\sigma(b_1, b_2, \dots, b_n).$$

Această propoziţie ne spune că mulţimea congruenţelor unei algebre este o familie Moore. Încercaţi să caracterizaţi operatorul de închidere asociat.

1.1 Congruenţe de grupuri

Grupurile sunt privite ca algebre cu trei operatii: una binară $\times$, una unara * şi o constantă e. Două condiţii se impun pentru conceptul de congruenţa

1. $a \sim b$ şi $c \sim d$ implică $a \times c \sim b \times d$,
2. $a \sim b$ implică $a^* \sim b^*$.

Practic rămâne numai una deoarece a doua condiţie este o consecinţa a primeia: folosind reflexivitatea din $a^* \sim a^*$, $a \sim b$ şi $b^* \sim b^*$ deducem $a^* \times a \times b^* \sim a^* \times b \times b^*$, prin urmare $b^* \sim a^*$, deci $a^* \sim b^*$.

Propoziţie 1.4 *În orice grup conceptul de congruenţa este echivalent cu cel de subgrup normal*

Demonstraţie:
1 Dată o congruenţă clasa elementului neutru este un subgrup normal.
2 Dat un subgrup normal N relaţia definită prin

$$a \sim b \text{ dacă şi numai dacă } a \times b^* \in N$$

este o congruenţă. □

Conceptul de parte stabilă coincide cu cel de subgrup.
Conceptul de morfism de Σ-algebră coincide cu cel de morfism de grup.

1.2 Congruenţe de inele

Pentru simplicitate vom prefera cazul comutativ.

Signatura conceptului de inel conţine pe lângă cele trei operaţii corespunzătoare grupurilor încă două simboluri de aritate 2, respectiv 0 corespunzătoare structurii monoidului multiplicativ. Având în vedere cazul grupurilor este suficient să punem condiţia de congruenţă numai pentru cele două operaţii binare.

Propoziţie 1.5 *În orice inel comutativ conceptul de congruenţă este echivalent cu cel de ideal.*

Demonstraţie:
1 Dată o congruenţă clasa elementului neutru este un ideal.
2 Dat un ideal N relaţia definită prin

$$a \sim b \text{ dacă şi numai dacă } a - b \in N$$

este o congruenţă. □

Conceptul de parte stabilă coincide cu cel de subinel.
Conceptul de morfism de Σ-algebră coincide cu cel de morfism de inel.

1.3 Concluzie

Conceptul de congruenţă este cel important. Este un fapt întâmplător că pentru anumite structuri algebrice congruenţele pot fi caracterizate de unele substructuri particulare.

2 Algebre cât

Proprietatea de universalitate a mulţimii cât. Fie $\sim$ o relaţie de echivalenţa în mulţimea A. Fie $A/\sim$ mulţimea cât şi $\hat{}: A \longrightarrow A/\sim$ surjecţia naturală de factorizare. Pentru orice funcţie $f : A \longrightarrow B$ dacă $\sim \subseteq Ker(f)$, atunci există o unică funcţie $f^{\#} : A/\sim \longrightarrow B$ cu proprietatea $\hat{}\,; f^{\#} = f$. □

Fie (A_s, A_σ) o Σ-algebră şi $\sim$ o congruenţă. Definim operaţiile algebrei cât

$$\mathcal{A}/\sim = (\{A_s/\sim\}_{s\in S}, A/\sim_\sigma)$$

pentru orice $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $a_i \in A_{s_i}$ pentru orice $1 \le i \le n$ prin

$$A/\sim_\sigma (\widehat{a_1}, \widehat{a_2}, \dots, \widehat{a_n}) = A_\sigma(a_1, \widehat{a_2}, \dots, a_n).$$

Teorema 2.1 Proprietatea de universalitate a algebrei cât. *Pentru orice morfism de Σ-algebre $f : \mathcal{A} \longrightarrow \mathcal{B}$ dacă $\sim \subseteq Ker(f)$, atunci există un unic morfism de Σ-algebre $f^{\#} : \mathcal{A}/\sim \longrightarrow \mathcal{B}$ cu proprietatea $\hat{}\,; f^{\#} = f$.*

Demonstraţie: Din proprietatea de universalitate a mulţimii cât deducem existenţa unei unice funcţii $f^{\#} : A/\sim \longrightarrow B$ cu proprietatea $\hat{}\,; f^{\#} = f$. Observăm că $f^{\#}_s(\hat{a}) = f_s(a)$ pentru orice $s \in S$ şi $a \in A_s$. Mai trebuie să demonstrăm că funcţia $f^{\#}$ este un morfism de Σ-algebre.

Fie $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $a_i \in A_{s_i}$ pentru orice $1 \le i \le n$. Observăm că

$$f^{\#}_s(A/\sim_\sigma (\widehat{a_1}, \widehat{a_2}, \dots, \widehat{a_n})) = f^{\#}_s(A_\sigma(a_1, \widehat{a_2}, \dots, a_n)) = f_s(A_\sigma(a_1, a_2, \dots, a_n)) = B_\sigma(f_{s_1}(a_1), f_{s_2}(a_2), \dots, f_{s_n}(a_n)) =$$

$$B_\sigma(f^{\#}_{s_1}(\widehat{a_1}), f^{\#}_{s_2}(\widehat{a_2}), \dots, f^{\#}_{s_n}(\widehat{a_n})).$$

Deci $f^{\#}$ este morfism de Σ-algebre. □

Să se arate că pentru orice Σ-morfism $h : \mathcal{A} \longrightarrow \mathcal{B}$ algebrele $\mathcal{A}/Ker(h)$ şi $h(\mathcal{A})$ sunt izomorfe.

3 Algebre proiective

Fie (S, Σ) o signatură multisortată. Vom lucra în categoria Σ-algebrelor $\mathcal{A}lg_\Sigma$. Pentru simplicitate vom scrie pe scurt algebră şi morfism în loc de Σ-algebră şi respectiv morfism de Σ-algebre.

Cei care nu sunt interesaţi de toate detaliile pot sări peste propoziţia următoare dar în continuare pot lua drept definiţie pentru Σ-algebre: epimorfism = morfism cu toate componentele surjective.

Intr-o categorie, un morfism $e : A \longrightarrow B$ se numeşte epimorfism dacă pentru orice pereche de morfisme $f : B \longrightarrow C$ şi $g : B \longrightarrow C$ dacă $e; f = e; g$, atunci $f = g$.

Propoziţie 3.1 *Fie $\mathcal{A}$ şi $\mathcal{B}$ două (S, Σ)-algebre şi $h : \mathcal{A} \longrightarrow \mathcal{B}$ un (S, Σ)-morfism. Morfismul h este epimorfism în $\mathcal{A}_\Sigma$ dacă şi numai dacă h_s este surjectiv pentru orice $s \in S$.*

Demonstraţie: Fie $\mathcal{A} = (\{A_s\}_{s\in S}, \{A_\sigma\}_{\sigma\in\Sigma})$ şi $\mathcal{B} = (\{B_s\}_{s\in S}, \{B_\sigma\}_{\sigma\in\Sigma})$.
($\Leftarrow$) Fie $\mathcal{C}$ o (S, Σ)-algebră şi $m, n : \mathcal{B} \longrightarrow \mathcal{C}$ două morfisme astfel încât $h; m = h; n$. Rezultă că $h_s; m_s = h_s; n_s$ pentru orice $s \in S$. Deoarece pentru orice $s \in S$, h_s este surjectiv rezultă că h_s este epimorfism în categoria mulţimilor, deci $m_s = n_s$. Am demonstrat că $m = n$.
($\Rightarrow$) Fie $C = \{C_s\}_{s\in S}$ o mulţime S-sortată, cu $C_s = B_s \amalg (B_s - h_s(A_s))$ pentru orice $s \in S$. Simbolul $\amalg$ indică o reuniune disjunctă de mulţimi.

Fie $\iota : B \longrightarrow C$ funcţia incluziune de mulţimi S-sortate şi $r : B \longrightarrow C$ definit prin

$$r_s(b) = \begin{cases} b, & \text{dacă } b \in h_s(A_s) \\ \overline{b} & \text{dacă } b \in B_s - h_s(A_s) \end{cases}$$

pentru orice $s \in S$ şi $b \in B_s$ (pentru $b \in B_s - h_s(A_s)$ am notat cu $\overline{b}$ elementul care îi corespunde în C_s).

În continuare vom defini pe C o structură de (S, Σ)-algebră astfel încât ι şi r să devină morfisme de (S, Σ)-algebre.

Fie $\sigma \in \Sigma_{s1\cdots sn, s}$. Definim:
1) $C_\sigma(b1, \dots, bn) = B_\sigma(b1, \dots, bn)$ dacă $bi \in B_{si}$ pentru orice $i \in [n]$;
2) $C_\sigma(r_{s1}(b1), \dots, r_{sn}(bn)) = r_s(B_\sigma(b1, \dots, bn))$ dacă $bi \in B_{si}$ pentru orice $i \in [n]$;
3) în celelalte situaţii operaţia C_σ poate fi definită oricum.

Să demonstrăm că definiţia de mai sus nu este contradictorie.

Fie $b1 \in B_{s1}$, ..., $bn \in B_{sn}$ astfel încât $r_{si}(bi) = bi \in B_{si}$ pentru orice $i \in [n]$. Atunci $b_{si} \in h_{si}(A_{si})$ de unde rezultă că $B_\sigma(b1, \ldots, bn) \in h_s(A_s)$ şi deci $B_\sigma(b1, \ldots, bn) = r_s(B_\sigma(b1, \ldots, bn))$.

Din 1) rezultă că ι este morfism iar din 2) rezultă că r este morfism.

Am definit astfel o (S, Σ)-algebră $\mathcal{C} = (\{C_s\}_{s \in S}, \{C_\sigma\}_{\sigma \in \Sigma})$ şi două morfisme $\iota, r : \mathcal{B} \longrightarrow \mathcal{C}$. Se observă că $h\iota = hr$ deoarece pe $h(A)$ cele două morfisme acţionează la fel. Deoarece h este epimorfism rezultă că $\iota = r$ de unde obţinem $B_s = h_s(A_s)$ pentru orice $s \in S$ (dacă există $s \in S$ astfel încât $B_s - h_s(A_s) \neq \emptyset$ atunci există $b \in B_s - h_s(A_s)$ şi $\iota(b) = b$, $r(b) = \overline{b}$ deci $\iota \neq r$ ceea ce contrazice faptul că h este epimorfism). Am demonstrat că h_s este surjecţie pentru orice $s \in S$. □

Intr-o categorie, un obiect P se numeşte *proiectiv* dacă pentru orice epimorfism e : A $\longrightarrow$ B si pentru orice morfism f : P $\longrightarrow$ B există un morfism g : P $\longrightarrow$ A astfel încât g;e = f.

Propoziţie 3.2 *In Alg_Σ orice algebră liberă este proiectivă.*

Demonstraţie: Fie $\mathcal{A}$, $\mathcal{B}$ două (S, Σ)-algebre şi X o mulţime S-sortată de variabile. Dacă $e : \mathcal{A} \longrightarrow \mathcal{B}$ este un morfism cu toate componentele surjective şi $f : T_\Sigma(X) \longrightarrow \mathcal{B}$ este un morfism oarecare trebuie să demonstrăm că există un morfism $g : T_\Sigma(X) \longrightarrow \mathcal{A}$ astfel încât $g; e = f$.

Pentru a defini morfismul g este suficient să dăm acţiunea lui pe variabile.

Fie $s \in S$ şi $x \in X_s$. Deoarece e_s este surjectiv, deci există $a_x \in A_s$ astfel încât $f_s(x) = e_s(a_x)$. Definim g ca fiind unicul morfism cu proprietatea că $g_s(x) = a_x$ pentru orice $s \in S$ şi pentru orice $x \in X_s$. Este evident faptul că $(g; e)_s(x) = f_s(x)$ pentru orice $s \in S$ şi orice $x \in X_s$. Deoarece morfismele $g; e$ şi f coincid pe generatorii algebrei libere $T_\Sigma(X)$ rezultă că $g; e = f$ şi demonstraţia este încheiată. □

Comentariu. Faptul că $g_s(x)$ poate fi ales arbitrar în $e_s^{-1}(\{f_s(x)\})$ nu garantează unicitatea lui g. □

Lemă 3.3 *Orice subalgebră a unei algebre libere este algebră liberă.*

Demonstraţie: Într-o algebră definim relaţia

$$a \vdash b \text{ dacă şi numai dacă } a = \sigma(\ldots, b, \ldots).$$

Observăm că în orice algebră liberă relaţia $\vdash$ este noetheriană.

Fie $\mathcal{P}$ o subalgebră a unei algebre libere $\mathcal{L}$. Relaţia $\vdash$ este noetheriană şi în subalgebra $\mathcal{P}$.

Fie X mulţimea tuturor elementelor din $\mathcal{P}$ care nu sunt rezultatul aplicării vreunei operaţii din $\mathcal{P}$ unor elemente din $\mathcal{P}$. Vom dovedi ca $\overline{X}$, subalgebra generată de X, este chiar $\mathcal{P}$. Presupunem prin absurd că există $p_0 \in \mathcal{P} - \overline{X}$. Deoarece $p_0 \notin X$ el este rezultatul aplicării cel puţin al unei operaţii. În plus cel puţin unul dintre argumentele acestei operaţii nu este din $\overline{X}$ deoarece în caz contrar obţinem contradicţia $p_0 \in \overline{X}$. Prin urmare există $p_1 \in \mathcal{P} - \overline{X}$ astfel încât $p_0 \vdash p_1$. Continuând raţionamentul prin inducţie ar rezulta că relaţia $\vdash$ nu este noetheriană în $\mathcal{P}$, o contradicţie.

Observăm că $\mathcal{P}$ este algebră Peano peste X. Chiar din definiţia lui X rezultă că rezultatul aplicării unei operaţii din $\mathcal{P}$ unor elemente din $\mathcal{P}$ nu este în X. Ultima condiţie din definiţia algebrelor Peano este adevărată în $\mathcal{P}$ deoarece este adevărată în $\mathcal{L}$.

Deci $\mathcal{P}$ este liber generată de X deoarece este Peano peste X.

Propoziţie 3.4 *In Alg_Σ orice algebră proiectivă este liberă.*

Demonstraţie: Fie $\mathcal{P}$ o algebră proiectivă. Fie $\epsilon : T_\Sigma(P) \longrightarrow \mathcal{P}$ unicul morfism a cărui restricţie la P este 1_P, identitatea lui P. Evident ϵ este epimorfism.

Deoarece $\mathcal{P}$ este o algebră proiectivă există un morfism $g : \mathcal{P} \longrightarrow T_\Sigma(P)$ cu proprietatea $g; \epsilon = 1_P$. Observăm că g este o injecţie. Prin urmare algebra $\mathcal{P}$ este izomorfă cu subalgebra $g(P)$ a lui $T_\Sigma(P)$.

Algebra $g(P)$ este liberă deoarece este subalgebră a unei algebre libere.

Algebra P este liberă deoarece este izomorfă cu o algebră liberă.

8 SPRE ABSTRACTIZAREA TIPURILOR DE DATE

Virgil Emil Căzănescu

March 22, 2008

1 Ecuaţii

Să analizăm conceptul de axiomă aşa cum apare el în algebră. De exemplu comutativitatea şi asociativitatea se scriu

$$(\forall x \forall y) x?y = y?x \qquad (\forall x \forall y \forall z) x?(y?z) = (x?y)?z.$$

Ce sunt acestea? Sunt egalităţi de două expresii cuantificate universal prin mulţimea variabilelor conţinute în cele două expresii. Deci o astfel de axiomă are forma

$$(\forall X) l \doteq r$$

unde l şi r sunt din algebra liber generată de mulţimea X de variabile.

Ce înseamnă că o axiomă este adevărată într-o algebră $\mathcal{D}$? Intuitiv este necesar ca rezultatul evaluării celor două expresii l şi r în algebra $\mathcal{D}$ să fie acelaşi indiferent de valorile date în $\mathcal{D}$ variabilelor din X. Această idee intuitivă conduce la:

Definiţia 1.1 Axioma $(\forall X) l \doteq r$ este satisfăcută în algebra $\mathcal{D}$ dacă şi numai dacă pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ este adevărată egalitatea $h(l) = h(r)$.

În continuare vom folosi pentru $(\forall X) l \doteq r$ termenul de *ecuaţie* în locul celui de axiomă, pentru a ne conforma cu terminologia internaţională.

În plus vor intra în joc şi aşa zisele *ecuaţii condiţionate*. De exemplu

$$(\forall x \forall y \forall z)(x * y = x * z \Rightarrow y = z)$$

ceea ce corespunde axiomei de simplificare la stânga care este adevărată în orice grup sau în orice monoid liber.

2 Ecuaţii condiţionate

În logica ecuaţională o axiomă este o implicaţie

$$a_1 = c_1, a_2 = c_2, \ldots, a_n = c_n \Rightarrow a = c$$

unde ipoteza este o conjuncţie de egalităţi formale şi concluzia o egalitate formală. Toată implicaţia este cuantificată universal, fapt care nu apare scris mai sus. În acest cadru o axiomă, numită în continuare şi ecuaţie condiţionată a logicii ecuaţionale, poate fi scrisă sub forma

$$(\forall X)\ a \doteq_s c \textbf{ if } H$$

unde a şi c sunt elemente de acelaşi sort s, iar H este o mulţime **finită** de egalităţi formale din algebra liber generată de mulţimea X de variabile. Ipoteza implicaţiei este dată de mulţimea H.

Pentru a verifica dacă o algebră $\mathcal{D}$ satisface axioma de mai sus se dau valori arbitrare variabilelor din X în $\mathcal{D}$ fapt ce poate fi făcut printr-o funcţie arbitrară $f : X \longrightarrow \mathcal{D}$ sau echivalent printr-un morfism arbitrar $h : T_\Sigma(X) \longrightarrow \mathcal{D}$. Apoi se evaluează expresiile din H pentru a se verifica dacă rezultatul evaluarii conduce la egalităţi adevărate, caz în care trebuie ca $h_s(a) = h_s(c)$. Deci Σ-algebra $\mathcal{D}$ satisface ecuaţia condiţionată $(\forall X) a \doteq_s c$ **if** H fapt notat prin

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

dacă şi numai dacă

$$(\forall h : T_\Sigma(X) \longrightarrow \mathcal{D})\ (\forall u =_t v \in H) h_t(u) = h_t(v) \text{ implică } h_s(a) = h_s(c).$$

Credem că este bine să menţionăm diferenţa esenţială între semnele $\doteq_s$ şi $=$, diferenţă care va fi menţinută constant pe parcursul întregului text. Egalul peste care s-a pus un punct ($\doteq_s$) indică o egalitate formală care poate fi adevărată sau falsă. Egalul $=$ are semnificaţia uzuală indicând deobicei o egalitate adevărată.

3 Necesitatea utilizării cuantificatorilor în ecuaţii

Vom ilustra printr-un exemplu necesitatea utilizării cuantificatorilor în ecuaţiile logicii ecuaţionale multisortate.

Fie signatura $S = \{a, bool\}$ şi $\Sigma = \{g, F, T\}$. Rangurile simbolurilor de operaţii sunt date prin desenul următor :

$$a \xrightarrow{\ g\ } bool \underset{T}{\overset{F}{\leftleftarrows}} \cdot$$

Vom lucra cu două Σ–algebre.

Σ–algebra iniţială este $\mathcal{I} = (\emptyset, \{F, T\}, I_g, I_F, I_T)$ unde $F \neq T$, $I_F = F$, $I_T = T$ şi $I_g : \emptyset \longrightarrow \{F, T\}$ este funcţia incluziune.

Pentru orice variabilă x de sort a, Σ-algebra liber generată de această variabilă $T_\Sigma(\{x\}, \emptyset)$ are suporturile $\{x\}$ şi {g(x),F,T}.

Are loc relaţia $\mathcal{I} \models_\Sigma F = T$? Sau mai intuitiv: este egalitatea $F = T$ adevărată în algebra $\mathcal{I}$? Vom arăta că raspunsul depinde de algebra liberă în care este scrisă egalitatea.

Sunt posibile cel puţin două variante.

1. $\mathcal{I} \models_\Sigma (\forall\emptyset)T = F \quad \Leftrightarrow \quad \forall h : \mathcal{I} \longrightarrow \mathcal{I}$ morfism, $h_{bool}(F) = h_{bool}(T)$,
   ceea ce este *fals* deoarece $h_{bool}(F) = F \neq T = h_{bool}(T)$.

2. $\mathcal{I} \models_\Sigma (\forall x)T = F \quad \Leftrightarrow \quad \forall h : T_\Sigma(\{x\}, \emptyset) \longrightarrow \mathcal{I}$ morfism, $h_{bool}(F) = h_{bool}(T)$,
   ceea ce este *adevărat* deoarece nu există nici un morfism $h : T_\Sigma(\{x\}, \emptyset) \longrightarrow \mathcal{I}$.

Am arătat că $\mathcal{I} \models_\Sigma (\forall\emptyset)T = F$ este falsă şi că $\mathcal{I} \models_\Sigma (\forall x)T = F$ este adevărată. Dacă omitem cuantificatorii obţinem: "$\mathcal{I} \models_\Sigma T = F$ este falsă şi $\mathcal{I} \models_\Sigma T = F$ este adevărată."

Contradicţia obţinută prin omiterea cuantificatorilor din faţa egalităţii $F = T$ arată că în logica ecuaţională multisortată prezenţa cuantificatorilor în ecuaţii este necesară.

4 In primul rând semantica

Pentru orice algebră $\mathcal{D} = (D_s, D_\sigma)$ notăm cu

$$Sen(\mathcal{D}) = \{a \doteq_s c : s \in S,\ a, c \in D_s\}$$

mulţimea *propoziţiilor* sale. Propoziţiile sunt de fapt egalităţi formale care pot fi adevărate sau false.

Să observăm că $Sen(\mathcal{D})$ se poate identifica cu produsul cartezian $D \times D$. Poate cea mai bună reprezentare a unei propoziţii din $\mathcal{D}$ este un triplet format dintr-un sort s si două elemente de sort s din $\mathcal{D}$.

Definiţia 4.1 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 $T_\Sigma(X)$. □

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 $(\forall X)l \doteq_s r$ **if** $\emptyset$.

Definiţia 4.2 Algebra $\mathcal{D}$ *satisface* ecuaţia condiţionată $(\forall X)l \doteq_s r$ **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 $h_s(l) = h_s(r)$. □

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 $h : T_\Sigma(X) \longrightarrow \mathcal{D}$.

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

Definiţia 4.3 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.

Un morfism de Σ-algebre între două Γ-algebre se numeşte morfism de Γ-algebre sau mai scurt Γ-morfism. □

5 Punctul de vedere local

Fixăm o algebră $\mathcal{A}$ şi lucrăm cu propoziţii din $Sen(\mathcal{A})$. Acesta este punctul *local* de vedere al logicii ecuaţionale. Cazul în care ne interesează propoziţii din algebre diferite este numit *global* dar va fi puţin folosit în acest curs. În cazul global o propoziţie din $\mathcal{D}$ va fi scrisă $(\forall \mathcal{D})a \doteq_s c$ în loc de $a \doteq_s c$. Notaţia fără cuantificatori folosită în cazul local nu contrazice ceea ce am scris despre necesitatea utilizării cuantificatorilor în ecuaţii. Pur si simplu nu scriem cuantificatorul pentru că fixând algebra $\mathcal{A}$ el va fi mereu acelaşi $(\forall \mathcal{A})$ şi deci îl ştim chiar dacă nu-l vedem scris.

O altă idee pe care dorim să o subliniem este folosirea unei algebre arbitrare $\mathcal{A}$ în locul unei algebre libere. Principalele rezultate privind rescrierile, care se referă atât la rescrierile de termeni cât şi la rescrierile modulo ecuaţii, rămân valabile în acest cadru mai general. Aceasta prezentare unitară a diverselor tipuri de rescriere este de fapt contribuţia autorului la modernizarea lecţiilor despre rescriere. Există în lume trei cărţi privind rescrieriile. Cea mai veche este Term Rewriting and All That, scrisă de Franz Baader şi Tobias Nipkow.

Menţionăm că algebra $\mathcal{A}$ nu are legătură cu algebra $T_\Sigma(X)$ folosită în vreo axiomă $(\forall X)l =_s r$ **if** H din Γ. Menţionăm şi că în axiome diferite putem folosi algebre libere diferite. Aceasta corespunde cazului practic, de exemplu scriem comutativitatea $xy = yx$ într-o algebră liberă cu doi generatori x şi y iar asociativitatea $(xy)z = x(yz)$ într-o algebră liberă cu trei generatori x, y şi z.

6 Semantică şi Corectitudine

Vom lucra într-o Σ-algebră $\mathcal{A}$. Vom grupa într-o relaţie toate tautologiile(propoziţiile valide) din algebră ale logicii ecuaţionale.

Fie $\equiv_\Gamma^{\mathcal{A}}$ relaţia pe A definită prin

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

Dacă nu există pericol de confuzie vom prefera să scriem $\equiv_\Gamma$ în loc de $\equiv_\Gamma^{\mathcal{A}}$.

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.

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

Definim regula de deducţie a substituţiei utilizată atât în logica ecuaţională cât şi în rescrierea termenilor.

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 $h_s(l) \doteq_s h_s(r)$.

Lemă 6.1 *Pentru orice morfism $f : \mathcal{A} \longrightarrow \mathcal{M} \models \Gamma$, $Ker(f)$ este închis la* **Sub**$_\Gamma$.

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 $h_t(u)\ Ker(f)\ h_t(v)$ pentru orice $u \doteq_t v \in H$. Prin urmare $(h; f)_t(u) = (h; f)_t(v)$ 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 $h_s(l)\ Ker(f)\ h_s(r)$.□

Propoziţie 6.2 *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. □

Propoziţie 6.3 *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 $h_s(l) = h_s(r)$. □

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 6.4 $\mathcal{A}_\Gamma \models \Gamma$

Demonstraţie: Se aplică propoziţiile 6.2 şi 6.3. □

Teorema 6.5 *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 aplicăm proprietatea de universalitate a algebrei cât. Într-adevăr $a \equiv_\Gamma c$ implică $h(a) = h(c)$ deoarece $h\colon \mathcal{A} \longrightarrow \mathcal{B} \models \Gamma$. □

Corolar 6.6 *Dacă* $\mathcal{A}$ *este* Σ*-algebră iniţială, atunci* $\mathcal{A}_\Gamma$ *este* Γ*-algebră iniţială.*

Corolar 6.7 *Pentru orice signatură* Σ *şi pentru orice mulţime* Γ *de ecuaţii condiţionate există o* Γ*-algebră iniţială.*

Propoziţie 6.8 *Fie* $h : \mathcal{A} \longrightarrow \mathcal{B}$ *un morfism. Dacă* $a \equiv_\Gamma^{\mathcal{A}} c$*, atunci* $h(a) \equiv_\Gamma^{\mathcal{B}} h(c)$.

Demonstraţie: Fie $f : \mathcal{B} \longrightarrow \mathcal{C} \models \Gamma$. Din $a \equiv_\Gamma^{\mathcal{A}} c$ folosind morfismul $h; f : \mathcal{A} \longrightarrow \mathcal{C} \models \Gamma$ deducem $(h; f)(a) = (h; f)(c)$, prin urmare $f(h(a)) = f(h(c))$. Deci $h(a) \equiv_\Gamma^{\mathcal{B}} h(c)$.

7 Problema programării prin rescriere

Principala problemă este : "poate o maşină să demonstreze că $a \equiv_\Gamma c$*?".*

Se ştie că în unele cazuri rescrierile ne dau o soluţie.

8 Tipuri abstracte de date

Am văzut în lecţiile precedente că orice signatură determină prin algebra sa iniţială un tip abstract de date. Tipul abstract de date al numerelor naturale a fost determinat de signatura formată din constanta 0 şi operaţia unară cunoscută sub numele de succesor.

Acum am pus în evidenţă un instrument mai puternic deoarece orice signatură împreună cu o mulţime Γ de ecuaţii condiţionate determină prin Γ-algebra iniţială, a cărei existenţă am dovedit-o mai sus, un tip abstract de date.

8.1 Tipul abstract al numerelor naturale - continuare

Considerăm signatura cu un singur sort *nat*, o singură constantă de sort nat şi o singură operaţie unară cu argument şi rezultat de sort nat:

```
sort nat .
op 0 : ⟶ nat .
op s : nat ⟶ nat .
```

Elementele algebrei iniţiale sunt

$$0,\ s(0),\ s(s(0)),\ s(s(s(0))),\ s(s(s(s(0)))),\ \ldots$$

şi ele reprezintă numerele naturale 0 1 2 3 4 ...

Propoziţie 8.1 *Algebra* $(N, 0_N, s_N)$ *definită prin:* N *este mulţimea numerelor naturale,* 0_N *este numărul natural zero şi* $s_N(n) = n + 1$ *pentru orice număr natural* n*; este iniţială.*

Propoziţia anterioară ne arată cum pot fi definite numerele naturale prim metoda algebrei iniţiale ca tip abstract de date.

Deocamdată prin signatura de mai sus calculatorul învaţă numerele naturale dar nu ştie încă să calculeze. Să-l învăţăm să adune şi să înmulţească. La cele de mai sus adăugăm.

```
op _+_ : nat nat ⟶ nat .
var X Y : nat .
eq X + 0 = 0 .
```

eq X + s(Y) = s(X+Y) .
op _*_ : nat nat ⟶ nat .
eq X * 0 = 0 .
eq X * s(Y) = X*Y + X .

Trebuie să menţionăm că egalităţiile de mai sus sunt folosite în două moduri:
a) ca ecuaţii care împreună cu signatura de patru operaţii formează o mulţime Γ pentru a defini o structură algebrică şi
b) ca reguli de rescriere în timpul execuţiei programelor.

Propoziţie 8.2 *Algebra* $\mathcal{N} = (N, 0_N, s_N, +_N, *_N)$ *este* Γ*-algebră iniţială.*

Demonstraţie: Fie $\mathcal{A} = (A, O_A, s_A, +_A, *_A)$ o Γ-algebră. Menţionăm că algebra $\mathcal{A}$ satisfac ecuaţiile din Γ, adică

1) $a +_A 0_A = a$ pentru orice a din A,
2) $a +_A s_A(b) = s_A(a + b)$ pentru orice a, b din A,
3) $a *_A 0_A = 0_A$ pentru orice a din A,
4) $a *_A s_A(b) = a *_A b +_A a$ pentru orice a, b din A.

Vom proba că există un unic morfism de Γ-algebre de la $\mathcal{N}$ la $\mathcal{A}$.
Să începem cu unicitatea. Dacă $h : \mathcal{N} \longrightarrow \mathcal{A}$ este un Γ-morfism, atunci $h : (N, 0_N, s_N) \longrightarrow (A, 0_A, s_A)$ este morfism prin urmare coincide cu unicul morfism dat de propoziţia de mai sus.
Pentru a demonstra existenţa nu avem decât o singură şansă şi anume să dovedim că unicul morfism $h : (N, 0_N, s_N) \longrightarrow (A, 0_A, s_A)$ este şi morfism de Γ-algebre. Reamintim că

$$h(0_N) = 0_A \text{ şi pentru orice } n \text{ număr natural } h(n + 1) = s_A(h(n)).$$

Probăm că $h(n +_N m) = h(n) +_A h(m)$ prin inducţie după m

$h(n +_N 0_N) = h(n) = h(n) +_A 0_A = h(n) +_A h(0_N)$ şi
$h(n+_N(m+1)) = h(s_N(n+_N m)) = s_A(h(n+_N m)) = s_A(h(n)+_A h(m)) = h(n)+_A s_A(h(m)) = h(n)+_A h(m+1)$.

Probăm că $h(n *_N m) = h(n) *_A h(m)$ prin inducţie după m

$h(n *_N 0_N) = h(0_N) = 0_A = h(n) *_A 0_A = h(n) *_A h(0_N)$ şi
$h(n *_N (m + 1)) = h(n *_N m +_N n) = h(n *_N m) +_A h(n) = (h(n) *_A h(m)) +_A h(n) = h(n) *_A s_A(h(m)) = h(n) *_A h(m + 1)$. □

Propoziţia anterioară demonstrează corectitudinea definiţiei de mai sus. Deoarece algebra $\mathcal{N}$ este iniţială rezultă că ea este izomorfă cu Γ-algebra iniţială, deci specificaţia de mai sus caracterizează prin Γ-algebra sa iniţială tipul de date al numerelor naturale.

2 LOGICA ECUAŢIONALĂ MULTISORTATĂ

Virgil Emil Căzănescu

April 26, 2008

Fie Γ o mulţime de ecuaţii condiţionale şi o Σ-algebra $\mathcal{A}$ fixată.
Mulţimea propoziţiilor adevărate, tautologiile, din $\mathcal{A}$ este chiar congruenţa semantică

$$\equiv_\Gamma = \{a \doteq_s b \in Sen(A) \colon (\forall M \models \Gamma)(\forall f \colon A \longrightarrow M) f_s(a) = f_s(b)\}.$$

Conform tradiţiei mai scriem

$$\models a \doteq_s b \text{ dacă şi numai dacă } a \equiv_\Gamma b$$

Căutăm o mulţime corectă şi completă de reguli de deducţie pentru $\equiv_\Gamma^{\mathcal{A}}$.

1 Reguli de deducţie, corectitudine

Definim prin $\mathbf{R_E}$ mulţimea următoarelor reguli de deducţie pentru logica ecuaţională multisortată:

R	$a \doteq_s a$
S	$a \doteq_s b$ implică $b \doteq_s a$
T	$a \doteq_s b$ şi $b \doteq_s c$ implică $a \doteq_s c$
CΣ	Pentru orice $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$: $a_i \doteq_{s_i} b_i$ pentru $1 \leq i \leq n$ implică $A_\sigma(a_1, a_2, \ldots, a_n) \doteq_s A_\sigma(b_1, b_2, \ldots, b_n)$.
Sub$_\Gamma$	Pentru orice $(\forall X)\, l \doteq_s r$ **if** $H \in \Gamma$ şi pentru orice $h \colon T_\Sigma(X) \longrightarrow \mathcal{A}$ $h_t(u) \doteq_t h_t(v)$ pentru orice $u \doteq_t v \in H$ implică $h_s(l) \doteq_s h_s(r)$

Conform tradiţiei notăm prin $\vdash a \doteq_s b$ faptul că egalitatea formală $a \doteq_s b$ este demonstrabilă cu regulile de mai sus.

Observaţia 1.1 *Observăm că o mulţime de egalităţi formale este închisă la regula* **T** *dacă şi numai dacă este o relaţie tranzitivă.*

Mai observăm că astfel de observaţii pot fi făcute pentru oricare din regulile de mai sus.

Teorema 1.2 *Regulile de deducţie* $\mathbf{R_E}$ *sunt corecte pentru* $\equiv_\Gamma$.

Demonstraţie: Deoarece $\equiv_\Gamma$ este congruenţă rezultă ea este închisă la primele patru reguli, prin urmare ele sunt corecte pentru $\equiv_\Gamma$. Corectitudinea ultimei reguli rezultă din închiderea congruenţei semantice la substituţii. □

Corolar 1.3 $\vdash a \doteq_s b$ *implică* $\models a \doteq_s b$. □

Demonstraţie: Deoarece congruenţa semantică este închisă la $\mathbf{R_E}$ şi deoarece mulţimea egalităţilor formale demonstrabile este cea mai mică mulţime închisă la $\mathbf{R_E}$.

2 Completitudine

Pentru orice $s \in S$ şi $a, b \in A_s$ se defineşte relatia $\sim_\Gamma$ in $\mathcal{A}$ prin:

$$a \sim_\Gamma b \Longleftrightarrow \vdash a \doteq_s b.$$

Deoarece regulile **R, S** şi **T** sunt în $\mathbf{R_E}$ deducem că $\sim_\Gamma$ este o echivalenţă. Mai mult, deoarece **C**Σ este în $\mathbf{R_E}$ rezultă că $\sim_\Gamma$ este congruenţă.

Fie $\mathcal{A}_\Gamma$ câtul lui $\mathcal{A}$ prin $\sim_\Gamma$ şi fie $\eta_\mathcal{A} \colon \mathcal{A} \longrightarrow \mathcal{A}_\Gamma$ Σ-morfismul canonic de factorizare. Deoarece $\sim_\Gamma$ este închisă la substituţii rezulta:

Observaţia 2.1 $\mathcal{A}_\Gamma \models \Gamma$

Teorema 2.2 *Teoremă de completitudine:* $\models a \doteq_s b$ *implică* $\vdash a \doteq_s b$.

Demonstraţie: Fie $\models a \doteq_s b$. Din definiţia tautologiilor deducem $(\forall h\colon A \longrightarrow B \models \Gamma)\ h_s(a) = h_s(b)$. Întrucât $\mathcal{A}_\Gamma \models \Gamma$ şi $\eta_\mathcal{A}\colon \mathcal{A} \longrightarrow \mathcal{A}_\Gamma$ este Σ-morfism rezultă că $\eta_\mathcal{A}(a) = \eta_\mathcal{A}(b)$. Prin urmare $a \sim_\Gamma b$, deci $\vdash a \doteq_s b$. □

În concluzie congruenţele $\sim_\Gamma$ şi $\equiv_\Gamma$ coincid. Prin urmare algebra $\mathcal{A}_\Gamma$ şi morfismul $\eta_\mathcal{A}$ coincid cu cele introduse într-un mod numai aparent diferit în lecţia precedentă. Proprietatea de universalitate a algebrei $\mathcal{A}_\Gamma$ demonstrată atunci rămâne adevărată şi în noul context.

Următorul pas după o teoremă de completitudine care ne asigură existenţa unei demonstraţii pentru orice tautologie este de a gasi aceste demonstraţii. Informaticienii sunt ceva mai pretenţioşi deoarece doresc ca acete demonstraţii să fie găsite de un calculator.

3 RESCRIERE LOCALĂ

VEC

May 26, 2008

Rescrierile sunt un fapt pe care-l întâlnim din primii ani de şcoală. De exemplu în şirul de egalităţi

$$2 * (3 + 4) = 2 * 7 = 14$$

facem două rescrieri(înlocuiri): 3+4 esre rescris în 7 şi apoi 2*7 este rescris în 14. Aceste rescrieri sunt permise de regulile adunării şi înmulţirii. Partea expresiei care rămâne neschimbată deoarece rescrierea are loc în interiorul ei se numeşte context. La prima rescriere contextul este $2 * z$, unde z este un semn special care arată locul în care se face rescrierea. Deoarece a doua rescriere se face la vârf contextul este z.

De obicei rescrierile se fac de la expresii mai complicate spre expresii mai simple. Rescrierea lui 3+4 în 7 este ceva firesc, pe când rescrierea lui 7 în 3+4 este ceva artificial. Dece 3+4 şi nu 2+5? Prin urmare spre deosebire de egalitate care este simetrică, **rescrierea nu este simetrică**.

Deoarece se doreşte ca rescrierile să fie făcute de calculator, practica programării ne dă un argument foarte puternic împotriva simetriei. Simetria este o regula care conduce la neterminarea programelor, deoarece după ce am rescris a în b putem rescrie pe b în a, pe a în b şi aşa mai departe.

Eliminarea simetriei dintre regulile de deducţie este principala diferenţă dintre calculul cu egalităţi reflectat de logica ecuaţională. Deoarece eliminarea simetriei duce la pierderea completitudinii, simetria va trebui înlocuită cu altceva pentru a reobţine completitudinea.

1 Preliminarii

Signatura Σ, mulţimea Γ de axiome şi algebra $\mathcal{A}$ în care se fac rescrierile(localizare) sunt fixate.

Fie X o mulţime S–sortată de variabile şi $T_\Sigma(X)$ Σ–algebra liber generată de X. Pentru orice $x \in X$ şi $\alpha \in T_\Sigma(X)$, vom nota cu $nr_x(\alpha)$ numărul de apariţii ale lui x în α. Observăm că $nr_x(x) = 1$, $nr_x(y) = 0$ pentru orice $y \in X - \{x\}$ şi $nr_x(\sigma(e_1, e_2, \ldots, e_k)) = nr_x(e_1) + nr_x(e_2) + \ldots nr_x(e_k)$.

Fie $\mathcal{A}$ o Σ–algebră, z o variabilă de sort s, $z \notin A_s$. Considerăm algebra liber generată de $A \cup \{z\}$, şi anume $T_\Sigma(A \cup \{z\})$ pe care o notăm $\mathcal{A}[z]$.

Un element c din $\mathcal{A}[z]$ se numeşte *context* dacă $nr_z(c) = 1$. Dacă $c = \sigma(c_1, c_2, \ldots, c_n)$ este un context, atunci există un $1 \leq i \leq n$ astfel încât c_i este context şi $c_j \in T_\Sigma(A)$ pentru orice $j \neq i$.

Pentru $d \in A_s$, vom nota cu $z \leftarrow d : \mathcal{A}[z] \longrightarrow \mathcal{A}$ unicul morfism de Σ–algebre 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 $(z \leftarrow a)(t)$.

Dacă în t nu apare z, adică t este din Σ-algebra liber generată de A, atunci $t[a] = t[d]$ pentru orice $a, d \in A_s$.

2 Închiderea la contexte

Definiţia 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_s, $a\,Q\,d$ implică $c[a]\,Q\,c[d]$.

Introducem regula de deducţie

CAΣ $a \doteq_{s_i} d$ implică $A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) \doteq_s A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n)$
$(\forall \sigma \in \Sigma_{s_1 \ldots s_n, s}$, orice $1 \leq i \leq n$, unde $a_j \in A_{s_j}$ pentru orice $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$ şi $a, d \in A_{s_i}$.

Definiţia 2.2 O relaţie $\rho \subset A \times A$ se numeşte **compatibilă pe argumente cu operaţiile** algebrei $\mathcal{A}$ dacă este închisă la **CA**Σ.

Propoziţie 2.3 *O relaţie este închisă la contexte dacă şi numai dacă este compatibilă pe argumente cu operaţiile algebrei.*

Demonstraţie: *Presupunem Q închisă la contexte. Pentru a demonstra compatibilitatea pe argumente 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 un context $c = \sigma(a_1, \ldots, a_{i-1}, c', a_{i+1}, \ldots, a_n)$ unde $c' \in \mathcal{A}[z]$ este un context şi $a_i \in T_\Sigma(A)$

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

La fel $c[d] = A_\sigma(a_1[d], \ldots, a_{i-1}[d], c'[d], a_{i+1}[d], \ldots, a_n[d])$. Mai observăm că $a_j[a] = a_j[d]$ pentru orice $j \neq i$.
Din ipoteza de inducţie $c'[a]\, Q\, c'[d]$ şi ţinând cont de compatibilitatea pe argumente obţinem $c[a]\, Q\, c[d]$.

Definiţia 2.4 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 2.5 *$\longrightarrow_Q$ 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 argumente 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 $\longrightarrow_Q \subseteq R$. □

3 Închiderea la preordini compatibile cu operaţiile

Reamintim regulile de deducţie în mulţimea egalităţilor formale între elementele algebrei $\mathcal{A}$ 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 orice $i \in [n]$ implică $A_\sigma(a_1, a_2, \ldots, a_n) \doteq_s A_\sigma(c_1, c_2, \ldots, c_n)$ pentru orice $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$

Remarcăm că **C**Σ şi **R** implică **CA**Σ.
Reciproca **CA**Σ, **R** şi **T** implică **C**Σ este demonstrată de lema următoare.

Lemă 3.1 *Fie $\rho \subset A \times A$ o relaţie tranzitivă şi reflexivă în algebra $\mathcal{A}$. Dacă ρ este compatibilă pe argumente cu operaţiile algebrei $\mathcal{A}$, atunci ρ e compatibilă cu operaţiile, adică închisă la **C**Σ.*

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

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

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 3.2 *$\overset{*}{\longrightarrow}_Q$ este cea mai mică preordine compatibilă cu operaţiile care include Q.*

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

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

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 argumente 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.*

Din proprietăţile operatorilor de închidere şi $R \subseteq Q$ deducem $\longrightarrow_R \subseteq \longrightarrow_Q$ şi $\xrightarrow{*}_R \subseteq \xrightarrow{*}_Q$.

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

4 Γ–rescriere

Un prim pas pentru suplinirea simetriei este înlocuirea substituţiei cu **rescrierea**. Vom mai folosi şi **rescrierea în subtermeni**.

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

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

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

Deasemenea **Rew**$_\Gamma$ şi regula contextului implică **SRew**$_\Gamma$.

Comparând substituţia şi rescrierea observăm următoarele.
1) **Rew**$_\Gamma$ şi **R** implică **Sub**$_\Gamma$.
2) **Sub**$_\Gamma$, **S** şi **T** implică **Rew**$_\Gamma$.

Probăm ultima afirmaţie. Fie $(\forall X)\, l \doteq_s r$ **if** $H \in \Gamma$ şi un morfism $h\colon T_\Sigma(X) \longrightarrow \mathcal{A}$ cu proprietatea că pentru orice $u \doteq_{s'} v \in H$ există $d \in A_{s'}$ cu $h_{s'}(u) \doteq_{s'} d$ şi $h_{s'}(v) \doteq_{s'} d$. Pentru orice $u \doteq_{s'} v \in H$ cu **S** deducem $d \doteq_s h_{s'}(v)$ şi apoi cu **T** obţinem $h_{s'}(u) \doteq_{s'} h_{s'}(v)$. In final aplicăm **Sub**$_\Gamma$ şi deducem $h_s(l) \doteq_s h_s(r)$.

Definim prin inducţie şirul crescător de mulţimi de egalităţi formale din $\mathcal{A}$.

$Q_0 = \emptyset$

$Q_{n+1} = \{h_s(l) \doteq_s h_s(r) \ :\ (\forall Y) l \doteq_s r$ **if** $H \in \Gamma,\ h : T_\Sigma(Y) \longrightarrow \mathcal{A},$ şi $(\forall u \doteq_t v \in H) h_t(u) \downarrow_{Q_n} h_t(v)\}$

Pentru a ne convinge că şirul de mai sus este crescător putem demonstra prin inducţie că $(\forall n) Q_n \subseteq Q_{n+1}$.

Deoarece $\longrightarrow_\bullet$ şi $\xrightarrow{*}_\bullet$ sunt operatori de închidere deducem că şirurile $\longrightarrow_{Q_n}$ şi $\xrightarrow{*}_{Q_n}$ sunt crescatoare.
Prin definiţie Q este reuniunea şirului crescător definit mai sus

$$Q = \bigcup_{n \in N} Q_n.$$

Remarcăm că $a \longrightarrow_Q d$ implică existenţa unui n natural cu proprietatea $a \longrightarrow_{Q_n} d$. Deasemenea $a \xrightarrow{*}_Q d$ implică existenţa unui n natural cu proprietatea $a \xrightarrow{*}_{Q_n} d$.

Şirul $\downarrow_{Q_n}$ este crescător. În plus $a \downarrow_Q b$ implică existenţa unui n natural cu proprietatea $a \downarrow_{Q_n} b$.

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

Demonstraţie: Fie $(\forall Y) l \doteq_s r$ **if** $H \in \Gamma$, $h : T_\Sigma(Y) \longrightarrow \mathcal{A}$ morfism cu proprietatea $h_t(u) \downarrow_Q h_t(v)$ pentru orice $u \doteq_t v \in H$ 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 paşilor utilizat în $h_t(u) \downarrow_Q h_t(v)$ unde $u \doteq_t v \in H$ este finit există un n natural astfel încât $h_t(u) \downarrow_{Q_n} h_t(v)$ pentru orice $u \doteq_t v \in H$. Prin urmare $h_s(l) \doteq_s h_s(r) \in Q_{n+1}$. Deoarece $h_s(l) \doteq_s h_s(r) \in Q$ deducem $c[h_s(l)] \stackrel{*}{\Longrightarrow}_\Gamma c[h_s(r)]$. □

Propoziţie 4.2 *$\stackrel{*}{\Longrightarrow}_\Gamma$ este cea mai mică relaţie închisă la* $\mathbf{R}$, $\mathbf{T}$, $\mathbf{C}\Sigma$ *şi* $\mathbf{Rew}_\Gamma$.

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

Fie W o relaţie închisă la $\mathbf{R}$, $\mathbf{T}$, $\mathbf{C}\Sigma$ *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 $h_s(l) \doteq_s h_s(r) \in Q_n$, unde $(\forall Y) l \doteq_s r$ **if** *$H \in \Gamma$, $h : T_\Sigma(Y) \longrightarrow \mathcal{A}$ şi $h_t(u) \downarrow_{Q_{n-1}} h_t(v)$ pentru orice $u \doteq_t v \in H$. 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* $\mathbf{R}$ *şi* $\mathbf{T}$ *rezultă că $\stackrel{*}{\longrightarrow}_{Q_{n-1}} \subseteq W$.*
Din $(\forall u \doteq_t v \in H) h_t(u) \downarrow_{Q_{n-1}} h_t(v)$ obţinem $(\forall u \doteq_t v \in H) h_t(u) \downarrow_W h_t(v)$. Folosind închiderea lui W la $\mathbf{Rew}_\Gamma$ *rezultă că $h_s(l) \doteq_s h_s(r) \in W$, deci $Q_n \subseteq W$.*

Prin urmare $Q \subseteq W$ şi folosind propoziţia 3.2 deducem că $\stackrel{*}{\longrightarrow}_Q \subseteq W$.

5 Corectitudinea rescrierii

Teorema 5.1 *Toate regulile de deducţie de mai sus sunt corecte pentru congruenţa semantică $\equiv_\Gamma$.*

Demonstraţie: Din lecţia privind logica ecuaţională ştim că regulile de deducţie $\mathbf{R}$, $\mathbf{S}$, $\mathbf{T}$, $\mathbf{C}\Sigma$ şi $\mathbf{Sub}_\Gamma$ sunt corecte.

Deoarece $\mathbf{C}\Sigma$ şi $\mathbf{R}$ implică $\mathbf{CA}\Sigma$ deducem că $\mathbf{CA}\Sigma$ este corectă.

Deoarece $\mathbf{Sub}_\Gamma$, $\mathbf{S}$ şi $\mathbf{T}$ implică $\mathbf{Rew}_\Gamma$ rezultă că $\mathbf{Rew}_\Gamma$ este corectă.

Deoarece $\mathbf{Rew}_\Gamma$ şi $\mathbf{CA}\Sigma$ implică $\mathbf{SRew}_\Gamma$ deducem că $\mathbf{SRew}_\Gamma$ este corectă. □

Corolar 5.2 $\stackrel{*}{\Longrightarrow} \subseteq \equiv_\Gamma$. *(Corectitudinea rescrierii pentru congruenţa semantică)*

Rescrierea nu este completă pentru congruenţa semantică deoarece $\mathbf{Rew}_\Gamma$ nu captează întreaga forţa a simetriei.

4 Relaţia de întâlnire, Forme normale

VEC

May 26, 2008

1 Confluenţă

Definiţia 1.1 O relaţie $\succ$ pe o mulţime A se numeşte **confluentă** dacă

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

Propoziţie 1.2 *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 $a \equiv b$. $\square$

Observaţia 1.3 *Dacă $\succ$ este o preordine confluentă şi compatibilă pe o Σ-algebră multisortată, atunci $\downarrow$ este o congruenţă.*

Demonstraţie: Fie $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ 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

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

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

2 Completitudinea întâlnirii prin recriere

Prin definiţie, pentru orice $s \in S$ şi orice $a, d \in A_s$:

$$a \downarrow_\Gamma d \text{ dacă şi numai dacă există } c \in A_s \text{ astfel încât } a \stackrel{*}{\Rightarrow}_\Gamma c \text{ şi } d \stackrel{*}{\Rightarrow}_\Gamma c.$$

Propoziţie 2.1 $\downarrow_\Gamma \subseteq \equiv_\Gamma$.

Demonstraţie: Din $a \downarrow_\Gamma d$ deducem existenţa lui c astfel încât: $a \stackrel{*}{\Rightarrow}_\Gamma c$ şi $d \stackrel{*}{\Rightarrow}_\Gamma c$. Din aceasta deducem că $a \equiv_\Gamma c$ şi $d \equiv_\Gamma c$ deci $a \equiv_\Gamma d$. $\square$

Aceasta dovedeşte corectitudinea lui $\downarrow_\Gamma$. Pentru a demonstra şi completitudinea sa, adică incluziunea inversă, avem nevoie să presupunem că $\stackrel{*}{\Rightarrow}_\Gamma$ este confluentă. Folosind observaţia 1.3 deducem că $\downarrow_\Gamma$ este o congruenţă.

Lemă 2.2 *Dacă $\stackrel{*}{\Rightarrow}_\Gamma$ este confluentă, atunci congruenţa $\downarrow_\Gamma$ este închisă la substituţie.*

Demonstraţie: *Fie $(\forall X)\, l =_s r$ **if** $H \in \Gamma$ şi fie $h\colon T_\Sigma(X) \longrightarrow \mathcal{A}$ un morfism pentru care $h_t(u) \downarrow_\Gamma h_t(v)$ pentru orice $u \doteq_t v \in H$. Pentru orice $u \doteq_t v \in H$ există $a_{uv} \in A_t$ cu proprietăţile $h_t(u) \stackrel{*}{\Rightarrow}_\Gamma a_{uv}$ şi $h_t(v) \stackrel{*}{\Rightarrow}_\Gamma a_{uv}$. Deoarece $\stackrel{*}{\Rightarrow}_\Gamma$ este închisă la $\mathbf{Rew_\Gamma}$ deducem că $h_s(l) \stackrel{*}{\Rightarrow}_\Gamma h_s(r)$. Prin urmare $h_s(l) \downarrow_\Gamma h_s(r)$.* □

Fie $\mathcal{A}^\Gamma$ factorizarea lui $\mathcal{A}$ prin $\downarrow_\Gamma$ şi fie $\tau\colon \mathcal{A} \longrightarrow \mathcal{A}^\Gamma$ morfismul de factorizare. Doarece congruenţa $\downarrow_\Gamma$ este închisă la substituţie deducem că $\mathcal{A}^\Gamma \models_\Gamma$.

Teorema 2.3 *Dacă $\stackrel{*}{\Rightarrow}_\Gamma$ este confluentă, atunci $\downarrow_\Gamma = \equiv_\Gamma$.*

Demonstraţie: Doarece congruenţa $\downarrow_\Gamma$ este închisă la substituţie deducem că $\mathcal{A}^\Gamma \models_\Gamma$.

Folosind propoziţia 2.1 avem de demonstrat doar $\equiv_\Gamma \subseteq \downarrow_\Gamma$.

Fie $a \equiv_\Gamma c$. Deoarece $\mathcal{A}^\Gamma \models \Gamma$ şi $\tau\colon \mathcal{A} \longrightarrow \mathcal{A}^\Gamma$ este morfism rezultă că $\tau_s(a) = \tau_s(c)$ adică $a \downarrow_\Gamma c$.

În concluzie $\downarrow_\Gamma = \equiv_\Gamma$ ceea ce demonstrează completitudinea lui $\downarrow_\Gamma$. □

Din punct de vedere practic această teoremă arată că demonstrarea propoziţiei $a \equiv_\Gamma c$ poate fi redusă la a demonstra $a \downarrow_\Gamma c$ dacă rescrierea este confluentă. Prin urmare principala problemă devine

"poate o maşină să demonstreze că $a \downarrow_\Gamma c$?"

3 Forme normale

3.1 La mulţimi

Presupunem în continuare că $\succ$ este o preordine pe mulţimea A.

Definiţia 3.1 Elementul $n \in A$ se numeşte **o formă normală** pentru relaţia $\succ$ pe mulţimea A dacă

$$(\forall b \in A)(n \succ b \Rightarrow n = b).$$

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

$$\mathbf{FN!}\ \ (\forall a \in A)(\exists!\, fn(a) \in N) a \succ fn(a).$$

Observaţia 3.2 *Dacă $a \succ d$, atunci $fn(a) = fn(d)$.*

Demonstraţie: Din ipoteză şi $d \succ fn(d)$ deducem $a \succ fn(d)$, deci din unicitatea formei normale a lui a deducem $fn(a) = fn(d)$.

Observaţia 3.3 *Axioma* **FN!** *implică $\succ$ este confluentă.*

Demonstraţie: Presupunem $a \succ d$ şi $a \succ c$. Deducem $fn(a) = fn(d) = fn(c)$, deci $d \succ fn(d)$ şi $c \succ fn(d)$.

Observaţia 3.4 *Funcţia $fn\colon A \longrightarrow N$ este surjectivă şi*

$$a \downarrow d \Leftrightarrow fn(a) = fn(d).$$

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

Presupunem $fn(a) = fn(d)$. Deoarece $a \succ fn(a)$ şi $d \succ fn(a)$ deducem $a \downarrow d$.

Presupunem $a \downarrow d$. Fie $c \in A$ astfel încât $a \succ c$ şi $d \succ c$. Deducem $fn(a) = fn(c)$ şi $fn(d) = fn(c)$, deci $fn(a) = fn(d)$. □

Din punct de vedere practic această observaţie arată că dacă rescrierea are proprietatea **FN!** *demonstrarea lui $a \downarrow_\Gamma d$ este echivalentă cu egalitatea formelor normale pentru rescriere ale lui a şi d.*

Remarcăm că N este un sistem complet şi independent de reprezentanţi pentru $\downarrow$.

Observaţia 3.5 *Există o unică bijecţie $B\colon N \longrightarrow A/_\downarrow$ astfel încât compunerea $fn; B$ coincide cu funcţia naturală de factorizare.*

3.2 La Algebre

În Σ-algebra $\mathcal{A} = (A_s, A_\sigma)$ presupunem că $\succ$ este o preordine multi-sortată compatibilă cu proprietatea **FN!**. Σ-algebra $\mathcal{N} = (N_s, N_\sigma)$ a formelor normale este definiă prin:

$N_s = \{a \in A_s : a \text{ este formă normală } \}$ pentru $s \in S$ şi

$N_\sigma(n_1, n_2, \ldots, n_k) = fn(A_\sigma(n_1, n_2, \ldots, n_k))$ pentru orice $\sigma \in \Sigma_{s_1 s_2 \ldots s_k, s}$ şi $n_i \in N_{s_i}$ pentru $1 \leq i \leq k$.

Observaţia 3.6 *$fn : \mathcal{A} \longrightarrow \mathcal{N}$ este un Σ-morfism.*

Demonstraţie: Fie $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ şi $a_i \in A_{s_i}$. Deoarece $a_i \succ fn_{s_i}(a_i)$ pentru orice $i \in [n]$ deducem că

$$A_\sigma(a_1, a_2, \ldots, a_n) \succ A_\sigma(fn_{s_1}(a_1), fn_{s_2}(a_2), \ldots, fn_{s_n}(a_n)),$$

prin urmare

$fn_s(A_\sigma(a_1, a_2, \ldots, a_n)) = fn_s(A_\sigma(fn_{s_1}(a_1), fn_{s_2}(a_2), \ldots, fn_{s_n}(a_n))) = N_\sigma(fn_{s_1}(a_1), fn_{s_2}(a_2), \ldots, fn_{s_n}(a_n))$. □

Propoziţie 3.7 *Dacă în Σ-algebra $\mathcal{A} = (A_s, A_\sigma)$, $\succ$ este o preordine multi-sortată compatibilă cu proprietatea* **FN!**, *atunci algebrele $\mathcal{A}/_\downarrow$ şi $\mathcal{N}$ sunt izomorfe.*

Demonstraţie: Deoarece fn este surjectiv pe componente şi $Ker(fn) = \downarrow$ rezultă că $\mathcal{A}/_\downarrow$ şi $\mathcal{N}$ sunt izomorfe.

Teorema 3.8 *Dacă $\overset{*}{\Rightarrow}_\Gamma$ satisface axioma* **FN!**, *atunci algebra formelor normale este isomorfă cu $\mathcal{A}^\Gamma$.*

4 Relaţii canonice

In practică pentru a verifica că $\overset{*}{\Rightarrow}_\Gamma$ satisface axioma **FN!**, preferăm uneori să verificăm că satisface proprietăţile de confluenţă şi terminare.

Definiţia 4.1 Spunem că $\succ$ are proprietatea de **terminare** dacă nu există şiruri $\{a_n\}$ de elemente din A astfel încât pentru orice număr natural n să avem $a_n \succ a_{n+1}$ şi $a_n \neq a_{n+1}$. □

Observaţia 4.2 *Dacă $\succ$ are proprietatea de terminare, atunci pentru orice element a din A există un element în formă normală $n \in N$ cu proprietatea $a \succ n$.*

Demonstraţie: *Raţionând prin absurd, a nu este în forma normală; prin urmare există $a_1 \in A$ cu proprietăţile $a \succ a_1$ şi $a \neq a_1$. Raţionând în continuare prin absurd, a_1 nu este în forma normală; prin urmare există $a_2 \in A$ cu proprietăţile $a_1 \succ a_2$ şi $a_1 \neq a_2$. Continuând acest raţionament prin inducţie putem construi un şir de elemente care contrazice proprietatea de terminare.* □

Dacă preordinea $\succ$ este confluentă şi are proprietatea de terminare spunem că $< A, \succ >$ este **canonică**. □

Propoziţie 4.3 *Dacă $< A, \succ >$ este canonică atunci axioma* **FN!** *este verificată.*

Demonstraţie: Unicitatea va rezulta din confluenţă. Presupunând că n' şi n" sunt două elemente în formă normală cu proprietatea $a \succ n'$ şi $a \succ n$" din confluenţă există $d \in A$ astfel încât $n' \succ d$ şi n" $\succ d$. Deoarece n' şi n" în sunt formă normală deducem că $n' = d$ şi n" $= d$, deci $n' = n$". □

Din punct de vedere practic terminarea este necesară pentru a opri execuţia calculatorului iar confluenţa este necesară pentru completitudine. Aceste proprietăţi implică **FN!**. *Pentru a demonstra că $a \equiv_\Gamma c$ este suficient să rescriem a si c in forma lor normală $fn(a)$ şi $fn(c)$ pentru ca apoi să verificăm egalitatea $fn(a) = fn(c)$.*

5 RESCRIERE ÎN SUBTERMENI

Virgil Emil Căzănescu

February 14, 2010

Am văzut că rescrierea canonică dă o metodă de demonstrare automată a propoziţiilor valide. Următoarea problemă constă în a arăta cum regulile de deducţie pentru rescriere pot fi utilizate în mod practic.

Teoremele din această lecţie explică modul în care se fac rescrierile în limbajele de programare declarativă.

1 Completitudinea rescrierii în subtermeni

Teorema precedentă ne asigură că rescrierea este cea mai mică preordine închisă la $\mathbf{SRew}_\Gamma$ compatibilă cu toate operaţiile din $\mathcal{A}$. Teorema următoare ne spune ceva mai mult.

Teorema 1.1 *Rescrierea este cea mai mică preordine din A închisă la $\mathbf{SRew}_\Gamma$.*

Demonstraţie: Fie R_Γ cea mai mică relaţie pe A care este închisă la $\mathbf{R}$, $\mathbf{T}$ şi $\mathbf{SRew}_\Gamma$. Din definiţie deducem că $R_\Gamma \subseteq \stackrel{*}{\Rightarrow}_\Gamma$.

Pentru a demonstra incluziunea contrară este suficient să arătăm că R_Γ este închisă în $\mathbf{C}\Sigma$.

Dar R_Γ este tranzitivă şi reflexivă, deci este suficient să arătăm pentru fiecare $\sigma \in \Sigma_{s_1\dots s_n,s'}$ că $a\, R_\Gamma\, d$ implică

$$A_\sigma(a_1,\dots,a_{i-1},a,a_{i+1},\dots,a_n)\, R_\Gamma\, A_\sigma(a_1,\dots,a_{i-1},d,a_{i+1},\dots,a_n).$$

Demonstrăm că mulţimea

$$D = \{a\, R_\Gamma\, d \mid A_\sigma(a_1,\dots,a_{i-1},d,a_{i+1},\dots,a_n)\, R_\Gamma\, A_\sigma(a_1,\dots,a_{i-1},a,a_{i+1},\dots,a_n)\}$$

este închisă la $\mathbf{R}$, $\mathbf{T}$ şi $\mathbf{SRew}_\Gamma$.

Reflexivitatea lui D rezultă din reflexivitatea lui R_Γ.

Presupunem că (a,u) şi (u,d) sunt în D. Folosind tranzitivitatea lui R_Γ din $a\, R_\Gamma\, u$ şi $u\, R_\Gamma\, d$ deducem $a\, R_\Gamma\, d$ şi din $A_\sigma(a_1,\dots,a_{i-1},a,a_{i+1},\dots,a_n)\ R_\Gamma\ A_\sigma(a_1,\dots,a_{i-1},u,a_{i+1},\dots,a_n)$ şi
$A_\sigma(a_1,\dots,a_{i-1},u,a_{i+1},\dots,a_n)\ R_\Gamma\ A_\sigma(a_1,\dots,a_{i-1},d,a_{i+1},\dots,a_n)$ deducem
$A_\sigma(a_1,\dots,a_{i_1},a,a_{i+1},\dots,a_n)\, R_\Gamma\, A_\sigma(a_1,\dots,a_{i-1},d,a_{i+1},\dots,a_n,d)$.

Probăm că D este închisă la $\mathbf{SRew}_\Gamma$. Fie $(\forall X) l \doteq_s r$ **if** $H \in \Gamma$, un morfism $h : T_\Sigma(X) \to \mathcal{A}$ cu proprietatea că pentru orice $u \doteq_t v \in H$. există $a_{uv} \in A_t$ astfel încât $(h_t(u), a_{uv}) \in D$ şi $(h_t(v), a_{uv}) \in D$ Pentru orice context $c \in \mathcal{A}[z]_{s'}$ trebuie arătat că $(c[h_s(l)], c[h_s(r)]) \in D$.

Deoarece R_Γ este închisă la $\mathbf{SRew}_\Gamma$ din $h_t(u)\, R_\Gamma\, a_{uv}$ şi $h_t(v)\, R_\Gamma\, a_{uv}$ pentru orice $u \doteq_t v \in H$ deducem

$$c[h_s(l)]\, R_\Gamma\, c[h_s(r)] \qquad \text{şi} \qquad c'[h_s(l)]\, R_\Gamma\, c'[h_s(r)]$$

unde $c' = \sigma(a_1,\dots,a_{i-1},c,a_{i+1},\dots,a_n)$ este un context.
Din $c'[h_s(l)] = A_\sigma(a_1,\dots,a_{i-1},c[h_s(l)],a_{i+1}\dots,a_n)$ şi $c'[h_s(r)] = A_\sigma(a_1,\dots,a_{i-1},c[h_s(r)],a_{i+1},\dots,a_n)$ obţinem

$$A_\sigma(a_1,\dots,a_{i-1},c[h_s(l)],a_{i+1},\dots,a_n)\, R_\Gamma\, A_\sigma(a_1,\dots,a_{i-1},c[h_s(r)],a_{i+1},\dots,a_n).$$

Deci $(c[h_s(l)], c[h_s(r)]) \in D$.

În concluzie $R_\Gamma = \stackrel{*}{\Rightarrow}_\Gamma$. □

2 Rescriere într-un pas

Fie c în $\mathcal{A}[z]_{s'}$ un context. Fie $(\forall X) l \doteq_s r$ **if** $H \in \Gamma$ şi fie $h : T_\Sigma(X) \to \mathcal{A}$ un morfism astfel încât pentru orice $u \doteq_t v \in H$ există $a_{uv} \in A_t$ cu proprietăţile $h_t(u) \stackrel{*}{\Rightarrow}_\Gamma a_{uv}$ şi $h_t(v) \stackrel{*}{\Rightarrow}_\Gamma a_{uv}$. Dacă $c[h_s(l)] \neq c[h_s(r)]$ spunem că $c[h_s(l)]$ se rescrie într-un pas în $c[h_s(r)]$ şi scriem

$$c[h_s(l)] \Rightarrow c[h_s(r)].$$

Propoziţie 2.1 *Rescrierea este închiderea reflexiv-tranzitivă a rescrierii într-un pas.*

Demonstraţie: Fie $\stackrel{*}{\Rightarrow}$ închiderea reflexiv-tranzitivă a rescrierii într-un pas.

Deoarece $\Rightarrow \subseteq \stackrel{*}{\Rightarrow}_\Gamma$ şi $\stackrel{*}{\Rightarrow}_\Gamma$ este preordine deducem incluziunea $\stackrel{*}{\Rightarrow} \subseteq \stackrel{*}{\Rightarrow}_\Gamma$.

Reciproc, deoarece $\stackrel{*}{\Rightarrow}$ este reflexivă şi tranzitivă este suficient să arătăm că $\stackrel{*}{\Rightarrow}$ este închisă la $\mathbf{SRew}_\Gamma$.
Fie $(\forall X)\, l \doteq_s r$ **if** $H \in \Gamma$, $h : T_\Sigma(X) \to \mathcal{A}$ un morfism astfel încât pentru orice $u \doteq_t v \in H$ există $a_{uv} \in A_t$ cu proprietăţile $h_t(u) \stackrel{*}{\Rightarrow} a_{uv}$ şi $h_t(v) \stackrel{*}{\Rightarrow} a_{uv}$ şi c un context din $\mathcal{A}[z]$. Din incluziunea deja demonstrată rezultă că pentru orice $u \doteq_t v \in H$ există $a_{uv} \in A_t$ cu proprietăţile $h_t(u) \stackrel{*}{\Rightarrow}_\Gamma a_{uv}$ şi $h_t(v) \stackrel{*}{\Rightarrow}_\Gamma a_{uv}$. Avem cazurile:

1. Dacă $c[h(l)] = c[h(r)]$ deducem din reflexivitatea lui $\stackrel{*}{\Rightarrow}$ că $c[h(l)] \stackrel{*}{\Rightarrow} c[h(r)]$.
2. Altfel, dacă $c[h(l)] \neq c[h(r)]$ atunci $c[h(l)] \stackrel{*}{\Rightarrow} c[h(r)]$, deoarece rescrierea într-un pas este inclusă în închiderea reflexivă şi tranzitivă a rescrierii într-un pas. □

În continuare rescrierea într-un pas va fi notată cu $\Rightarrow_\Gamma$ sau $\Rightarrow$.

3 Consideraţii metodologice

Pentru a demonstra că $a \equiv_\Gamma b$ este suficient să arătăm că $fn_s(a) = fn_s(b)$ şi aceasta se face prin rescrierea lui a, respectiv a lui b în formele lor normale, pentru a verifica că sunt egale.

Fie $a \in A_s$ elementul de rescris. Se caută

$$(\forall X) l \doteq_s r \textbf{ if } H \text{ în } \Gamma,\ h\ :\ T_\Sigma(X) \longrightarrow \mathcal{A} \text{ şi un context c in } \mathcal{A}[z] \text{ astfel încât } c[h_s(l)] = a.$$

Pentru a face rescrierea este suficient să arătăm că $h_t(u) \downarrow_\Gamma h_t(v)$ pentru orice $u \doteq_t v \in H$. Se întrerupe rescrierea principală pentru a face acestă verificare şi în caz de reuşită se rescrie a în $c[h_s(r)]$. In caz de nereuşită se continuă căutarile pentru a reajunge în situaţia de mai sus. Intrucât în timpul verificării uneia dintre condiţiile $h_t(u) \downarrow_\Gamma h_t(v)$ se rescriu atât $h_t(u)$ cât şi $h_t(v)$ în forma lor normală, fenomenul de mai sus se poate repeta se apelează la mecanismul numit backtracking.

Dacă Γ conţine doar ecuaţii necondiţionate, rescrierea într-un singur pas nu depinde de rescriere, ca în cazul general. Acest fapt explică de ce, în acest caz metoda backtracking nu este folosită în implementarea rescrierii.

Exemplu Signatura are două sorturi *nat* şi *bool*; trei constante *true*, $false$ de sort *bool* şi 0 de sort *nat*; două simboluri de operaţii $s : nat \longrightarrow nat$ şi $\leq$: $nat\ nat \longrightarrow bool$. Axiomele acceptate cu x şi y variabile de sort *nat* sunt

1. $0 \leq x = true$ 2. $s(x) \leq s(y) = true$ **if** $x \leq y = true$ 3. $s(x) \leq 0 = false$

Vrem să rescriem $s(s(0)) \leq s(s(z))$. Se poate aplica numai axioma 2 pentru $x = s(0)$ şi $y = s(z)$.

Înainte de aceasta trebuie să rescriem $s(0) \leq s(z)$. Se poate aplica numai axioma 2 pentru $x = 0$ şi $y = z$.

Înainte de aceasta rescriem $0 \leq z \Rightarrow_\Gamma true$ cu axioma 1.

Deoarece am obţinut *true* putem rescrie $s(0) \leq s(z) \Rightarrow_\Gamma true$.
Deoarece prin rescrierea lui $s(0) \leq s(z)$ am obţinut *true* putem rescrie $s(s(0)) \leq s(s(z)) \Rightarrow_\Gamma true$.

6 UNIFICARE, LOCAL CONFLUENŢĂ

Virgil Emil Căzănescu

May 26, 2008

1 Unificare

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 $l_i =_{s_i} r_i$ şi se cere găsirea unui unificator, adică a unei substituţii u cu proprietatea $u(l_i) = u(r_i)$ pentru orice i.

1.1 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 $t = t$.
2. **Descompune.** Orice ecuaţie din lista de rezolvat de forma
$$f(a_1, a_2, \ldots, a_n) = f(b_1, b_2, \ldots, b_n)$$
se înlocuieşte cu ecuaţiile $a_1 = b_1$, $a_2 = b_2$, ..., $a_n = b_n$,
3. **Elimină.** Orice ecuaţie din lista de rezolvat, de forma $x = t$ sau $t = x$ în care x este o variabilă care nu apare în t este mutată sub forma $x = t$ 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 $f(a_1, a_2, \ldots, a_n) = g(b_1, b_2, \ldots, b_m)$ cu $f \neq g$,
2) Existenţa în lista de rezolvat a unei ecuaţii de forma $x = t$ sau $t = x$ 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.

1.2 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 aparţiilor simbolurilor(semnelor) care apar în lista de rezolvat, care se micşorează în cele două cazuri care ne mai interesează **Scoate** şi **Descompune**.

1.3 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 egaliatea

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

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

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

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 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 $x = t$ 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 $x_i = t_i$ 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 $U; U = U$.

In semantica operaţională a limbajelor de specificaţie rescrierea joacă un rol primordial. Pentru a rescrie un element a cu ajutorul unei reguli $l =_s r$ 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.

2 Local confluenţă

Reamintim că relaţia rescrierea este confluentă dacă

$$(\forall a, b, c)\ [(a \overset{*}{\Longrightarrow} b \text{ şi } a \overset{*}{\Longrightarrow} c) \text{ implică } (\exists d)\ (b \overset{*}{\Longrightarrow} d \text{ şi } c \overset{*}{\Longrightarrow} d)].$$

Definiţia 2.1 Spunem că rescrierea este **local confluentă** dacă

$$(\forall a, b, c)\ [(a \Longrightarrow b \text{ şi } a \Longrightarrow c) \text{ implică } (\exists d)\ (b \overset{*}{\Longrightarrow} d \text{ şi } c \overset{*}{\Longrightarrow} d)].$$

Definiţia 2.2 Se spune că rescrierea are proprietatea de terminare, dacă nu există şiruri $\{a_n\}_n$ cu proprietatea $a_n \Longrightarrow a_{n+1}$ pentru orice n număr natural.

Propoziţie 2.3 *Dacă rescrierea este local confluentă şi are proprietatea de terminare atunci rescrierea este confluentă.*

Demonstraţie: Reamintim că proprietatea de terminare ne aisgură pentru orice a existenţa unei forme normale n cu proprietatea $a \overset{*}{\Longrightarrow} n$. Vom demonstra unicitatea formei normale fapt ce implică confluenţa. Notăm cu N mulţimea formelor normale. Fie

$$M = \{a : (\exists n_1, n_2 \in N) a \overset{*}{\Longrightarrow} n_1, a \overset{*}{\Longrightarrow} n_2, n_1 \neq n_2\}.$$

Probăm că $a \in M$ implică $(\exists b \in M) a \Longrightarrow b$.

Deoarece $a \in M$ există $n_1, n_2 \in N$ cu proprietăţile $a \overset{*}{\Longrightarrow} n_1$, $a \overset{*}{\Longrightarrow} n_2$ şi $n_1 \neq n_2$. Dacă a ar fi o formă normală, atunci $a = n_1$ şi $a = n_2$ ceea ce contrazice $n_1 \neq n_2$.

Deoarece a nu este formă normală există elemntul b cu proprietăţile $a \Longrightarrow b$ şi $b \overset{*}{\Longrightarrow} n_1$. La fel există c cu proprietăţile $a \Longrightarrow c$ şi $c \overset{*}{\Longrightarrow} n_2$. Deoarece rescrierea este local confluenţa există elementul d cu proprietăţile $b \overset{*}{\Longrightarrow} d$ şi $c \overset{*}{\Longrightarrow} d$. Deoarece terminarea implică existenţa formei normale deducem $\exists n \in N$ cu proprietatea $d \overset{*}{\Longrightarrow} n$.

Deoarece $n_1 \neq n_2$ deducem $n \neq n_1$ sau $n \neq n_2$.

Dacă $n \neq n_1$ deoarece $b \overset{*}{\Longrightarrow} n_1$ şi $b \overset{*}{\Longrightarrow} n$ decucem $b \in M$. Dacă $n \neq n_2$ deoarece $c \overset{*}{\Longrightarrow} n$ şi $c \overset{*}{\Longrightarrow} n_2$ deducem $c \in M$.

Probăm unicitatea formei normale. Raţionând prin absurd deducem că M este nevidă. Plecând de la $a_1 \in M$ aplicând observaţia de mai sus deducem existenţa lui $a_2 \in M$ cu proprietatea $a_1 \Longrightarrow a_2$. Repetând raţionamentul printr-o inducţie deducem existenţa unui şir $\{a_n\}_n$ cu proprietatea $a_n \Longrightarrow a_{n+1}$ pentru orice n natural. Deci rescrierea nu are proprietatea de terminare, în contradicţie cu ipoteza. □

7 RESCRIERE MODULO ECUATII

Virgil Emil Căzănescu

February 14, 2010

Motivare : Comutativitatea folosită ca regulă de rescriere conduce la pierderea proprietăţii de terminare.

Deoarece unele ecuaţii, de exemplu comutativitatea, nu pot fi folosite ca axiome pentru rescriere, s-a găsit o altă cale pentru utilizarea acestora şi anume rescrierea modulo ecuaţii. Mulţimea E reprezintă ecuaţiile care în loc de a fi folosite ca axiome vor fi utilizate pentru rescrierea modulo ecuaţii.

1 Motivare semantică

Fie E o mulţime de Σ–ecuaţii şi Γ o mulţime de Σ–ecuaţii condiţionate.

Se ştie că în categoria Σ-algebrelor monomorfismele coincid cu morfismele care au toate componentele injective.

Propoziţie 1.1 *Fie $m : \mathcal{A} \longrightarrow \mathcal{B}$ un monomorfism din categoria Σ–algebrelor. Dacă $\mathcal{B} \models \Gamma$, atunci $\mathcal{A} \models \Gamma$.*

Demonstraţie: Fie $(\forall X) l \doteq_s r$ **if** $H \in \Gamma$ şi $h : T_\Sigma(X) \longrightarrow \mathcal{A}$ un morfism astfel încât $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$. Atunci $h; m : T_\Sigma(X) \longrightarrow \mathcal{B}$ este un morfism şi $(h;m)_t(u) = (h;m)_t(v)$. Deoarece $\mathcal{B}$ satisface Γ rezultă că $(h;m)_s(l) = (h;m)_s(r)$, adică $m_s(h_s(l)) = m_s(h_s(r))$. Dar m este monomorfism deci $h_s(l) = h_s(r)$. Aşadar $\mathcal{A}$ satisface orice $(\forall X) l \doteq_s r$ **if** $H \in \Gamma$, adică $\mathcal{A} \models \Gamma$. □

Se ştie că în categoria Σ-algebrelor epimorfismele coincid cu morfismele care au toate componentele surjective.

Propoziţie 1.2 *Fie $e : \mathcal{A} \longrightarrow \mathcal{B}$ un epimorfism din categoria Σ–algebrelor. Dacă $\mathcal{A} \models E$, atunci $\mathcal{B} \models E$.*

Demonstraţie: Fie $(\forall X) l \doteq_s r \in E$ şi $h : T_\Sigma(X) \longrightarrow \mathcal{B}$ un morfism. Deoarece Σ-algebra liberă $T_\Sigma(X)$ este *proiectivă* rezultă că există un morfism $g : T_\Sigma(X) \longrightarrow \mathcal{A}$ astfel încât $g; e = h$. Atunci

$$\mathcal{A} \models (\forall X) l \doteq_s r \in E \Rightarrow g_s(l) = g_s(r) \Rightarrow e_s(g_s(l)) = e_s(g_s(r)) \Rightarrow h_s(l) = h_s(r).$$

Deci $\mathcal{B}$ satisface orice ecuaţie $(\forall X) l \doteq_s r$ din E, adică $\mathcal{B}$ satisface E. □

Reamintim că $\mathcal{A}_\Gamma = \mathcal{A}/_{\equiv_\Gamma}$. Pentru a,c $\in A_s$ reamintim deasemenea că $a \equiv_\Gamma c$ dacă şi numai dacă $h_s(a) = h_s(c)$ pentru orice $h : \mathcal{A} \longrightarrow \mathcal{B} \models \Gamma$.

Notăm $\mathcal{A}_{E,\Gamma} = (\mathcal{A}_E)_\Gamma$, iar cu $\rho : \mathcal{A} \longrightarrow \mathcal{A}_E$ şi $j : \mathcal{A}_E \longrightarrow \mathcal{A}_{E,\Gamma}$ morfismele canonice de factorizare.

Observaţia 1.3 *Algebrele $\mathcal{A}_{\Gamma\cup E}$ şi $\mathcal{A}_{E,\Gamma}$ sunt izomorfe.*

Demonstraţie: Ideea demonstraţiei este de a arăta că $\mathcal{A}_{E,\Gamma}$ are proprietăţile care caracterizează abstracţie de un izomorfism Σ–algebra $\mathcal{A}_{\Gamma\cup E}$.

Evident $\mathcal{A}_{E,\Gamma} \models \Gamma$. Deoarece $j : \mathcal{A}_E \longrightarrow \mathcal{A}_{E,\Gamma}$ este epimorfism şi $\mathcal{A}_E \models E$ rezultă din propoziţia 1.2 că $\mathcal{A}_{E,\Gamma} \models E$, deci $\mathcal{A}_{E,\Gamma} \models \Gamma \cup E$.

Pentru restul demonstraţiei puteţi utiliza figura 1.

Fie $\mathcal{B}$ o Σ–algebră cu proprietatea $\mathcal{B} \models \Gamma \cup E$. Deoarece $\mathcal{B} \models E$, din proprietatea de universalitate a algebrei $\mathcal{A}_E$ deducem că există un unic morfism $h' : \mathcal{A}_E \longrightarrow \mathcal{B}$ astfel încât $\rho; h' = h$. Deoarece $\mathcal{B} \models \Gamma$ rezultă că există un unic morfism $h'' : \mathcal{A}_{E,\Gamma} \longrightarrow \mathcal{B}$ astfel încât $j; h'' = h'$. Atunci $(\rho; j); h'' = h$.

Unicitatea lui h'' rezultă din surjectivitatea lui $\rho; j$. □

Propoziţie 1.4 $a \equiv_{\Gamma\cup E} c \Leftrightarrow \rho_s(a) \equiv_\Gamma \rho_s(c)$ *pentru oricare $a, c \in A_s$.*

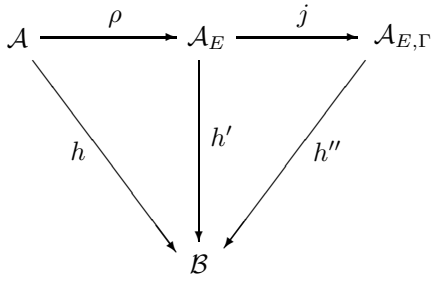


Figure 1:

Demonstraţie: Fie $t : \mathcal{A} \longrightarrow \mathcal{A}_{\Gamma\cup E}$ morfismul canonic de factorizare şi $i : \mathcal{A}_{E,\Gamma} \longrightarrow \mathcal{A}_{\Gamma\cup E}$ unicul izomorfism cu proprietatea $\rho; j; i = t$.

Presupunem $\rho_s(a) \equiv_\Gamma \rho_s(c)$. Deoarece $j_s(\rho_s(a)) = j_s(\rho_s(c))$ aplicând izomorfismul i şi ţinând cont de egalitatea de mai sus deducem $t_s(a) = t_s(c)$ deci $a \equiv_{\Gamma\cup E} c$.

Reciproc, presupunem $a \equiv_{\Gamma\cup E} c$. Deoarece $t_s(a) = t_s(c)$ aplicând inversul izomorfismului i şi ţinând cont de egalitatea de mai sus deducem $j_s(\rho_s(a)) = j_s(\rho_s(c))$ deci $\rho_s(a) \equiv_\Gamma \rho_s(c)$. □

Propoziţia 1.4 ne dă o metodă de a evita ecuaţiile E care nu pot fi folosite ca axiome în rescriere.

Presupunem că avem de demonstrat $a \equiv_{\Gamma\cup E} c$ în $\mathcal{A}$. Propoziţia 1.4 ne spune că în loc să demonstrăm acest lucru, putem arăta că $\rho_s(a) \equiv_\Gamma \rho_s(b)$ în $\mathcal{A}_E$. Dacă am dorii să folosim metoda dată în lecţiile anterioare, pentru a o demonstra vom ajunge la rescrieri în $\mathcal{A}_E$.

2 Rescrierea modulo o relaţie de echivalenţă

Rescrierea în $\mathcal{A}_E$ se numeşte *rescriere de clase*. Deoarece clasele de echivalenţă ale relaţiei $\equiv_E$ pot fi infinite, vom prefera să le înlocuim pe acestea cu un reprezentant din $\mathcal{A}$ care la rândul lui va putea fi substituit cu un alt reprezentant al aceleiaşi clase şi apoi rescris în $\mathcal{A}$. Această rescriere din $\mathcal{A}$ este denumită *rescriere modulo E*. Scopul acesteia este ca trecând la clase, să producă rescrierea din $\mathcal{A}_E$.

Conceptul de rescriere modulo E poate fi generalizat prin conceptul de **rescriere modulo o relaţie de echivalenţă** $\sim$ ce poate fi cu uşurinţă formalizat. Conceptul de rescriere modulo o relaţie de echivalenţă $\sim$ se obţine din conceptul clasic de rescriere la care se adaugă o nouă regulă de deducţie numită regula claselor.

Cl$_\sim$ $\quad a \sim_s b$ implică $a \doteq_s b$

De remarcat faptul că **Cl**$_\sim$ este o regulă deductivă mai puternică decît **R**, prin urmare în rescrierea modulo $\sim$ regula **Cl**$_\sim$ va înocui pe **R**.

Utilizarea fără restricţii a regulei claselor duce la pierderea proprietăţii de terminare. Un element poate fi rescris în altul din aceeaşi clasă, acesta la rândul lui poate fi rescris în alt element echivalent şi aşa mai departe. Observăm că un şir finit de astfel de resrieri poate fi înlocuit cu una singură deoarece toate elementele care apar în rescriere sunt din aceeaşi clasă. Prin urmare, restricţia de a nu folosi de două ori consecutiv regula claselor nu restrânge puterea acestei reguli şi este suficientă pentru implementare.

2.1 Preliminarii

Fie un morfism $h\colon \mathcal{A} \longrightarrow \mathcal{B}$ şi z o variabilă de sort s.
Fie $h^z\colon T_\Sigma(\mathcal{A} \cup \{z\}) \longrightarrow T_\Sigma(\mathcal{B} \cup \{z\})$ unicul morfism astfel încît

$$h^z(x) = \begin{cases} z & x = z \\ h(x) & x \neq z \end{cases}, \text{ pentru orice } x \in \mathcal{A}.$$

Pentru orice c din $T_\Sigma(\mathcal{B} \cup \{z\})$ fie $nr_z^{\mathcal{B}}(c)$ numărul de apariţii ale lui z în c. Deoarece $h^z; nr_z^{\mathcal{B}} = nr_z$ rezultă că pentru c din $T_\Sigma(\mathcal{A} \cup \{z\})$, $h^z(c)$ este context dacă şi numai dacă c este context.

Pentru oricare a din $\mathcal{A}$

$$(z \leftarrow a); h = h^z; (z \leftarrow h(a)).$$

Dacă în plus morfismul $h\colon \mathcal{A} \longrightarrow \mathcal{B}$ este surjectiv pe componente, atunci morfismul $h^z\colon T_\Sigma(\mathcal{A} \cup \{z\}) \longrightarrow T_\Sigma(\mathcal{B} \cup \{z\})$ este şi el surjectiv pe componente şi pentru fiecare context $c' \in T_\Sigma(\mathcal{B} \cup \{z\})$ există un context $c \in T_\Sigma(\mathcal{A} \cup \{z\})$ astfel încît $h^z(c) = c'$.

2.2 Corectitudinea

Rescrierea din algebra $\mathcal{B}$ este notată cu $\overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma}$. În continuare vom nota prin $\overset{h}{\Rightarrow}$ rescrierea din $\mathcal{A}$ modulo echivalenţa nucleară $Ker(h)$ a morfismului $h\colon \mathcal{A} \longrightarrow \mathcal{B}$.

Propoziţie 2.1 *Fie un morfism $h\colon \mathcal{A} \longrightarrow \mathcal{B}$. Dacă $a \overset{h}{\Rightarrow} d$ atunci $h(a) \overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma} h(d)$.*

Demonstraţie: Vom arăta că mulţimea de propoziţii din $\mathcal{A}$

$$D = \{a \doteq_s d \ \ : \ \ h_s(a) \overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma} h_s(d)\}$$

este închisă la regulile $\mathbf{Cl}_{Ker(h)}$, $\mathbf{T}$ şi $\mathbf{SRew}_\Gamma$.

Deoarece demonstraţia este evidentă pentru $\mathbf{Cl}_{Ker(h)}$ şi $\mathbf{T}$ vom demonstra numai închiderea faţă de regula $\mathbf{SRew}_\Gamma$.

Fie $(\forall X) l \doteq_s r$ **if** $H \in \Gamma$ şi $f\colon T_\Sigma(X) \longrightarrow \mathcal{A}$ un morfism astfel încît pentru orice $u \doteq_t v$ există $a_{uv} \in A_t$ cu proprietăţile $f_t(u) \doteq_t a_{uv} \in D$ şi $f_t(v) \doteq_t a_{uv} \in D$. Prin urmare $h_t(f_t(u)) \overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma} h_t(a_{uv})$ şi $h_t(f_t(v)) \overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma} h_t(a_{uv})$.

Pentru orice context c din $T_\Sigma(\mathcal{A} \cup \{z\})$, deoarece $\overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma}$ este închis la $\mathbf{SRew}_\Gamma$ obţinem

$$h^z(c)[(fh)_s(l)] \ \overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma} \ h^z(c)[(fh)_s(r)].$$

Rezultă că

$$h(c[f_s(l)]) \ \overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma} \ h(c[f_s(r)])$$

ceea ce încheie demonstraţia, deoarece arată că $c[f_s(l)] \doteq_s c[f_s(r)] \in D$.

Deoarece D este închisă la regulile $\mathbf{Cl}_{Ker(h)}$, $\mathbf{T}$ şi $\mathbf{SRew}_\Gamma$ rezultă că

$$\overset{h}{\Rightarrow} \ \subseteq \ D$$

fapt care implică concluzia. □

Din propoziţia de mai sus aplicată pentru morfismul de factorizare canonic rezultă că dacă a se rescrie modulo o echivalenţă $\sim$ în b, atunci clasa de echivalenţă a lui a se rescrie în clasa de echivalenţă a lui b.

Corolar 2.2 *Dacă $a \overset{*}{\Rightarrow}_\Gamma d$ atunci $h(a) \overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma} h(d)$.*

2.3 Completitudinea

Propoziţie 2.3 *Fie $h\colon \mathcal{A} \longrightarrow \mathcal{B}$ un morfism surjectiv pe componente. Dacă $h(a) \overset{*}{\Rightarrow}^{\mathcal{B}}_{\Gamma} h(d)$ atunci $a \overset{h}{\Rightarrow} d$.*

Demonstraţie: Fie mulţimea

$$D = \{(h_s(a), h_s(d)) \ : \ a \overset{h}{\Rightarrow} d\}.$$

Observăm că $(h_s(u), h_s(v)) \in D$ implică $u \overset{h}{\Rightarrow} v$. Din ipoteză există a, b asfel încât $h_s(u) = h_s(a)$, $a \overset{h}{\Rightarrow} d$ şi $h_s(d) = h_s(v)$, prin urmare $u \overset{h}{\Rightarrow} v$.

Probăm că D este închisă la $\mathbf{R}$,$\mathbf{T}$ şi $\mathbf{SRew}_\Gamma$.

R. Fie $d \in B_s$. Deoarece h_s este surjectivă există $a \in A_s$ cu $h_s(a) = d$. Din $a \overset{h}{\Rightarrow} a$ deoarece $\overset{h}{\Rightarrow}$ este reflexivă, rezultă că $h_s(a) \doteq_s h_s(a) \in D$, prin urmare $d \doteq_s d \in D$. Deci D este reflexivă.

T. Fie (x, y) şi (y, z) în D. există $a \overset{h}{\Rightarrow} v$ şi $u \overset{h}{\Rightarrow} d$ cu $x = h(a)$, $h(v) = y$, $h(u) = y$ şi $h(d) = z$.

Deoarece $h(v) = h(u)$ deducem $v \overset{h}{\Rightarrow} u$. Folosind şi $a \overset{h}{\Rightarrow} v$ şi $u \overset{h}{\Rightarrow} d$ deducem prin **T** că $a \overset{h}{\Rightarrow} d$, deci $(x, z) \in D$. În concluzie D este tranzitivă.

SRew$_\Gamma$. Fie $(\forall X) l \doteq_s r$ **if** $H \in \Gamma$, un morfism $f : T_\Sigma(X) \to \mathcal{B}$ astfel încât pentru orice $u \doteq_t v \in H$ există $b_{uv} \in B_t$ cu proprietăţile $f_t(u) \doteq_t b_{uv} \in D$ şi $f_t(v) \doteq_t b_{uv} \in D$. Pentru orice context $c \in T_\Sigma(\mathcal{B} \cup \{z\})_{s'}$ dorim să dovedim că

$$(c[f_s(l)], c[f_s(r)]) \in D.$$

Deoarece Σ-algebra liberă $T_\Sigma(X)$ este proiectivă şi h este un morfism surjectiv pe componente există un morfism $g\colon T_\Sigma(X) \longrightarrow \mathcal{A}$ astfel încît $g; h = f$.

Deoarece h este surjectiv pe componente pentru orice $u \doteq_t v \in H$ există $a_{uv} \in A_t$ cu proprietatea $h_t(a_{uv}) = b_{uv}$.

Observăm că pentru orice $u \doteq_t v \in H$, din

$$(h_t(g_t(u)), h_t(a_{uv})) \in D \text{ şi } (h_t(g_t(u)), h_t(a_{uv})) \in D$$

deducem

$$g_t(u) \overset{h}{\Rightarrow} a_{uv} \text{ şi } g_t(v) \overset{h}{\Rightarrow} a_{uv}.$$

Deoarece există un context $c' \in T_\Sigma(\mathcal{A} \cup \{z\})_{s'}$ astfel încît $h^z_{s'}(c') = c$ folosind **SRew**$_\Gamma$ deducem că $c'[g_s(l)] \overset{h}{\Rightarrow} c'[g_s(r)]$.

Prin urmare

$$(h_{s'}(c'[g_s(l)]), h_{s'}(c'[g_s(r)])) \in D.$$

Dar,

$$c[f_s(l)] = h^z_{s'}(c')[h_s(g_s(l))] = h_{s'}(c'[g_s(l)])$$

şi

$$c[f_s(r)] = h^z_{s'}(c')[h_s(g_s(r))] = h_{s'}(c'[g_s(r)]).$$

Deci $(c[f_s(l)], c[f_s(r)]) \in D$. □

Corolar 2.4 *a se rescrie modulo o echivalenţă $\sim$ în b, dacă şi numai dacă clasa de echivalenţă a lui a se rescrie în clasa de echivalenţă a lui b.*

Aceste propoziţii arată că rescrierea claselor într-o algebră cît $\mathcal{A}/\sim$ poate fi înlocuită cu rescrierea modulo $\sim$ din $\mathcal{A}$. Rescrierea modulo ecuaţii este doar un caz particular.

Comentariu. În acest moment cititorul poate înţelege dece am preferat să ne bazăm lecţiile pe rescrieri într-o algebră, stil pe care nu l-am mai întâlnit până în prezent. Autorii preferă să înceapă cu rescrieri într-o algebră liberă(term rewriting) şi în momentul când ajung la rescrierile modulo ecuaţii să expună aceeaşi teorie privind acest alt tip de rescriere. În concluzie rescrierile într-o algebra unfică rescrieriele într-o algebră liberă cu rescrierile modulo ecuaţii.

8 DEMONSTRAREA ECUAŢIILOR CONDIŢIONATE

Virgil Emil Căzănescu

February 14, 2010

Pentru două mulţimi de Σ-ecuaţii condiţionate Γ şi Γ', vom scrie $\Gamma \models_\Sigma \Gamma'$ dacă şi numai dacă $\mathcal{A} \models_\Sigma \Gamma$ implică $\mathcal{A} \models_\Sigma \Gamma'$ pentru orice Σ-algebră $\mathcal{A}$. În acest caz spunem că Γ' este **consecinţă semantică** a lui Γ.

Până acum ne-am ocupat numai de ecuaţiile necondiţionate adevărate în orice Γ-algebră. In această lecţie vom da o cale prin care se poate demonstra că o ecuaţie condiţionată γ este adevărată în orice Γ-algebră, adică $\Gamma \models_\Sigma \gamma$.

1 Preliminarii

Se arată uşor că

$$\mathcal{B} \models_\Sigma (\forall X) l \doteq_s r \textbf{ if } H \text{ şi } \mathcal{B} \models_\Sigma \{(\forall X) u \doteq_t v \mid u \doteq_t v \in H\} \text{ implică } \mathcal{B} \models_\Sigma (\forall X) l \doteq_s r.$$

Întradevăr pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{B}$ din $\mathcal{B} \models_\Sigma \{(\forall X) u \doteq_t v \mid u \doteq_t v \in H\}$ deducem $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$. Prin urmare din $\mathcal{B} \models_\Sigma (\forall X) l \doteq_s r$ **if** H rezultă că $h_s(l) = h_s(r)$.

Din păcate reciproca nu este adevărată în general, ci numai în cazul particular al Σ-algebrei iniţiale.

Lemă 1.1

$$\mathcal{B} \models_\Sigma (\forall \emptyset) l \doteq_s r \textbf{ if } H \quad \textit{dacă şi numai dacă} \quad [\mathcal{B} \models_\Sigma \{(\forall \emptyset) u \doteq_t v \mid u \doteq_t v \in H\} \quad \textit{implică} \quad \mathcal{B} \models_\Sigma (\forall \emptyset) l \doteq_s r].$$

Demonstraţie: Presupunem că $\mathcal{B} \models_\Sigma \{(\forall \emptyset) u \doteq_t v \mid u \doteq_t v \in H\}$ implică $\mathcal{B} \models_\Sigma (\forall \emptyset) l \doteq_s r$.

Fie $h \colon T_\Sigma \longrightarrow \mathcal{B}$ cu $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$.

Probăm că $\mathcal{B} \models_\Sigma \{(\forall \emptyset) u \doteq_t v \mid u \doteq_t v \in H\}$. Fie $g \colon T_\Sigma \longrightarrow \mathcal{B}$. Deoarece T_Σ este iniţială deducem că $g = h$, deci $g_t(u) = g_t(v)$ pentru orice $u \doteq_t v \in H$.

Din ipoteză deducem $\mathcal{B} \models_\Sigma (\forall \emptyset) l \doteq_s r$, deci $h_s(l) = h_s(r)$. □

Corolar 1.2

$$\Gamma \models_\Sigma (\forall \emptyset) l \doteq_s r \textbf{ if } H \quad \textit{dacă şi numai dacă} \quad \Gamma \cup \{(\forall \emptyset) u \doteq_t v \mid u \doteq v \in H\} \models_\Sigma (\forall \emptyset) l \doteq_s r.$$

Demonstraţie: Demonstraţia se va face printr-un şir de echivalenţe. Plecăm de la

$$\Gamma \models_\Sigma (\forall \emptyset) l \doteq_s r \textbf{ if } H$$

care prin definiţie este echivalentă cu

$$(\forall \mathcal{B} \models_\Sigma \Gamma) \mathcal{B} \models_\Sigma (\forall \emptyset) l \doteq_s r \textbf{ if } H.$$

Conform lemei afirmaţia de mai sus este echivalentă cu

$$(\forall \mathcal{B} \models_\Sigma \Gamma)(\mathcal{B} \models_\Sigma \{(\forall \emptyset) u \doteq_t v \mid u \doteq v \in H\} \text{ implică } \mathcal{B} \models_\Sigma (\forall \emptyset) l \doteq_s r)$$

care este echivalentă cu

$$(\forall \mathcal{B} \models_\Sigma (\Gamma \cup \{(\forall \emptyset) u \doteq_t v \mid u \doteq v \in H\})) \mathcal{B} \models_\Sigma (\forall \emptyset) l \doteq_s r$$

adică cu

$$\Gamma \cup \{(\forall \emptyset) u \doteq_t v \mid u \doteq v \in H\} \models_\Sigma (\forall \emptyset) l \doteq_s r. \square$$

Lema şi corolarul precedent ne permit să spargem o ecuaţie condiţionată de demonstrat în părţile ei componente. Din păcate acest fapt este posibil numai pentru ecuaţiile condiţionate fără variabile. Teorema constantelor ne va permite să înlocuim demonstrarea unei ecuaţii condiţionate cu variabile cu o ecuaţie condiţionată fără variabile.

2 Schimbarea signaturii

Prima etapă constă în înlocuirea ecuaţiei condiţionate cu o alta cu o alta "echivalentă" dintr-o algebră iniţială. Pentru aceasta este necesară schimbarea signaturii prin transformarea variabilelor din ecuaţia condiţionată de demonstrat în constante (simboluri de operaţii fără argumente).

Fie Σ o signatură S-sortată şi X o mulţime S-sortată de variabile disjunctă de Σ. Vom lucra în continuare cu o nouă signatură S-sortată Σ_X care include Σ şi în care fiecare $x \in X_s$ devine un simbol de operaţie fără argumente de sort s.

Vom nota cu $j : \Sigma \longrightarrow \Sigma_X$ morfismul incluziune de signaturi. Vom nota functorul uituc cu

$$Mod(j) : Alg_{\Sigma_X} \longrightarrow Alg_{\Sigma}.$$

In categoria Σ_X-algebrelor Alg_{Σ_X} vom utiliza o notaţie specială şi anume o Σ_X-algebră va fi scrisă ca o pereche $(\mathcal{A}, a)$ unde $\mathcal{A}$ este o Σ-algebră şi $a : X \longrightarrow \mathcal{A}$ este o funcţie S-sortată. Prin Σ-algebra $\mathcal{A}$ se dau suporturile şi operaţiile corespunzătoare simbolurilor din Σ iar prin funcţia a se dau operaţiile corespunzătoare simbolurilor din X, adică operaţia corespunzătoare lui $x \in X_s$ este $a_s(x)$.

Observăm că $h : (\mathcal{A}, a) \longrightarrow (\mathcal{B}, b)$ este un morfism de Σ_X-algebre dacă şi numai dacă $h : \mathcal{A} \longrightarrow \mathcal{B}$ este morfism de Σ-algebre şi $a; h = b$.

Având in vedere notaţiile de mai sus observăm că

$$Mod(j)(\mathcal{A},\, a) = \mathcal{A} \text{ pentru orice } \Sigma_X\text{-algebră } (\mathcal{A},\, a) \text{ şi}$$

$$Mod(j)(h) = h \text{ pentru orice } \Sigma_X\text{-morfism } h.$$

Schimbarea signaturii Σ cu Σ_X impune translatarea Σ-axiomelor în Σ_X-axiome.

3 Translatarea ecuaţiilor

Vom nota cu Y o mulţime de variabile cuantificată universal într-o Σ-axiomă. Fără a restrânge generalitatea putem presupune că Y este disjunctă de X. Vom nota cu

$$i : T_\Sigma(Y) \longrightarrow T_\Sigma(X \cup Y)$$

unicul Σ-morfism pentru care $i(y) = y$ pentru orice y din Y. Fără a restrânge generalitatea putem presupune că i este o incluziune.

Mai observăm că

$$T_{\Sigma_X}(Y) = (T_\Sigma(X \cup Y), X \hookrightarrow T_\Sigma(X \cup Y)).$$

Prin urmare $Mod(j)(T_{\Sigma_X}(Y)) = T_\Sigma(X \cup Y)$.

Observaţia 3.1 *Pentru orice Σ_X-algebră $(\mathcal{M}, m)$ funcţia*

$$F : Alg_{\Sigma_X}(T_{\Sigma_X}(Y), (\mathcal{M}, m)) \longrightarrow Alg_\Sigma(T_\Sigma(Y), \mathcal{M})$$

definită prin $F(h) = i; Mod(j)(h)$ este o bijecţie.

Demonstraţie: Observăm că $F(h)$ este restricţia lui h la $T_\Sigma(Y)$.

Fie Σ-morfismul $u : T_\Sigma(Y) \longrightarrow \mathcal{M}$ şi $g : T_\Sigma(X \cup Y) \longrightarrow \mathcal{M}$ unicul morfism cu proprietăţile

$$g(z) = \begin{cases} u(z) & \text{dacă } z \in Y, \\ m(z) & \text{dacă } z \in X. \end{cases}$$

Observăm că $g : T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m)$ este unicul Σ_X-morfism a cărui restricţie la $T_\Sigma(Y)$ coincide cu u. □

Faptele expuse până acum ca şi lema următoare au un caracter tehnic. Deoarece signatura a fost schimbată este necesar ca axiomele să fie rescrise în noua signatură. Mulţimea Y reprezintă variabilele cuantificate universal dintr-o axiomă. Ele nu sunt afectate prin trecerea la noua signatură. Schimbarea signaturii axiomelor este mai mult formală caci în noile axiome constantele obţinute din variabilele din X nu sunt practic folosite. Aceasta este motivul pentru care în lema şi cololarul care urmează funcţia m, prin care sunt date constantele, apare numai în membrul stâng, membrul drept fiind independent de aceasta.

Fiecărei Σ-ecuaţii condiţionate

$$\gamma = (\forall Y) l \doteq_s r \textbf{ if } H$$

îi ataşăm o altă Σ_X-ecuaţie condiţionată

$$J(\gamma) = (\forall Y) i_s(l) \doteq_s i_s(r) \textbf{ if } \{i_t(u) \doteq i_t(v) \mid u \doteq_t v \in H\}.$$

Funcţia J duce o Σ-ecuaţie necondiţionată într-o Σ_X-ecuaţie necondiţionată.

Lemă 3.2 Satisfacerea. *Pentru orice Σ_X-algebră $(\mathcal{M}, m)$*

$$(\mathcal{M}, m) \models_{\Sigma_X} J(\gamma) \textit{ dacă şi numai dacă } \mathcal{M} \models_\Sigma \gamma.$$

Demonstraţie: Observăm că $(\mathcal{M}, m) \models_{\Sigma_X} J(\gamma)$ este, prin definiţie, echivalentă cu

$$\forall h \colon T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m) \text{ dacă } h_t(i_t(u)) = h_t(i_t(v)) \text{ pentru orice } u \doteq_t v, \text{ atunci } h_s(i_s(l)) = h_s(i_s(r)),$$

prin urmare echivalentă cu

$$\forall h \colon T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m) \text{ dacă } (i; Mod(j)(h))_t(u) = (i; Mod(j)(h))_t(v) \text{ pentru orice } u \doteq_t v \in H, \text{ atunci } (i; Mod(j)(h))_s(l) = (i; Mod(j)(h))_s(r);$$

adică echivalentă cu

$$\forall h \colon T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m) \text{ dacă } F(h)_t(u) = F(h)_t(v) \text{ pentru orice } u \doteq_t v \in H, \text{ atunci } F(h)_s(l) = F(h)_s(r).$$

Folosind bijecţia de mai sus deducem că $(\mathcal{M}, m) \models_{\Sigma_X} J(\gamma)$ este echivalentă cu

$$\forall g \colon T_\Sigma(Y) \longrightarrow \mathcal{M} \text{ dacă } g_t(u) = g_t(v) \text{ pentru orice } u \doteq_t v \in H, \text{ atunci } g_s(l) = g_s(r)$$

adică cu $\mathcal{M} \models_\Sigma \gamma$. □

Pentru o mulţime Γ de Σ-ecuaţii condiţionate notăm $J(\Gamma) = \{J(\gamma) \mid \gamma \in \Gamma\}$.

Corolar 3.3 *Pentru orice Σ_X-algebră $(\mathcal{M}, m)$ şi orice mulţime Γ de Σ-ecuaţii condiţionate*

$$(\mathcal{M}, m) \models_{\Sigma_X} J(\Gamma) \textit{ dacă şi numai dacă } \mathcal{M} \models_\Sigma \Gamma. \square$$

4 Teorema constantelor

Teorema 4.1 Teorema constantelor:

$$\Gamma \models_\Sigma (\forall X) l \doteq_s r \textbf{ if } H \textit{ dacă şi numai dacă } J(\Gamma) \models_{\Sigma_X} (\forall \emptyset) l \doteq_s r \textbf{ if } H.$$

Demonstraţie: La început vom menţiona forme echivalente pentru cei doi membri ai echivalenţei din enunţ.
Membrul stâng

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

este echivalent succesiv cu:

1. Oricare ar fi Σ-algebra $\mathcal{A}$, dacă $\mathcal{A} \models_\Sigma \Gamma$, atunci $\mathcal{A} \models_\Sigma (\forall X) l \doteq_s r$ **if** H,
2. Oricare ar fi Σ-morfismul $h : T_\Sigma(X) \longrightarrow \mathcal{A} \models \Gamma$ dacă $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$, atunci $h_s(l) = h_s(r)$.

Membrul drept

$$J(\Gamma) \models_{\Sigma_X} (\forall \emptyset) l \doteq_s r \textbf{ if } H$$

este echivalent succesiv cu:

1. Oricare ar fi Σ_X-algebra $(\mathcal{M}, m)$, dacă $(\mathcal{M}, m) \models_{\Sigma_X} J(\Gamma)$, atunci $(\mathcal{M}, m) \models_{\Sigma_X} (\forall \emptyset) l =_s r$ **if** H,
2. Oricare ar fi Σ_X-morfismul $h \colon T_{\Sigma_X} \longrightarrow (\mathcal{M}, m) \models_{\Sigma_X} J(\Gamma)$ dacă $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$, atunci $h_s(l) = h_s(r)$.

Presupunem că $\Gamma \models_\Sigma (\forall X) l \doteq_s r$ **if** H. Fie $h\colon T_{\Sigma_X} \longrightarrow (\mathcal{M}, m) \models_{\Sigma_X} J(\Gamma)$ cu $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$.

Deoarece $\mathcal{M} \models_\Sigma \Gamma$ rezultă că $\mathcal{M} \models_\Sigma (\forall X) l \doteq_s r$ **if** H prin urmare din $h\colon T_\Sigma(X) \longrightarrow \mathcal{M}$ şi $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$ deducem $h_s(l) = h_s(r)$.

Reciproc, presupunem că $J(\Gamma) \models_{\Sigma_X} (\forall \emptyset) l \doteq_s r$ **if** H. Fie $h\colon T_\Sigma(X) \longrightarrow \mathcal{M} \models_\Sigma \Gamma$ cu $h_t(u) = h_t(v)$ pentru orice $u \doteq_t v \in H$.

Deoarece $(\mathcal{M}, h_{/X}) \models_{\Sigma_X} J(\Gamma)$, $h : T_{\Sigma_X} \longrightarrow (\mathcal{M}, h_{/X})$ este un Σ_X-morfism şi $h_t(u) = h_t(v)$ pentru norice $u \doteq_t v \in H$ din ipoteză deducem $h_s(l) = h_s(r)$. □

Acestă teoremă ne permite să eliminăm cuantificatorul universal prin înlocuirea variabilelor cuantificate prin simboluri de operaţii constante. Astfel primul obiectiv de a ne muta într-o algebră iniţială a fost atins.

Teorema următoare nu face decât să tragă concluziile, furnizând o cale prin care putem încerca demonstarea ecuaţiilor condiţionate. Problema care rămâne este demonstrarea corectitudinii programului obţinut folosind noul set de axiome în ipoteza că programul dat de axiomele Γ este corect.

Teorema 4.2 Teorema deducţiei:

$$\Gamma \models_\Sigma (\forall X) l \doteq_s r \textbf{ if } H \quad \textit{dacă şi numai dacă} \quad J(\Gamma) \cup \{(\forall \emptyset) u \doteq_t v \mid u \doteq_t v \in H\} \models_{\Sigma_X} (\forall \emptyset) l \doteq_s r.$$

Demonstraţie: Conform teoremei constantelor

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

este echivalentă cu

$$J(\Gamma) \models_{\Sigma_X} (\forall \emptyset) l \doteq_s r \textbf{ if } H$$

care conform corolarului 1.2 este echivalentă cu

$$J(\Gamma) \cup \{(\forall \emptyset) u \doteq_t v \in H\} \models_{\Sigma_X} (\forall \emptyset) l \doteq_s r. \square$$

In concluzie pentru demonstrarea Σ-ecuaţiei condiţionate $(\forall X) l \doteq_s r$ **if** H putem proceda astfel

- schimbăm signatura Σ cu signatura Σ_X, translatând axiomele din Γ în noua signatură
- adăugăm $\{(\forall \emptyset) u \doteq_t v \mid u \doteq_t v \in H\}$ la axiome şi
- demonstrăm $(\forall \emptyset) l \doteq_s r$ în noul context.

9 PERECHI CRITITCE

Virgil Emil Căzănescu

January 26, 2010

Toate algebrele sunt libere. Presupunem în plus că Γ conţine numai ecuaţii necondiţionate.

Mai menţionăm că membrul stâng al unei ecuaţii nu poate fi format numai dintr-o variabilă, deoarece în caz contrar orice expresie ar putea fi rescrisă. O consecinţă a acestui fapt este că nici o variabilă nu poate fi rescrisă.

Lemă 0.1 *Pentru orice $c \in \mathcal{A}[z]$ dacă $a \stackrel{*}{\Rightarrow}_\Gamma d$, atunci $c[a] \stackrel{*}{\Rightarrow}_\Gamma c[d]$.*

Demonstraţie: Prin inducţie structurală după c.

Dacă $c = z$ concluzia coincide cu ipoteza.

Dacă $c \in A$ utilizăm reflexivitatea.

Dacă $c = \sigma(c_1, c_2, \ldots, c_n)$ din ipoteza de inducţie $c_i[a] \stackrel{*}{\Rightarrow}_\Gamma c_i[d]$ pentru orice i, prin urmare aplicând $\mathbf{C}\Sigma$ deducem

$$A_\sigma(c_1[a], c_2[a], \ldots, c_n[a]) \stackrel{*}{\Rightarrow}_\Gamma A_\sigma(c_1[d], c_2[d], \ldots, c_n[d]),$$

deci $c[a] \stackrel{*}{\Rightarrow}_\Gamma c[d]$. □

Lemă 0.2 *Pentru orice $c \in \mathcal{A}[z]$ dacă $a \downarrow_\Gamma d$, atunci $c[a] \downarrow_\Gamma c[d]$.*

Demonstraţie: Deoarece există m astfel încât $a \stackrel{*}{\Rightarrow}_\Gamma m$ şi $d \stackrel{*}{\Rightarrow}_\Gamma m$ din lema precedentă deducem
$c[a] \stackrel{*}{\Rightarrow}_\Gamma c[m]$ şi $c[d] \stackrel{*}{\Rightarrow}_\Gamma c[m]$ de unde rezultă concluzia. □

Lemă 0.3 *Dacă y nu apare în c, atunci $(y \longrightarrow b)(c[a]) = c[(y \longrightarrow b)(a)]$.* □

Definiţia 0.4 Fie $(\forall X_1) l_1 \doteq_{s'} r_1$ şi $(\forall X_2) l_2 \doteq_{s''} r_2$ două axiome din Γ fără variabile comune, ipoteză care nu restrânge generalitatea. Fie c în $\mathcal{A}[z]_{s'}$ un context astfel încât

$$l_1 = c[d]$$

unde d nu este o variabilă. Fie u un cel mai general unificator al lui d şi al lui l_2. Elementele

$$u(r_1) \text{ şi } u(c[r_2])$$

formează o **pereche critică**. □

Pe scurt notăm mulţimea perechilor critice

$$\mathbf{PC}_\Gamma = \{(u(r_1), u(c[r_2])) \ : \ (\forall X_1) l_1 \doteq_{s'} r_1, (\forall X_2) l_2 \doteq_{s''} r_2 \in \Gamma,\ l_1 = c[d],\ d \text{ nu este variabilă şi } u = cgu\{d, l_2\}\ \}$$

Propoziţiile din care se formează perechile critice sunt axiome, eventual identice, din Γ unde, la nevoie, variabilele uneia sunt înlocuite cu variabile noi pentru ca cele două propoziţii să devină fără variabile comune.

Observaţia 0.5 *Dacă rescrierea este local confluentă, atunci $\mathbf{PC}_\Gamma \subseteq \downarrow_\Gamma$.*

Demonstraţie: Observăm că

$$u(l_1) \Rightarrow_\Gamma u(r_1) \text{ şi}$$

$$u(l_1) = u^z(c)[u(d)] = u^z(c)[u(l_2)] \Rightarrow_\Gamma u^z(c)[u(r_2)] = u(c[r_2]).$$

Deoarece rescrierea este local confluentă deducem că

$$u(r_1) \downarrow_\Gamma u(c[r_2]).$$ □

Teorema 0.6 *Dacă* $\mathbf{PC}_\Gamma \subseteq \downarrow_\Gamma$, *atunci rescrierea este local confluentă.*

Demonstraţie: Fără a micşora generalitatea vom presupune că toate mulţimile de variabile cu care lucrăm sunt disjuncte două câte două şi că variabilele nou introduse sunt distincte între ele şi nu aparţin altor mulţimi de variabile.

Presupunem că

$$a \Rightarrow a_1 \text{ şi } a \Rightarrow a_2.$$

Prin urmare pentru $i \in [2]$ există $(\forall X_i) l_i \doteq r_i \in \Gamma$, contextul $c_i \in \mathcal{A}[z_i]$ şi morfismul $h_i : T_\Sigma(X_i) \longrightarrow \mathcal{A}$ astfel încât $a = c_i[h_i(l_i)]$ şi $a_i = c_i[h_i(r_i)]$.

Cei doi subarbori $h_1(l_1)$ şi $h_2(l_2)$ ai lui a pot fi disjuncţi(cazul 1 de mai jos) sau incluşi unul în celălalt(cazurile 2 şi 3 trei de mai jos). În cazurile 2 şi 3 vom presupune fără a restrânge generalitatea că $h_2(l_2)$ este subarbore a lui $h_1(l_1)$.

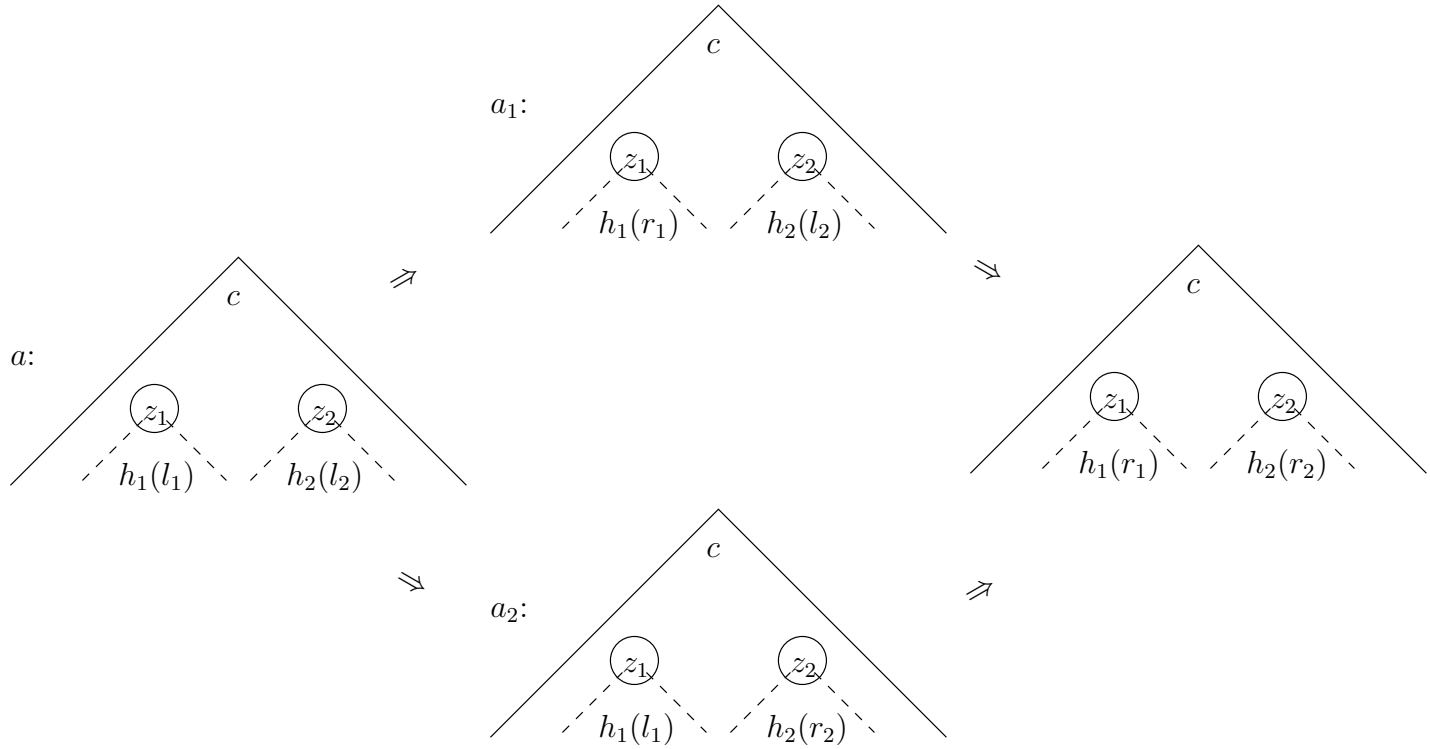


Figure 1: Cazul 1. Subarbori disjuncţi

Cazul 1(fără suprapunere). Există $c \in \mathcal{A}[z_1, z_2]$ cu câte o apariţie a lui z_1 şi z_2 astfel încât

$$c_1 = (z_2 \leftarrow h_2(l_2))(c) \text{ şi } c_2 = (z_1 \leftarrow h_1(l_1))(c).$$

Deoarece z_1 nu apare în $h_2(l_2)$ şi nici z_2 nu apare în $h_1(r_1)$ deducem

$$(z_1 \leftarrow h_1(r_1))[(z_2 \leftarrow h_2(l_2))(c)] = (z_2 \leftarrow h_2(l_2))[(z_1 \leftarrow h_1(r_1))(c)].$$

Observăm că
$a_1 = (z_1 \leftarrow h_1(r_1))[(z_2 \leftarrow h_2(l_2))(c)] = (z_2 \leftarrow h_2(l_2))[(z_1 \leftarrow h_1(r_1))(c)]$ prin urmare $a_1 \Rightarrow (z_2 \leftarrow h_2(r_2))[(z_1 \leftarrow h_1(r_1))(c)]$.
Observăm că
$a_2 = (z_2 \leftarrow h_2(r_2))([z_1 \leftarrow h_1(l_1)](c)) = \{(z_2 \leftarrow h_2(r_2))(c)\}[h_1(l_1)]$
prin urmare
$a_2 \Rightarrow (z_1 \leftarrow h_1(r_1))[(z_2 \leftarrow h_2(r_2))(c)] = [z_2 \leftarrow h_2(r_2)][(z_1 \leftarrow h_1(r_1))(c)]$, deci $a_1 \downarrow_\Gamma a_2$.

Cazurile 2-3(cu suprapunere, adică $h_2(l_2)$ este subexpresie în imaginea prin h_1 a lui l_1).
Cazul 2(cu suprapunere critică, adică $h_2(l_2)$ este imaginea prin h_1 al unui subtermen d al lui l_1 care nu este variabilă).
Prin urmare

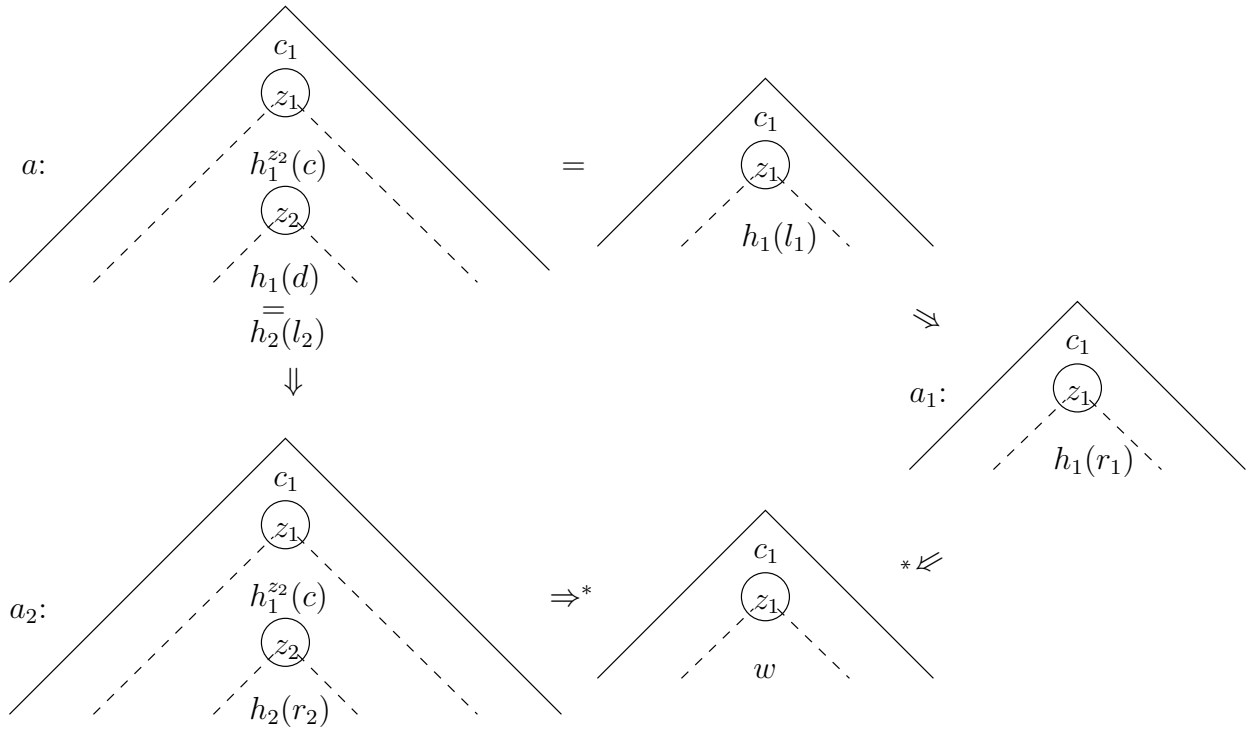


Figure 2: Cazul 2. Cu suprapunere critică

$l_1 = c[d]$ unde c este un context cu variabila distinsă z_2,

$$h_1(d) = h_2(l_2) \quad \text{şi} \quad c_2 = (z_1 \leftarrow h_1^{z_2}(c))(c_1).$$

Ultima egalitate este motivată de următoarele calcule

$$a = c_1[h_1(l_1)] = c_1[h_1(c[d])] = c_1[h_1^{z_2}(c)[h_1(d)]] = c_1[h_1^{z_2}(c)][h_2(l_2)].$$

Fie

$$h : T_\Sigma(X_1 \cup X_2) \longrightarrow \mathcal{A}$$

unicul morfism care extinde atât pe h_1 cât şi pe h_2. Observăm că h este un unificator pentru d şi l_2. Fie $u : T_\Sigma(X_1 \cup X_2) \longrightarrow \mathcal{T}$ un cel mai general unificator al lui d şi l_2 şi $t : \mathcal{T} \longrightarrow \mathcal{A}$ unicul morfism cu proprietatea $u; t = h$.

Doarece d nu este variabilă rezultă că $u(r_1)$ şi $u(c[r_2])$ formează o pereche critică. Din ipoteză deducem $u(r_1) \downarrow_\Gamma u(c[r_2])$. Aplicând morfismul t obţinem $t(u(r_1)) \downarrow_\Gamma t(u(c[r_2]))$, deci

$$h(r_1) \downarrow_\Gamma h(c[r_2]).$$

Prin urmare există w astfel încât

$$h_1(r_1) \stackrel{*}{\Rightarrow}_\Gamma w \text{ şi } h(c[r_2]) \stackrel{*}{\Rightarrow}_\Gamma w.$$

Din prima rescriere deducem $a_1 = c_1[h_1(r_1)] \stackrel{*}{\Rightarrow}_\Gamma c_1[w]$.

Utilizând a doua rescriere deducem

$a_2 = (z_1 \leftarrow h_1^{z_2}(c))(c_1)[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))((z_1 \leftarrow h_1^{z_2}(c))(c_1)) = (z_1 \leftarrow h_1^{z_2}(c); z_2 \leftarrow h_2(r_2))(c_1) =$
$=(z_1 \leftarrow h_1^{z_2}(c)[z_2 \leftarrow h_2(r_2)])(c_1) = c_1[h(c[r_2])] \stackrel{*}{\Rightarrow}_\Gamma c_1[w]$,
deci $a_1 \downarrow_\Gamma a_2$.

Cazul 3(cu suprapunere necritică, adică $h_2(l_2)$ este subexpresie în imaginea prin h_1 a unei variabile x din l_1).

Notăm cu $c \in \mathcal{A}[z_2]$ contextul în care se găseşte $h_2(l_2)$ în $h_1(x)$, prin urmare

$$h_1(x) = c[h_2(l_2)] \tag{1}$$

Fie

$$u : T_\Sigma(X_1) \longrightarrow \mathcal{A}[x]$$

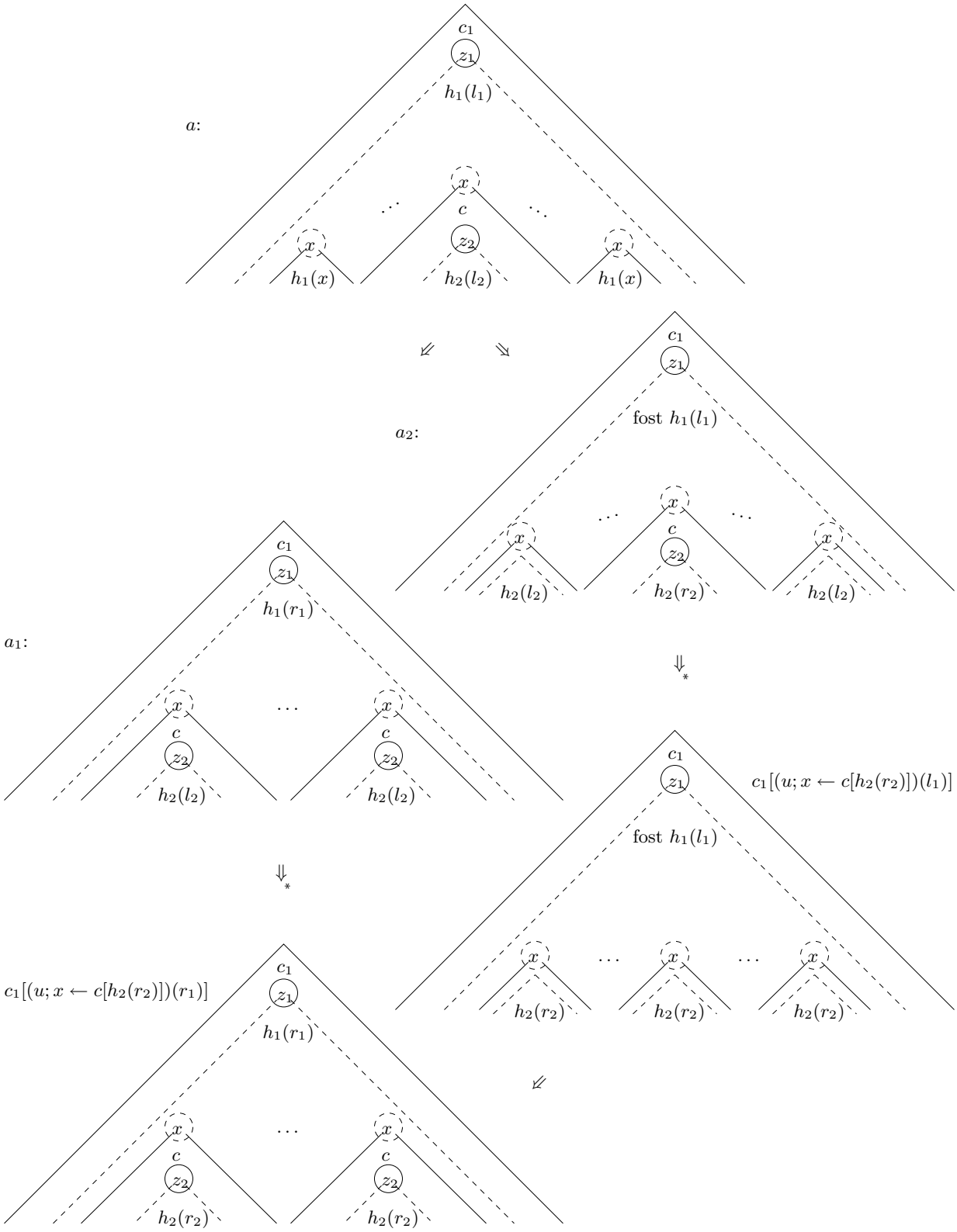


Figure 3: Cazul 3. Cu suprapunere necritică

unicul morfism cu $u(x) = x$ şi care acţionează ca h_1 pe $T_\Sigma(X_1 - \{x\})$. Observăm că $h_1 = u; x \leftarrow h_1(x)$.

În plus $x \leftarrow h_1(x) = x \leftarrow c[z_2 \leftarrow h_2(l_2)] = x \leftarrow c\,; z_2 \leftarrow h_2(l_2)$, deci

$$h_1 = u; x \leftarrow c; z_2 \leftarrow h_2(l_2).$$

Deoarece

$$h_1(r_1) = (z_2 \leftarrow h_2(l_2))((u; x \leftarrow c)(r_1))$$

observăm, deoarece z_2 nu apare în c_1, că

$$a_1 = c_1[(z_2 \leftarrow h_2(l_2))((u; x \leftarrow c)(r_1))] = (z_2 \leftarrow h_2(l_2))(c_1[(u; x \leftarrow c)(r_1)]) = (c_1[(u; x \leftarrow c)(r_1)])[h_2(l_2)],$$

prin urmare

$$a_1 \stackrel{*}{\Rightarrow}_\Gamma c_1[(u; x \leftarrow c)(r_1)])[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))(c_1[(u; x \leftarrow c)(r_1)]) = c_1[(z_2 \leftarrow h_2(r_2))((u; x \leftarrow c)(r_1))]$$

deci

$$a_1 \stackrel{*}{\Rightarrow}_\Gamma c_1[(u; x \leftarrow c; z_2 \leftarrow h_2(r_2))(r_1)] = c_1[(u; x \leftarrow c[h_2(r_2)])(r_1)].$$

Pentru a continua analizăm contextul c_2 şi observăm că în acesta apar subarbori de două tipuri provenind din variabila x din l_1

1) Apariţia lui x care crează subtermenul $h_2(l_2)$ care este rescris pentru a-l obţine pe a_2 şi care în c_2 este înlocuită cu subarborele c

2) Restul apariţiilor lui x din l_1 care în c_2 dau naştere unui subarbore $h_1(x)$.

Pentru a diferenţia aceste apariţii ale lui x în l procedăm după cum descriem în cele de mai jos.

Punând în evidenţă în l_1 apariţia lui x care prin aplicarea lui h_1 dă naştere subtermenului $h_2(l_2)$ care este rescris pentru a da naştere lui a_2 şi substituind-o cu o variabilă nouă z obţinem un context $l \in T_\Sigma(X_1 \cup \{z\})$ cu proprietatea $l_1 = l[x]$.

Fie $w : T_\Sigma(X_1 \cup \{z\}) \longrightarrow \mathcal{A}[z_2]$ unicul morfism cu proprietăţile $w(x_1) = h_1(x_1)$ pentru orice $x_1 \in X_1$ şi $w(z) = c$. Observăm că

$$z \leftarrow x; h_1 = w; z_2 \leftarrow h_2(l_2).$$

Folosind egalitatea tocmai dovedită, deoarece z_2 nu apare în c_1

$$a = c_1[h_1((z \leftarrow x)(l))] = c_1[(z_2 \leftarrow h_2(l_2))(w(l))] = (z_2 \leftarrow h_2(l_2))(c_1[w(l)]) = (c_1[w(l)])[h_2(l_2)]$$

deducem că

$$c_2 = c_1[w(l)].$$

Fie $p : T_\Sigma(X_1 \cup \{z\}) \longrightarrow \mathcal{A}[x]$ morfismul definit prin $p(x_1) = h_1(x_1)$ pentru orice $x_1 \in X_1 - \{x\}$, $p(x) = x$ şi $p(z) = c[h_2(r_2)]$. Observăm că

$$w; z_2 \leftarrow h_2(r_2) = p; x \leftarrow h_1(x).$$

$a_2 = c_2[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))(c_1[w(l)]) = c_1[(w; z_2 \leftarrow h_2(r_2))(l)] = c_1[(p; x \leftarrow h_1(x))(l)]$

$= c_1[(z_2 \leftarrow h_2(l_2))((p; x \leftarrow c)(l))] = (z_2 \leftarrow h_2(l_2))(c_1[(p; x \leftarrow c)(l)]) = (c_1[(p; x \leftarrow c)(l)])[h_2(l_2)]$.

Prin urmare

$$a_2 \stackrel{*}{\Rightarrow}_\Gamma (c_1[(p; x \leftarrow c)(l)])[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))(c_1[(p; x \leftarrow c)(l)]) = c_1[(p; x \leftarrow c; z_2 \leftarrow h_2(r_2))(l)].$$

Observăm că

$$p; x \leftarrow c; z_2 \leftarrow h_2(r_2) = z \leftarrow x; u; x \leftarrow c[h_2(r_2)]$$

prin urmare

$$a_2 \stackrel{*}{\Rightarrow}_\Gamma c_1[(z \leftarrow x; u; x \leftarrow c[h_2(r_2)])(l)] = c_1[(u; x \leftarrow c[h_2(r_2)])(l_1)] \stackrel{*}{\Rightarrow}_\Gamma c_1[(u; x \leftarrow c[h_2(r_2)])(r_1)]$$

deci

$$a_1 \downarrow_\Gamma a_2. \square$$

10 TERMINAREA şi PROCEDURA KNUTH-BENDIX

Virgil Emil Căzănescu

November 16, 2009

1 Terminarea programelor

Terminarea este o propritate impusă de practica. Din păcate nu dispunem încă de metode puternice pentru a proba această proprietate. Dăm în cele ce urmează câteva fapte care ne permit uneori să probăm terminarea.

Reamintim că prin însăşi definiţia rescrierii într-un pas, $a \Rightarrow_\Gamma d$ implică $a \neq d$. Rescrierea $\stackrel{*}{\Rightarrow}_\Gamma$ are proprietatea de terminare dacă nu există şiruri $\{a_n\}_{n\in N}$ cu proprietatea $a_n \Rightarrow_\Gamma a_{n+1}$ pentru orice $n \in N$.

În teorema următoare vom folosi notaţia

$$a \models d \text{ dacă şi numai dacă } a = d \text{ sau } \rho(a) \vdash \rho(d).$$

Teorema 1.1 *Următoarele proprietăţi sunt echivalente*

1. *Rescrierea are proprietatea de terminare,*
2. $\stackrel{+}{\Rightarrow}_\Gamma$ *este o relaţie de ordine strictă şi noetheriană,*
3. *există o relaţie de ordine* $\geq$ *pe* A *compatibilă cu structura algebrică cu proprietăţile:*
   (a) *varianta ei strictă* $>$ *este noetheriană,*
   (b) *pentru orice* $(\forall \mathcal{P}) l =_s r$ **if** $H \in \Gamma$ *şi orice* $h : \mathcal{P} \to \mathcal{A}$ *dacă* $h(H) \subseteq \downarrow_\Gamma$, *atunci* $h(l) \geq h(r)$
4. *există o relaţie noetheriană* $\vdash$ *pe o mulţime* M *şi o funcţie* $\rho\colon A \longrightarrow M$ *cu proprietăţile*
   (a) *pentru orice* $(\forall \mathcal{P}) l =_s r$ **if** $H \in \Gamma$ *şi orice* $h : \mathcal{P} \to \mathcal{A}$ *dacă* $h(H) \subseteq \downarrow_\Gamma$, *atunci* $h(l) \models h(r)$
   (b) *pentru orice* $n \geq 1$ *şi* $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$,
   $$\rho(a) \vdash \rho(d) \text{ implică } A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) \models A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n),$$
5. *există o relaţie noetheriană* $\vdash$ *pe o mulţime* M *şi* $\rho\colon A \longrightarrow M$ *o funcţie cu proprietăţile*
   (a) *pentru orice* $(\forall \mathcal{P}) l =_s r$ **if** $H \in \Gamma$ *şi orice* $h : \mathcal{P} \to \mathcal{A}$ *dacă* $h(H) \subseteq \downarrow_\Gamma$, *atunci* $h(l) \models h(r)$,
   (b) *pentru orice context* $c \in \mathcal{A}[z]_{s'}$ *şi pentru orice* $a, d \in A_s$ *dacă* $\rho(a) \vdash \rho(d)$, *atunci* $c[z \leftarrow a] \models c[z \leftarrow d]$,
6. *există o relaţie noetheriană* $\vdash$ *pe mulţimea* M *şi* $\rho\colon A \longrightarrow M$ *o funcţie cu proprietatea*
   (a) $a \Rightarrow_\Gamma d$ *implică* $\rho(a) \vdash \rho(d)$.

Demonstraţie:

1⟶2. E suficient să probăm că $a \stackrel{+}{\Rightarrow}_\Gamma a$ este falsă pentru orice $a \in A$. Presupunând prin absurd existenţa lui $a \in A$ cu proprietatea $a \stackrel{+}{\Rightarrow}_\Gamma a$ se poate construi un şir care contrazice terminarea.

2⟶3. Rescrierea $\stackrel{*}{\Rightarrow}_\Gamma$ verifică condiţiile cerute.

3⟶4. Pentru $M = A$ şi ρ aplicaţia identică, observăm că relaţia $>$ verifică condiţiile cerute.

4⟶5. Prin inducţie structurală după contextul c. Observăm că elementele algebrei $\mathcal{A}$ nu sunt contexte.
Dacă $c = z$, deoarece $c[z \leftarrow a] = a$ şi $c[z \leftarrow d] = d$ concluzia rezultă uşor din ipoteză.

Dacă $c = A[z]_\sigma(a_1, a_2, \ldots, a_n)$ unde $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s'}$. Deoarece c este context rezultă că $n \geq 1$. În plus există un $i \in [n]$ astfel încâ a_i este context şi restul argumentelor lui $A[z]_\sigma$ sunt din A. Din ipoteza de inducţie deducem că $\rho(a_i[z \leftarrow a]) \vdash \rho(a_i[z \leftarrow d])$. Din ipoteză rezultă că

$$A_\sigma(a_1, \ldots, a_{i-1}, a_i[z \leftarrow a], a_{i+1}, \ldots, a_n) \models A_\sigma(a_1, \ldots, a_{i-1}, a_i[z \leftarrow d], a_{i+1}, \ldots, a_n).$$

Deoarece $c[z \leftarrow a] = A_\sigma(a_1, \ldots, a_{i-1}, a_i[z \leftarrow a], a_{i+1}, \ldots, a_n)$ şi $c[z \leftarrow d] = A_\sigma(a_1, \ldots, a_{i-1}, a_i[z \leftarrow d], a_{i+1}, \ldots, a_n)$ deducem că $c[z \leftarrow a] \models c[z \leftarrow d]$.

5⟶6. Presupunem că $a \Rightarrow_\Gamma d$. Prin urmare există $(\forall \mathcal{P}) l =_s r$ **if** $H \in \Gamma$, morfismul $h : \mathcal{P} \to \mathcal{A}$ şi contextul $c \in \mathcal{A}[z]_{s'}$ cu proprietăţile $h(H) \subseteq \downarrow_\Gamma$, $a = c[z \leftarrow h_s(l)]$ şi $d = c[z \leftarrow h_s(r)]$. În plus $a \neq d$, prin urmare $h_s(l) \neq h_s(r)$.

Din prima ipoteză deducem $\rho(h_s(l)) \vdash \rho(h_s(r))$. Aplicând a doua ipoteză rezultă că
$\rho(c[z \leftarrow h_s(l)]) \vdash \rho(c[z \leftarrow h_s(r)])$. Prin urmare $\rho(a) \vdash \rho(d)$.

6⟶1. Presupunrm prin absurd că $\overset{*}{\Rightarrow}_\Gamma$ nu are proprietatea de terminare. Prin urmare există şirul $\{a_n\}_n$ cu proprietatea $a_n \Rightarrow_\Gamma a_{n+1}$ pentru orice n natural. Aplicând funcţia ρ elementelor acestui şir obţinem un alt şir care contrazice noetherianitatea relaţiei $\vdash$. □

Se poate observa diferenţa esenţială dintre cele două ipoteze ale variantelor 4 sau 5 din teorema 1.1. Prima se referă la Γ rolul algebrei $\mathcal{A}$ fiind secundar. A doua ipoteza nu are nici o legătură cu Γ fiind de fapt o proprietate de compatibilitate între algebra $\mathcal{A}$ şi relaţia utilizată.

2 Exemplul teoriei grupurilor

Vom aplica teoria de mai sus pentru programul corespunzător teoriei grupurilor.

```
obj GRUP is sort E .
  op _*_ : E E -> E .
  op e : -> E .
  op _' : E -> E [prec 2].
  var A B C : E .
  eq (A*B)' = B' * A' .
  eq A'' = A .
  eq e' = e .
  eq (A*B)*C = A*(B*C) .
  eq e * A = A .
  eq A * e = A .
  eq A' * A = e .
  eq A * A' = e .
  eq A' * (A * B) = B .
  eq A * (A' * B) = B .
endo
```

Ca relaţie noetheriană vom folosi relaţia "strict mai mare" uzuală $>$ pe mulţimea numerelor naturale. Algebra pentru care facem demonstraţia este presupusă a fi o algebră liberă. Această ipoteza permite definirea funcţiei ρ este astfel:

$$\rho(e) = 2, \quad \rho(x) = 2 \text{ pentru orice variabilă } x$$

$$\rho(t') = 2^{\rho(t)} \text{ şi } \rho(t * s) = \rho(t)^2 \rho(s).$$

Observăm că $\rho(t) \geq 2$ pentru orice termen t.

Verificăm prima ipoteză a echivalenţelor 4 sau 5 din teorema 1.1.

1. $\rho(h((A * B)')) = \rho((h(A) * h(B))') = 2^{\rho(h(A))^2 \rho(h(B))}$
   $\rho(h(B' * A')) = \rho(h(B)' * h(A)') = (2^{\rho(h(B))})^2 2^{\rho(h(A))} = 2^{2\rho(h(B)) + \rho(h(A))}$
   Notând $x = \rho(h(A)) - 2$ şi $y = \rho(h(B)) - 2$ observăm că este suficient să demonstrăm că
   $(x+2)^2(y+2) > 2(y+2) + (x+2)$ oricare ar fi numerele naturale x, y, ceea ce este uşor.

2. $\rho(h(A'')) = 2^{2^{\rho(h(A))}} > \rho(h(A))$

3. $\rho(h(e')) = \rho(h(e)') = 2^2 = 4 > 2 = \rho(h(e))$
4. $\rho(h((A * B) * C)) = (\rho(h(A))^2\rho(h(B)))^2\rho(h(C)) = \rho(h(A))^4\rho(h(B))^2\rho(h(C)) > \rho(h(A))^2\rho(h(B))^2\rho(h(C)) = \rho(h(A * (B * C)))$
5. $\rho(h(e * A)) = 2^2\rho(h(A)) > \rho(h(A))$
6. $\rho(h(A * e)) = 2\rho(h(A))^2 > \rho(h(A))$
7. $\rho(h(A' * A)) = (2^{\rho(h(A))})^2\rho(h(A)) > 2 = \rho(h(e))$
8. $\rho(h(A * A')) = \rho(h(A))^2 2^{\rho(h(A))} > 2 = \rho(h(e))$
9. $\rho(h(A' * (A * B))) = (2^{\rho(h(A))})^2\rho(h(A))^2\rho(h(B)) > \rho(h(B))$
10. $\rho(h(A * (A' * B))) = \rho(h(A))^2(2^{\rho(h(A))})^2\rho(h(B)) > \rho(h(B))$

Verificarea ipotezei a doua din echivalenţa 4 este imediată.

3 Completarea Knuth-Bendix

Este vorba de o procedură care ne permite în anumite condiţii destul de restrictive să găsim un program pentru o prezentare ecuaţională dată. Condiţiile sunt restrictive deoarece se presupune existenţa unei relaţii care să verifice condiţia (b) din condiţiile echivalente 4 sau 5 din teorema 1.1. În aceste condiţii atenţia se va concentra asupra veridicităţii condiţiei (a) din aceleaşi condiţii echivalente.

3.1 Un exemplu

Vom începe cu un exemplu care va arăta cum se obţine programul pentru teoria grupurilor din specificaţia uzuală a grupurilor.

Specificaţia uzuală de la care plecăm este inclusă în programul de mai sus.

$$e * A = A \tag{1}$$

$$A' * A = e \tag{2}$$

$$(A * B) * C = A * (B * C) \tag{3}$$

În acest exemplu ne bazăm mai mult pe intuiţie, încercând să realizăm întâlnirea prin rescrierea perechilor critice. Căutăm cel mai general unificator u pentru membrul stâng l_2 al unei ecuaţii E_2 şi un subtermen d al unui membru stâng l_1 al unei ecuaţii E_1, posibil aceeaşi. Prin urmare există un context c astfel încât $l_1 = c[d]$. Apoi rescriem $u(l_1)$ la două forme normale n_1 şi n_2 începând cu

$$u(l_1) \Rightarrow u(r_1) \quad \text{şi} \quad u(l_1) = u^z(c)[u(l_2)] \Rightarrow u^z(c)[u(r_2)] = u(c[r_2])$$

după care adăugăm una dintre ecuaţiile $n_1 = n_2$ sau $n_2 = n_1$ cu intenţia de a asigura local confluenţa. Vom specula existenţa funcţiei ρ şi a relaţiei $>$ pe numere naturale din exemplul dat în secţiunea 2, pentru a alege între cele două egalităţi şi anume dacă $\rho(n_1) > \rho(n_2)$ alegem $n_1 \doteq n_2$ pentru a o adăuga celorlalte reguli.

Unificând membrul stâng al lui (2) cu subtermenul $A * B$ din membrul stâng al lui (3) şi calculând formele normale obţinem

$$(A' * A) * C \xrightarrow{2} e * C \xrightarrow{1} C$$

$$(A' * A) * C \xrightarrow{3} A' * (A * C)$$

prin urmare adăugăm regula

$$A' * (A * C) = C \tag{4}$$

Unificând membrul stâng al lui (1) cu subtermenul $A * C$ din membrul stâng al lui (4) şi calculând formele normale obţinem

$$e' * (e * A) \xrightarrow{4} A$$

$$e' * (e * A) \xrightarrow{1} e' * A$$

prin urmare adăugăm regula

$$e' * A = A \tag{5}$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (2) cu subtermenul $A * C$ din membrul stâng al lui (4) şi calculând formele normale obţinem

$$B'' * (B' * B) \xrightarrow{2} B'' * e$$

$$B'' * (B' * B) \xrightarrow{4} B$$

prin urmare adăugăm regula

$$B'' * e = B \tag{6}$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (6) cu subtermenul $A * B$ din membrul stâng al lui (3) şi calculând formele normale obţinem

$$(B'' * e) * A \xrightarrow{3} B'' * (e * A) \xrightarrow{1} B'' * A$$

$$(B'' * e) * A \xrightarrow{6} B * A$$

prin urmare adăugăm regula

$$B'' * A = B * A \tag{7}$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (6) cu membrul stâng al lui (7) şi calculând formele normale obţinem

$$B'' * e \xrightarrow{6} B$$

$$B'' * e \xrightarrow{7} B * e$$

prin urmare adăugăm regula

$$B * e = B \tag{8}$$

Unificând membrul stâng al lui (6) cu membrul stâng al lui (8) şi calculând formele normale obţinem

$$B'' * e \xrightarrow{6} B$$

$$B'' * e \xrightarrow{8} B''$$

prin urmare adăugăm regula

$$B'' = B \tag{9}$$

Mărind lista regulilor cu noi reguli de rescriere, este posibil ca unele reguli să devină inutile, fiind posibilă înlăturarea lor fără a pierde din forţa rescrierii. Concret, dacă se constată pentru o regulă $l_1 \doteq l_2$ că folosind celelalte reguli (fără $l_1 \doteq l_2$) l_1 şi l_2 pot fi rescrise în aceeaşi formă normală, atunci vom elimina regula $l_1 \doteq l_2$.

În acest moment regulile (6) şi (7) pot fi eliminate deoarece

$$B'' * e \xrightarrow{9} B * e \xrightarrow{8} B \quad \text{şi}$$

$$B'' * A \xrightarrow{9} B * A.$$

Lucrăm în continuare cu regulile (1,2,3,4,5,8 şi 9).

Unificând membrul stâng al lui (5) cu membrul stâng al lui (8) şi calculând formele normale obţinem

$$e' * e \xrightarrow{5} e$$

$$e' * e \xrightarrow{8} e'$$

prin urmare adăugăm regula

$$e' = e \tag{10}$$

În acest moment regula (5) poate fi eliminată deoarece

$$e' * A \xrightarrow{10} e * A \xrightarrow{1} A$$

Lucrăm în continuare cu regulile (1,2,3,4,8,9 şi 10).

Unificând membrul stâng al lui (9) cu subtermenul A' din membrul stâng al lui (2) şi calculând formele normale obţinem

$$B'' * B' \xrightarrow{2} e$$

$$B'' * B' \xrightarrow{9} B * B'$$

prin urmare adăugăm regula

$$B * B' = e \tag{11}$$

Unificând membrul stâng al lui (11) cu subtermenul $A * B$ din membrul stâng al lui (3) şi calculând formele normale obţinem

$$(A * A') * C \xrightarrow{3} A * (A' * C)$$

$$(A * A') * C \xrightarrow{11} e * C \xrightarrow{1} C$$

prin urmare adăugăm regula

$$A * (A' * C) = C \tag{12}$$

Unificând membrul stâng al lui (3) cu membrul stâng al lui (11) şi calculând formele normale obţinem

$$(A * B) * (A * B)' \xrightarrow{3} A * (B * (A * B)')$$

$$(A * B) * (A * B)' \xrightarrow{11} e$$

prin urmare adăugăm regula

$$A * (B * (A * B)') = e \tag{13}$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (13) cu subtermenul $A * C$ din membrul stâng al lui (4) şi calculând formele normale obţinem

$$A' * (A * (B * (A * B)')) \xrightarrow{13} A' * e \xrightarrow{8} A'$$

$$A' * (A * (B * (A * B)')) \xrightarrow{4} B * (A * B)'$$

prin urmare adăugăm regula

$$B * (A * B)' = A' \tag{14}$$

Mai târziu vom observa că această regulă este superfluă.

În acest moment regula (13) poate fi eliminată deoarece

$$A * (B * (A * B)') \xrightarrow{14} A * A' \xrightarrow{11} e$$

Lucrăm în continuare cu regulile (1,2,3,4,8,9,10,11,12 şi 14).

Unificând membrul stâng al lui (14) cu subtermenul $A * C$ din membrul stâng al lui (4) şi calculând formele normale obţinem

$$B' * (B * (A * B)') \xrightarrow{14} B' * A'$$

$$B' * (B * (A * B)') \xrightarrow{4} (A * B)'$$

prin urmare adăugăm regula

$$(A * B)' = B' * A' \tag{15}$$

În acest moment regula (14) poate fi eliminată deoarece

$$B * (A * B)' \xrightarrow{15} B * (B' * A') \xrightarrow{12} A'$$

Regulile rămase (1,2,3,4,8,9,10,11,12 şi 15) coincid cu programul prezentat în secţiunea 2.

Despre problema punerii celor două expresii în membrul stâng, respectiv drept, al noilor ecuaţii am putea spune că am plasat expresia mai complexă în membrul stâng pentru ca regula să conducă de la complex la simplu. Dar oare ce înseamnă mai simplu sau mai complex?

De fapt le-am pus ca în programul din secţiunea 2 ca să dăm din nou peste el. Dar nici acesta nu este un criteriu, deoarece procedura are chiar rolul de a crea un program pe care nu-l cunoaştem.

Singura motivaţie corectă este că alegerea este făcută pentru ca regula creată să verifice condiţia necesară pentru demonstrarea terminării, fapt dat de funcţia ρ şi relaţia $>$ pentru numere naturale.

3.2 Procedura Knuth-Bendix, varianta redusă

Toate ecuaţiile utilizate în aceasta secţiune sunt necondiţionate.

Procedura Knuth-Bendix se aplică unei specificaţii ecuaţionale E. Vom prezenta la început, din motive didactice, o variantă incompletă.

Procedura lucrează cu o mulţime de ecuaţii şi o mulţime de reguli. Iniţial $E_0 = E$ şi $\Gamma_0 = \emptyset$.

La fiecare pas se alege o ecuaţie care la terminarea pasului se va elimina din mulţimea ecuaţiilor. Fiecare termen al ecuaţiei se aduce la o formă normală folosind mulţimea regulilor. Vom considera două cazuri.

A) Dacă cele două forme normale coincid pasul constă doar în eliminarea ecuaţiei.

B) Dacă cele două forme normale sunt diferite, ele vor alcătui o nouă regulă care se adaugă mulţimii regulilor. Din noua regulă şi din noua mulţime de reguli se calculează toate perechile critice posibile care se adaugă la mulţimea ecuaţiilor.

Dacă procedura se termină prin epuizarea mulţimii ecuaţiilor, mulţimea finală de reguli este local confluentă şi o specificaţie echivalentă cu cea iniţială.

Observaţia 3.1 *Dacă $\Gamma_1 \subseteq \Gamma$, atunci $\equiv^{\mathcal{A}}_{\Gamma_1} \subseteq \equiv^{\mathcal{A}}_{\Gamma}$ pentru orice algebră $\mathcal{A}$.*

Demonstraţie: Fie $a \equiv^{\mathcal{A}}_{\Gamma_1} b$.

Fie $h : \mathcal{A} \to \mathcal{B} \models \Gamma$. Deoarece $\mathcal{B} \models \Gamma_1$ deducem $h(a) = h(b)$. Deci $a \equiv^{\mathcal{A}}_{\Gamma} b$. □

Pentru două mulţimi de ecuaţii necondiţionate vom folosi notaţia

$$\Gamma_1 \ \subseteq \equiv_\Gamma$$

cu semnificaţia

$$u \equiv^{T_\Sigma(X)}_{\Gamma} v \text{ pentru orice } (\forall X) u = v \in \Gamma_1$$

Observaţia 3.2 $\Gamma \ \subseteq \equiv_\Gamma$

Propoziţie 3.3 *Dacă $\Gamma_1 \ \subseteq \equiv_\Gamma$, atunci*

1) Orice Γ-algebră este Γ_1-algebră şi

2) $\equiv^{\mathcal{A}}_{\Gamma_1} \subseteq \equiv^{\mathcal{A}}_{\Gamma}$.

Demonstraţie: 1) Presupunem $\mathcal{A} \models \Gamma$.

Fie $(\forall X) u = v \in \Gamma_1$ şi $h : T_\Sigma(X) \longrightarrow \mathcal{A}$. Deoarece $u \equiv^{T_\Sigma(X)}_{\Gamma} v$ şi $h : T_\Sigma(X) \longrightarrow \mathcal{A} \models \Gamma$ deducem $h(u) = h(v)$.

2) Presupunem în Σ-algebra $\mathcal{A}$ că $a \equiv_{\Gamma_1} d$ şi probăm că $a \equiv_\Gamma d$. Fie $h : \mathcal{A} \longrightarrow \mathcal{M} \models \Gamma$. Prin urmare $h : \mathcal{A} \longrightarrow \mathcal{M} \models \Gamma_1$, deci din $a \equiv_{\Gamma_1} d$ deducem $h(a) = h(d)$. □

Menţionăm utilitatea concluziei a doua. Pentru a demonstra egalitatea $\equiv_\Gamma = \equiv_{\Gamma_1}$ este suficient sa probăm incluziunile $\Gamma \ \subseteq \equiv_{\Gamma_1}$ şi $\Gamma_1 \ \subseteq \equiv_\Gamma$.

Observaţia 3.4 *Dacă (w, v) este o pereche critică pentru Γ, atunci $w \equiv_\Gamma v$.*

Demonstraţie: Presupunem $w = u(r_1)$ şi $v = u(c[z \leftarrow r_2])$ unde $(l_1 = c[z \leftarrow d], r_1)$ şi (l_2, r_2) sunt reguli cu mulţimi de variabile disjuncte din Γ şi $u = cgu(d, l_2)$. Deoarece $u(l_1) \Rightarrow_\Gamma w$ şi $u(l_1) \Rightarrow_\Gamma v$ din corectitudinea relaţiei $\overset{*}{\Rightarrow}_\Gamma$ deducem $w \equiv_\Gamma v$. □

Propoziţie 3.5 *Echivalenţa semantică asociată reuniunii mulţimii ecuaţiilor cu mulţimea regulilor este un invariant al procedurii.*

Demonstraţie: Să presupunem că la începutul pasului mulţimea de ecuaţii este $E \cup \{(\forall X) l = r\}$ cu ecuaţia aleasă $(\forall X) l = r$ şi mulţimea de reguli Γ. Fie l' forma normală a lui l şi r' forma normală a lui r.

În cazul A) $l \downarrow_\Gamma r$ implică $l \downarrow_{E \cup \Gamma} r$, prin urmare din corectitudinea relaţiei $\downarrow_{E\cup\Gamma}$ deducem $l \equiv_{E\cup\Gamma} r$. Prin urmare $E \cup \{(\forall X) l = r\} \cup \Gamma \ \subseteq \equiv_{E\cup\Gamma}$ deci din propoziţia 3.3 deducem $\equiv_{E\cup\{(\forall X) l=r\}\cup\Gamma} \subseteq \equiv_{E\cup\Gamma}$. Incluziunea contrară fiind evidentă obţinem concluzia.

Trecem la cazul B). Fie P mulţimea perechilor critice dintre $l' = r'$ şi $\Gamma \cup \{l' = r'\}$. Din observaţia 3.4 deducem $P \subseteq \equiv_{\Gamma\cup\{l'=r'\}}$.

Din $l \overset{*}{\Rightarrow}_\Gamma l'$ şi $r \overset{*}{\Rightarrow}_\Gamma r'$ deducem $l \equiv_\Gamma l'$ şi $r \equiv_\Gamma r'$. Folosind $l \equiv_{\Gamma\cup(\forall X) l=r} r$ şi $l' \equiv_{\Gamma\cup(\forall X) l'=r'} r'$ deducem $l \equiv_{\Gamma\cup(\forall X) l'=r'} r$ şi $l' \equiv_{\Gamma\cup(\forall X) l=r} r'$, prin urmare

$$l \equiv_{E\cup P\cup\Gamma\cup(\forall X) l'=r'} r \text{ şi } l' \equiv_{E\cup\Gamma\cup(\forall X) l=r} r'.$$

Din $l' \equiv_{\Gamma \cup (\forall X) l=r} r'$ deducem $\Gamma \cup (\forall X) l' = r' \subseteq \equiv_{\Gamma \cup (\forall X) l=r}$, aşadar $\equiv_{\Gamma \cup (\forall X) l'=r'} \subseteq \equiv_{\Gamma \cup (\forall X) l=r}$, deci

$$P \subseteq \equiv_{E \cup \Gamma \cup (\forall X) l=r} .$$

Din cele de mai sus deducem

$$E \cup P \cup \Gamma \cup (\forall X) l' = r' \subseteq \equiv_{E \cup \Gamma \cup (\forall X) l=r} \text{ şi}$$

$$E \cup \Gamma \cup (\forall X) l = r \subseteq \equiv_{E \cup P \cup \Gamma \cup (\forall X) l'=r'},$$

deci

$$\equiv_{E \cup \Gamma \cup (\forall X) l=r} = \equiv_{E \cup P \cup \Gamma \cup (\forall X) l'=r'} . \square$$

Propoziţie 3.6 *Dacă procedura se termină, mulţimea finală de reguli generează o rescrire local confluentă.*

Demonstraţie:

Observăm că pe parcursul algoritmului mulţimea regulilor devine din ce în ce mai mare, fără a fi scoase reguli din această mulţime. Prin urmare orice rescriere posibilă într-un anumit moment al algoritmului rămâne posibilă până la terminarea algoritmului.

Conform teoremei din lecţia precedentă este suficient să probăm că mulţimea perechilor critice este inclusă în relaţia de întâlnire prin rescriere.

Fie (a, b) o pereche critică pentru mulţimea finală de reguli. Din pasul B al algoritmului rezultă că perechea critică (a, b) este adăugată mulţimii ecuaţiilor. Pentru ca ecuaţia $a \doteq b$ să dispară din mulţimea ecuaţiilor se calculează $fn(a)$ forma normală a lui a şi $fn(b)$ forma normală a lui b. Dacă $fn(a) = fn(b)$ rezultă că a şi b se întâlnesc prin rescriere. În caz contrar se adaugă noua regulă $fn(a) \doteq fn(b)$ sau $fn(a) \doteq fn(b)$ ceea ce face ca a şi b să se întâlnească prin rescriere.

Pentru a obţine concluzia este suficient să mai aplicăm teorema perechilor critice pentru cazul ecuaţional. $\square$

3.3 Procedura Knuth-Bendix

Forma completă a procedurii Knuth-Bendix se obţine folosind un criteriu care să asigure terminarea programului final. Găsirea acestui criteriu este o problemă dificilă. Ipoteza care acceptă acest criteriu este partea slabă a acestei proceduri.

Criteriul este folosit numai în cazul B al procedurii în momentul adăugării unei noi reguli. Cele două forme normale diferite pot forma o regulă în două moduri diferite. Criteriul este folosit numai pentru alegerea uneia dintre cele două variante. Dacă criteriul nu poate alege, atunci procedura se blochează.

După cum am văzut mai sus procedura lucrează numai cu termeni, prin urmare criterul se referă numai la algebra $T_\Sigma(X)$ a termenilor.

Criteriul A: există o relaţie de ordine compatibilă cu structura algebrică, a cărei variantă strictă este noetheriană.

Criteriul B: există o relaţie noetheriană $\vdash$ pe o mulţime M şi o funcţie $\rho \colon A \longrightarrow M$ cu proprietatea pentru orice $n \geq 1$ şi $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$,

$$\rho(a) \vdash \rho(d) \text{ implică } A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) \models A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n),$$

Aplicarea criteriului la alegerea regulilor face posibilă aplicarea teoremei 1.1 programului final. Se observă deci că programul final se termină.

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 işi 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 $\psi;\widehat{} = h$.

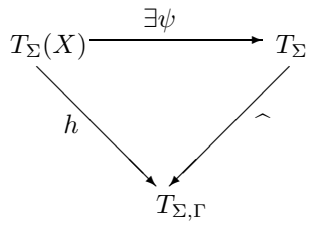


Figure 3.1: Proiectivitatea Σ-algebrei $T_\Sigma(X)$

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. □

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 $h(a) \equiv_\Gamma^{\mathcal{B}} h(b)$. □

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, $\Gamma \models (\forall \mathcal{B})h(G)$. □

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 $T_\Sigma(X)$. □

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 $u = \theta; u_{|_X}$. □

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. □

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, $(\forall x_1\ \forall x_2 \dots \forall x_n)f(x_1, x_2, \dots, x_n) \doteq f(x_1, x_2, \dots, x_n) \in \Gamma$ 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 $(\forall x_1\ \forall x_2 \dots \forall x_n)f(x_1, x_2, \dots, x_n) \doteq f(x_1, x_2, \dots, x_n)$.**

Vom demonstra acest lucru prin inducţie după structura termenului t.

Primul pas al inducţiei este chiar (1).

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

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

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

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

Mai observăm că

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

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

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

Fie $h : T_\Sigma(X) \to T_\Sigma(Y)$. Cum $var(G) = \{x_1, x_2, \dots, x_n\} \subseteq X$, 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 $1 \leq i \leq n$:

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

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

$$G' \to_p^* (z_1 \leftarrow h(x_1))(G) \to_p^* \dots \to_p^* (z_n \leftarrow h(x_n))(\dots(z_1 \leftarrow h(x_1))(G)\dots) = h(G).$$ □

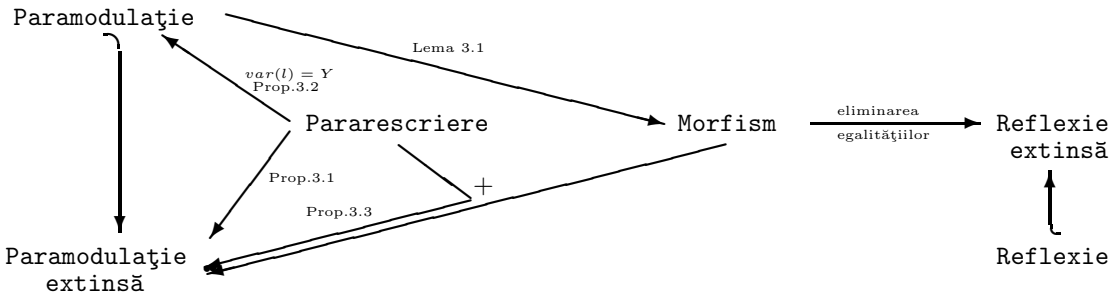


Figure 4.1: Legăturile dintre regulile de deducţie

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

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. □

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

Fie $S : T_\Sigma(X) \rightarrow \mathcal{B}$ 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 $S : T_\Sigma(X) \rightarrow \mathcal{B}$ 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 $h : \mathcal{B} \rightarrow \mathcal{M} \models \Gamma$. 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 $\theta;S;h : T_\Sigma(Y) \rightarrow \mathcal{M}$ şi relaţia (3), deducem că

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

Probăm că

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

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 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 $\{a \doteq_s d\} \stackrel{*}{\rightarrow}_{pr} \Delta$.*

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 $a \stackrel{*}{\rightarrow}_Q v$ şi $d \stackrel{*}{\rightarrow}_Q v$. Deoarece Q este reuniunea şirului crescător $\{Q_n\}_{n \in N}$, rezultă că există un număr natural n cu proprietatea $a \stackrel{*}{\rightarrow}_{Q_n} v$ şi $d \stackrel{*}{\rightarrow}_{Q_n} v$, deci $a \downarrow_{Q_n} d$.

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

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

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

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 $G \stackrel{*}{\rightarrow}_p \Delta$ 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 $\sigma(G) \stackrel{*}{\rightarrow}_{pr} \Delta$.

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 $\sigma(G) \stackrel{*}{\rightarrow}_p \Delta$ cu substituţia calculată identitatea.

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

Deci $G \stackrel{*}{\rightarrow}_p \Delta$ 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 $fn(a) = fn(d)$. □

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 $i_X; \theta$.□

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. □

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'.

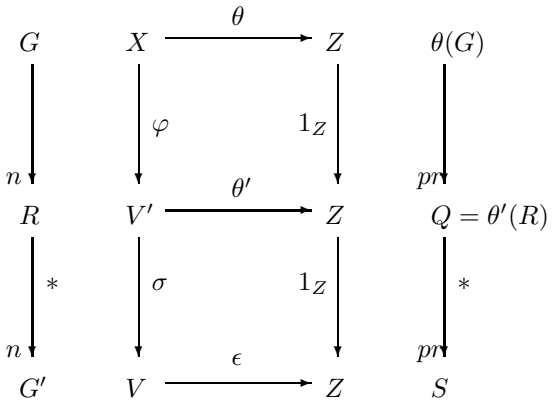
Notând cu $\varphi : T_\Sigma(X) \to T_\Sigma(V)$ restricţia lui ψ la $T_\Sigma(X)$ deducem că $\theta = \varphi\theta'$.

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:

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



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 $(\varphi\sigma)\epsilon = \varphi\theta' = \theta$. □

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 $h = u; v = (u; w); f$. □

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 $(\forall Y)\,\pi(v)\,\text{if H}$ 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 $(\forall Y)\,\pi(v) \doteq_b t\,\text{if H}^\text{e}$.
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:

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

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 $(\forall Y)\,\pi(v) \doteq_b t\,\text{if H}^\text{e}$ 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 $G_1 = F^e \cup \{t \doteq_b t\}$. □

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 $w = s\ s\ y'$.

Logica Clauzelor Horn Cu Egalitate

Virgil Emil Căzănescu

January 25, 2003

O *signatură multisortată de ordinul întâi* este un triplet (S, Σ, Π) unde (S, Σ) este o signatură multisortată şi $\Pi = (\Pi_u : u \in S^*)$ este o familie de mulţimi S^*-indexate.

Un Σ-Π-model $\mathcal{M} = (M_s, M_\sigma, M_\pi)$ este un triplet unde (M_s, M_σ) este o Σ-algebră şi $\{M_\pi\}_{\pi \in \Pi}$ este o familie de mulţimi Π-indexate astfel încât $M_\pi \subseteq M_{s_1} \times M_{s_2} \times \ldots \times M_{s_n}$ pentru fiecare $\pi \in \Pi_{s_1 s_2 \ldots s_n}$.

Un Σ-Π-morfism (sau un morfism de modele) $h : (M_s, M_\sigma, M_\pi) \longrightarrow (N_s, N_\sigma, N_\pi)$ este un morfism de Σ-algebre $h = \{h_s\}_{s \in S} : (M_s, M_\sigma) \longrightarrow (N_s, N_\sigma)$ astfel încât $(m_1, m_2, \ldots, m_k) \in M_\pi$ implică $(h_{s_1}(m_1), h_{s_2}(m_2), \ldots, h_{s_k}(m_k)) \in N_\pi$ pentru toţi $\pi \in \Pi_{s_1 s_2 \ldots s_k}$ şi $m_i \in M_{s_i}$ pentru $1 \leq i \leq k$.

Fie $Mod_{\Sigma,\Pi}$ categoria tuturor Σ-Π-modelelor şi morfismelor lor. Fie

$$U : Mod_{\Sigma,\Pi} \longrightarrow Alg_\Sigma$$

functorul uituc.

1 Formule atomice

O *formulă atomică* într-o Σ-algebră $\mathcal{A} = (A_s, \sigma_A)$ este $a =_s b$ unde $a, b \in A_s$ sau $\pi(a_1, a_2, \ldots, a_n)$ unde $\pi \in \Pi_{s_1 s_2 \ldots s_n}$ şi $a_i \in A_{s_i}$ pentru toţi $1 \leq i \leq n$. O Σ-propoziţie din această logică este $(\forall \mathcal{A})B$ unde $\mathcal{A}$ este o Σ-algebră şi B este o formulă atomică în $\mathcal{A}$.

Definim functorul $FA : Alg_\Sigma \longrightarrow \mathbf{Set}$ prin:

a) $FA(\mathcal{A})$ este mulţimea tuturor formulelor atomice din Σ-algebra $\mathcal{A}$, adică

$$FA(\mathcal{A}) = \{a =_s b | a, b \in A_s, s \in S\} \cup \{\pi(a_1, \ldots, a_n) \mid \pi \in \Pi_{s_1 \ldots s_n}; a_i \in A_{s_i}\}$$

b) pentru fiecare Σ-morfism $h : \mathcal{A} \longrightarrow \mathcal{B}$
$FA(h)(\pi(a_1, a_2, \ldots, a_n)) = \pi(h(a_1), h(a_2), \ldots, h(a_n))$ pentru $\pi \in \Pi_{s_1 s_2 \ldots s_n}$ şi $a_i \in A_{s_i}$, $FA(h)(a =_s b) = h_s(a) =_s h_s(b)$ pentru $a, b \in A_s$.

Vom demonstra că FA este functor. Fie $f \in Alg_\Sigma(\mathcal{A}, \mathcal{B})$ şi $g \in Alg_\Sigma(\mathcal{B}, \mathcal{C})$.

Avem $FA(f) \in \mathbf{Set}(FA(\mathcal{A}), FA(\mathcal{B}))$.

Vom demonstra proprietatea de functorialitate:

$$FA(f;g)(a =_s b) = (f;g)_s(a) =_s (f;g)_s(b) = g_s(f_s(a)) =_s g_s(f_s(b)) =$$

$$= FA(g)(f_s(a) =_s f_s(b)) = FA(g)(FA(f)(a =_s b)) = (FA(f); FA(g))(a =_s b).$$

Pentru $FA(f;g)(\pi(a_1,\ldots,a_n))$ avem

$$FA(f;g)(\pi(a_1,\ldots,a_n)) = \pi((f;g)_{s_1}(a_1),\ldots,(f;g)_{s_n}(a_n)) =$$
$$= \pi(g_{s_1}(f_{s_1}(a_1)),\ldots,g_{s_n}(f_{s_n}(a_n))) = FA(g)(\pi(f_{s_1}(a_1),\ldots,f_{s_n}(a_n))) =$$
$$= FA(g)(FA(f)(\pi(a_1,\ldots,a_n)) = (FA(f);FA(g))(\pi(a_1,\ldots,a_n)),$$

pentru orice $\pi \in \Pi_{s_1,\ldots,s_n}; a_i \in A_{s_i}$, pentru $1 \leq i \leq k$.

Deci $FA(f;g) = (FA(f);FA(g))$.

Vom arăta că $FA(1_{\mathcal{A}}) = 1_{FA(\mathcal{A})}$.

$$FA(1_{\mathcal{A}})(a =_s b) = (1_{A_s}(a) =_s 1_{A_s}(b)) = (a =_s b) = 1_{FA(\mathcal{A})}(a =_s b).$$
$$FA(1_{\mathcal{A}})(\pi(a_1,\ldots,a_n)) = \pi(1_{A_{s_1}}(a_1),\ldots,1_{A_{s_n}}(a_n)) = \pi(a_1,\ldots,a_n) =$$
$$1_{FA(\mathcal{A})}(\pi(a_1,\ldots,a_n)).$$

În concluzie, FA este functor.

2 Semantică

Fie $\mathcal{A}$ o Σ-algebră, $\mathcal{M}$ un Σ-Π-model şi $h\colon \mathcal{A} \longrightarrow U(\mathcal{M})$ un Σ-morfism. Definim

$Kr(h,\mathcal{M}) = \{a =_s b \mid s \in S,\ a,b \in A_s,\ h_s(a) = h_s(b)\} \cup$
$\cup\{\pi(a_1,\ldots,a_n) \mid \pi \in \Pi_{s_1\ldots s_n}; a_i \in A_{s_i},\ (h_{s_1}(a_1),\ldots,h_{s_n}(a_n)) \in M_\pi\}$

Spunem că Σ-Π-modelul $\mathcal{M}$ satisface Σ-propoziţia $(\forall\mathcal{A})B$ şi scriem $\mathcal{M} \models_\Sigma (\forall\mathcal{A})B$ dacă şi numai dacă $B \in Kr(h,\mathcal{M})$ pentru fiecare Σ-morfism $h\colon \mathcal{A} \longrightarrow U(\mathcal{M})$.

O Σ-Π-*clauză* este $(\forall\mathcal{P})B$ **if** C unde $\mathcal{P}$ este o Σ-algebră liberă, B este o formulă atomică din $\mathcal{P}$ şi C este o mulţime finită de formule atomice din $\mathcal{P}$.

Spunem că Σ-Π-modelul $\mathcal{M}$ satisface clauza $(\forall\mathcal{P})B$ **if** C şi scriem

$$\mathcal{M} \models_\Sigma (\forall\mathcal{P})B \textbf{ if } C$$

dacă şi numai dacă pentru orice Σ-morfism $h\colon \mathcal{P} \longrightarrow U(\mathcal{M})$, $C \subseteq Kr(h,\mathcal{M})$ implică $B \in Kr(h,\mathcal{M})$.

Fie Γ o mulţime de clauze. Spunem că modelul $\mathcal{M}$ satisface Γ sau că $\mathcal{M}$ este un Γ-model şi scriem $\mathcal{M} \models_\Sigma \Gamma$ dacă şi numai dacă $\mathcal{M}$ satisface fiecare clauză din Γ.
Subcategoria plină a tuturor Γ-modelelor este notată cu Mod_Γ.

Scriem că $\Gamma \models (\forall\mathcal{A})B$ dacă şi numai dacă $\mathcal{M} \models_\Sigma (\forall\mathcal{A})B$ pentru fiecare Γ-model $\mathcal{M}$.

Observăm că

$\Gamma \models (\forall\mathcal{A})B \Longleftrightarrow B \in Kr(g,\mathcal{M})$ pentru toate Γ-modelele $\mathcal{M}$ şi toate $g\colon \mathcal{A} \longrightarrow U(\mathcal{M})$.

3 Deducţie locală

Fixăm Σ-algebra $\mathcal{A}$. Vom arăta cum transformarea regulilor clasice de deducţie din logica clauzelor Horn cu egalitate în structura noastră ne oferă o mulţime de reguli de deducţie corecte şi complete pentru mulţimea formulelor atomice din $\mathcal{A}$ care sunt satisfăcute de fiecare Γ-model. Mulţimea de propoziţii este în cele ce urmează mulţimea formulelor atomice din Σ-algebra $\mathcal{A} = (A_s, \sigma_A)$.

Regulile de deducţie **HL** pentru logica clauzelor Horn sunt:

R	$a =_s a$
S	$a =_s b$ implică $b =_s a$
T	$a =_s b$ şi $b =_s c$ implică $a =_s c$
CΣ	$a_i =_{s_i} b_i$ pentru $1 \leq i \leq n$ implică $\sigma_A(a_1, a_2, \ldots, a_n) =_s \sigma_A(b_1, b_2, \ldots, b_n)$ pentru fiecare $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$.
CΠ	$a_i =_{s_i} b_i$ pentru $1 \leq i \leq n$ şi $\pi(a_1, a_2, \ldots, a_n)$ implică $\pi(b_1, b_2, \ldots, b_n)$ pentru fiecare $\pi \in \Pi_{s_1 s_2 \ldots s_n}$.
MP	$FA(h)(B')$ pentru orice B' din C implică $FA(h)(B)$ pentru orice $(\forall \mathcal{P})B$ **if** $C \in \Gamma$ şi orice Σ-morfism $h\colon \mathcal{P} \longrightarrow \mathcal{A}$.

Pentru fiecare formulă atomică B din $\mathcal{A}$ scriem $\Gamma \vdash (\forall \mathcal{A})B$ dacă B poate fi demonstrată folosind mulţimea de reguli de mai sus.

4 Corectitudine

Lemă 4.1 *Considerăm morfismele $h : \mathcal{A} \longrightarrow \mathcal{B}$ şi $g : \mathcal{B} \longrightarrow U(\mathcal{M})$. Pentru orice B din $FA(\mathcal{A})$ avem*

$$FA(h)(B) \in Kr(g, \mathcal{M}) \text{ dacă şi numai dacă } B \in Kr(h; g, \mathcal{M}).$$

Demonstraţie: B poate fi de forma $a =_s b$ sau $\pi(a_1, \ldots, a_n)$.

În primul caz $FA(h)(a =_s b) \in Kr(g, \mathcal{M})$ dacă şi numai dacă $h_s(a) = h_s(b) \in Kr(g, \mathcal{M})$ dacă şi numai dacă $(h; g)_s(a) = (h; g)_s(b)$ dacă şi numai dacă $a =_s b \in Kr(h; g, \mathcal{M})$.

Dacă B este de forma $\pi(a_1, \ldots, a_n)$, vom avea $FA(h)(\pi(a_1, \ldots, a_n)) \in Kr(g, \mathcal{M})$ dacă şi numai dacă $\pi(h_{s_1}(a_1), \ldots, h_{s_n}(a_n)) \in Kr(g, \mathcal{M})$ adică $((h; g)_{s_1}(a_1), \ldots, (h; g)_{s_1}(a_n)) \in M_\pi$ dacă şi numai dacă $\pi(a_1, \ldots, a_n) \in Kr((h; g), \mathcal{M})$.

Deci, am demonstrat că pentru orice $B \in FA(\mathcal{A})$ avem $FA(h)(B) \in Kr(g, \mathcal{M})$ dacă şi numai dacă $B \in Kr((h; g), \mathcal{M})$. □

Prima parte a propoziţiei următoare arată că regulile **HL** sunt corecte pentru mulţimea formulelor atomice din $\mathcal{A}$ care sunt satisfăcute în fiecare Γ-model.

Propoziţie 4.2 *Fie $\mathcal{M}$ un model şi $g\colon \mathcal{A} \longrightarrow U(\mathcal{M})$ un Σ-morfism.*
$Kr(g, \mathcal{M})$ este închisă la **R**, **S**, **T**, **C**Σ *şi* **C**Π. *Mai mult chiar,*

a) Dacă $\mathcal{M} \models_\Sigma \Gamma$ atunci $Kr(g, \mathcal{M})$ este închisă la **MP** *şi*

b) Dacă g este surjectivă pe componente şi $Kr(g, \mathcal{M})$ este închisă la **MP** *atunci $\mathcal{M} \models_\Sigma \Gamma$.*

Demonstraţie: Mai întâi vom demonstra că $Kr(g, \mathcal{M})$ este închisă la **R**, **S**, **T**, **C**Σ şi **C**Π.

R : $a =_s a \in Kr(g, \mathcal{M})$ dacă şi numai dacă $g_s(a) = g_s(a)$.

S : Presupunem că $a =_s b \in Kr(g, \mathcal{M})$ şi demonstrăm că $b =_s a \in Kr(g, \mathcal{M})$. Din ipoteză deducem $g_s(a) = g_s(b)$, prin urmare $g_s(b) = g_s(a)$ deci $b =_s a \in Kr(g, \mathcal{M})$.

T : Presupunem că $a =_s b \in Kr(g, \mathcal{M})$ şi $b =_s c \in Kr(g, \mathcal{M})$ şi demonstrăm că $a =_s c \in Kr(g, \mathcal{M})$. Din ipoteză deducem $g_s(a) = g_s(b)$ şi $g_s(b) = g_s(c)$, prin urmare $g_s(a) = g_s(c)$ deci $a =_s c \in Kr(g, \mathcal{M})$.

CΣ : Pentru $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ presupunem că $a_i =_{s_i} b_i \in Kr(g, \mathcal{M})$ pentru $1 \leq i \leq n$ deci $g_{s_i}(a_i) = g_{s_i}(b_i)$. Rezultă că

$$g_s(\sigma_{\mathcal{A}}(a_1, \ldots, a_n)) = \sigma_{\mathcal{A}}(g_{s_1}(a_1), \ldots, g_{s_n}(a_n)) = \sigma_{\mathcal{A}}(g_{s_1}(b_1), \ldots g_{s_n}(b_n)) = g_s(\sigma_{\mathcal{A}}(b_1, \ldots, b_n)),$$

deci $\sigma_{\mathcal{A}}(a_1, \ldots, a_n) = \sigma_{\mathcal{A}}(b_1, \ldots, b_n) \in Kr(g, \mathcal{M})$.

CΠ : Pentru $\pi \in \Pi_{s_1 s_2 \ldots s_n}$ presupunem că $a_i =_{s_i} b_i \in Kr(g, \mathcal{M})$ pentru $1 \leq i \leq n$ şi $\pi(a_1, \ldots, a_n) \in Kr(g, \mathcal{M})$. Prin urmare $g_{s_i}(a_i) = g_{s_i}(b_i)$ pentru $1 \leq i \leq n$ şi

$$(g_{s_1}(a_1), \ldots, g_{s_n}(a_n)) \in M_\pi.$$

Rezultă că $(g_{s_1}(b_1), \ldots, g_{s_n}(b_n)) \in M_\pi$ adică $\pi(b_1, \ldots, b_n) \in Kr(g, \mathcal{M})$.

Deci, în concluzie, dacă $\pi \in \Pi_{s_1 s_2 \ldots s_n}$; $a_i =_{s_i} b_i \in Kr(g, \mathcal{M})$ pentru $1 \leq i \leq n$, şi $\pi(a_1, \ldots, a_n) \in Kr(g, \mathcal{M})$ deducem $\pi(b_1, \ldots, b_n) \in Kr(g, \mathcal{M})$.

Deci multimea $Kr(g, \mathcal{M})$ este închisă la **R**, **S**, **T**, **C**Σ şi **C**Π.

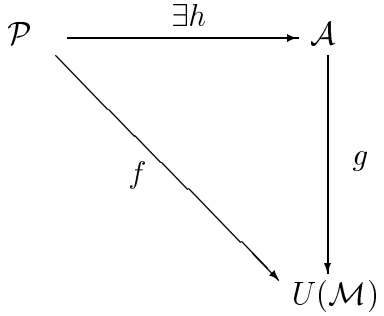
a) **MP** : Pentru $(\forall \mathcal{P}) B$ **if** $C \in \Gamma$ şi $h \colon \mathcal{P} \longrightarrow \mathcal{A}$ un Σ-morfism presupunem pentru orice B' din C că $FA(h)(B') \in Kr(g, \mathcal{M})$. Din lema precedentă rezultă pentru orice B' din C că $B' \in Kr(h; g, \mathcal{M})$. Deoarece $\mathcal{M} \models_\Sigma \Gamma$ rezultă că $B \in Kr(h; g, \mathcal{M})$, deci conform lemei $FA(h)(B) \in Kr(g, \mathcal{M})$.

Deci $Kr(g, \mathcal{M})$ este închisă la **MP** dacă $\mathcal{M} \models_\Sigma \Gamma$.

b) Presupunem că g este surjectiv pe componente şi $Kr(g, \mathcal{M})$ este închisă la **MP**. Va trebui să arătăm că $\mathcal{M} \models_\Sigma \Gamma$.

Fie $(\forall \mathcal{P}) B'$ **if** $C \in \Gamma$ şi $f \colon \mathcal{P} \longrightarrow U(\mathcal{M})$ cu proprietatea $C \subseteq Kr(f, M)$. Va trebui să demonstrăm că $B \in Kr(f, M)$.

Deoarece $\mathcal{P}$ este algebră proiectivă şi g este surjectiv pe componente $\Rightarrow \exists h \colon \mathcal{P} \longrightarrow \mathcal{A}$ astfel încât $h; g = f$.



Din $C \subseteq Kr(h; g, M)$ deducem din lemă că $(FA(h))(B') \in Kr(g, \mathcal{M})$ pentru orice $B' \in C$. Deoarece $Kr(g, \mathcal{M})$ este închis la **MP** rezultă că $(FA(h))(B) \in Kr(g, \mathcal{M})$, prin urmare conform lemei $B \in Kr(h; g, \mathcal{M})$, adică $B \in Kr(f, \mathcal{M})$. $\square$

Corolar 4.3 *Dacă* $\Gamma \vdash e$ *atunci* $\Gamma \models e$ *pentru orice* $e \in FA(\mathcal{A})$.

Demonstraţie: Presupunem $\Gamma \vdash (\forall \mathcal{A})B$, adică B e demonstrabilă folosind cele şase reguli **HL**. Pentru orice Γ-model $\mathcal{M}$ şi orice morfism $h: \mathcal{A} \longrightarrow U(\mathcal{M})$ mulţimea $Kr(h, \mathcal{M})$ este închisă la cele şase reguli **HL**. Prin urmare

$$\bigcap\{Kr(h, \mathcal{M}) | \mathcal{M} \models \Gamma, h: \mathcal{A} \longrightarrow U(\mathcal{M})\}$$

este închisă la cele şase reguli **HL**. Deoarece mulţimea formulelor demonstrabile în **HL** este cea mai mică mulţime închisă la cele şase reguli **HL** rezultă că

$$B \in \bigcap\{Kr(h, \mathcal{M}) | \mathcal{M} \models \Gamma, h: \mathcal{A} \longrightarrow U(\mathcal{M})\}$$

deci $\Gamma \models (\forall \mathcal{A})B$.

5 Completitudine

Deoarece mulţimea teoremelor din $\mathcal{A}$ este închisă la **R**, **S**, **T** şi **C**Σ putem defini o relaţie de congruenţă $=_\Gamma$ în $\mathcal{A}$ pentru fiecare $s \in S$ şi $a, b \in A_s$ prin

$$a =_\Gamma b \Longleftrightarrow \Gamma \vdash a =_s b.$$

Fie $\mathcal{A}/_{=_\Gamma}$ factorizarea lui $\mathcal{A}$ prin $=_\Gamma$ şi $q: \mathcal{A} \longrightarrow \mathcal{A}/_{=_\Gamma}$ morfismul de factorizare.

Modelul $\mathcal{A}_\Gamma = (\mathcal{A}/_{=_\Gamma}, \{P_\Gamma\}_{P \in \Pi})$ este definit pentru fiecare $P \in \Pi_{s_1 s_2 \ldots s_n}$ prin

$$P_\Gamma = \{(q(a_1), \ldots, q(a_n)) : \vdash P(a_1, \ldots, a_n)\}.$$

Afirmăm că mulţimea $Kr(q, \mathcal{A}_\Gamma)$ este egală cu mulţimea teoremelor.

Avem $a =_s b \in Kr(q, \mathcal{A}_\Gamma) \Leftrightarrow q_s(a) = q_s(b) \Leftrightarrow a =_\Gamma b \Leftrightarrow \vdash a =_s b$.

La fel $P(a_1, \ldots, a_n) \in Kr(q, \mathcal{A}_\Gamma) \Leftrightarrow (q_{s_1}(a_1), \ldots, q_{s_n}(a_n)) \in P_\Gamma \Leftrightarrow$
$\Leftrightarrow \exists \vdash P(b_1, \ldots, b_n)$ astfel încât $q_{s_i}(a_i) = q_{s_i}(b_i) \Leftrightarrow \exists \vdash P(b_1, \ldots, b_n)$ şi $\vdash a_i =_{s_i} b_i$ pentru orice i $\Leftrightarrow \vdash P(a_1, \ldots, a_n)$. Ultima echivalenţă este consecinţa faptului că mulţimea teoremelor este închisă la **C**Π.

Deci mulţimea $Kr(q, \mathcal{A}_\Gamma)$ este egală cu mulţimea teoremelor.

Observaţia 5.1 $\mathcal{A}_\Gamma \models \Gamma$.

Demonstraţie: Acest fapt rezultă din observaţia că este suficient ca mulţimea $Kr(q, \mathcal{A}_\Gamma)$ să fie închisă la **MP** conform punctului b) din Propoziţia precedentă. Dar mulţimea $Kr(q, \mathcal{A}_\Gamma)$ coincide cu mulţimea teoremelor deci este închisă la **MP**.

Teorema 5.2 *Logica clauzelor Horn cu egalitate este corectă şi completă: pentru fiecare formulă atomică* B *din* $\mathcal{A}$

$$\Gamma \vdash (\forall \mathcal{A})B \Longleftrightarrow \Gamma \models (\forall \mathcal{A})B.$$

Demonstraţie: Corolarul precedent arată că orice teoremă este o tautologie. Vom demonstra că orice tautologie este o teoremă.

Presupunem că $\Gamma \models (\forall \mathcal{A})B$. Deoarece morfismul q ia valori într-o Γ-algebră deducem $B \in Kr(q, \mathcal{A}_\Gamma)$:

1) dacă B este $a =_s b$ deducem $q_s(a) = q_s(b)$ deci $\vdash a =_s b$.

2) dacă B este $P(a_1, \ldots, a_n)$ deducem $(q_{s_1}(a_1), \ldots, q_{s_n}(a_n)) \in P_\Gamma$ deci există o teoremă $\vdash P(b_1, \ldots, b_n)$ şi $q_{s_i}(a_i) = q_{s_i}(b_i)$, adică $\vdash b_i =_{s_i} a_i$. Deoarece mulţimea teoremelor este închisă la **C**Π deducem că $\vdash P(a_1, \ldots, a_n)$.

Teorema 5.3 *Pentru orice Γ-model $\mathcal{B}$ şi pentru orice Σ-morfism $h\colon \mathcal{A} \longrightarrow U(\mathcal{B})$ există şi este unic un Σ-Π-morfism $h^\# \colon \mathcal{A}_\Gamma \longrightarrow \mathcal{B}$ astfel încât $q; h^\# = h$.*

Demonstraţie: Arătăm că $a =_\Gamma c$ implică $h(a) =_s h(c)$. Deoarece $\mathcal{B}$ este Γ-model deducem că $Kr(q, \mathcal{A}) \subseteq Kr(h, \mathcal{B})$. Din $a =_\Gamma c$ deducem $a =_s c \in Kr(q, \mathcal{A})$, prin urmare $a =_s c \in Kr(h, \mathcal{B})$ deci $h(a) =_s h(c)$.

Din proprietatea de universalitate a algebrei cât rezultă existenţa unui unic Σ-morfism $h^\# \colon U(\mathcal{A}_\Gamma) \longrightarrow U(\mathcal{B})$ cu proprietatea $q; h^\# = h$. Este uşor de observat că $h^\#$ este Σ-Π-morfism. $\square$

O adjuncţie în teoria modelelor

Virgil Emil Căzănescu

January 21, 2003

Fie (S, Σ, Π) o *signatură multisortată de ordinul întâi* şi Γ o mulţime de clauze Horn.

Teorema 0.1 *Pentru orice model $\mathcal{M}$ există un Γ-model $\mathcal{M}_\Gamma$ şi un morfism de modele $q : \mathcal{M} \longrightarrow \mathcal{M}_\Gamma$ astfel încât pentru orice Γ-model $\mathcal{M}'$ şi pentru orice morfism de modele $h : \mathcal{M} \longrightarrow \mathcal{M}'$ există un unic morfism de modele $h^\# : \mathcal{M}_\Gamma \longrightarrow \mathcal{M}'$ astfel încât $q; h^\# = h$.*

Demonstraţie: În $\mathcal{M} = (M_s, M_\sigma, M_\pi)$ definim Σ-congruenţa $\sim$ prin

$$m \sim m' \Longleftrightarrow (\forall h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma) h(m) = h(m').$$

Fie (N_s, N_σ) câtul algebrei (M_s, M_σ) prin $\sim$ şi

$$q : (M_s, M_\sigma) \longrightarrow (N_s, N_\sigma)$$

morfismul de factorizare.

Pentru orice $\pi \in \Pi_{s_1 s_2 \ldots s_n}$ definim

$$N_\pi = \{(q(m_1), q(m_2), \ldots, q(m_n)) : m_i \in M_{s_i} \text{ şi } (\forall h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma)(h(m_1), h(m_2), \ldots, h(m_n)) \in M'_\pi\}.$$

Remarcăm că proprietatea folosită la definerea lui N_π este independentă de reprezentanţi, adică $(\forall h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma)(h(m_1), h(m_2), \ldots, h(m_n)) \in M'_\pi$ şi $m_i \sim m'_i$ pentru toţi indicii i implică $(\forall h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma)(h(m'_1), h(m'_2), \ldots, h(m'_n)) \in M'_\pi$.

Fie $\mathcal{M}_\Gamma = (N_s, N_\sigma, N_\pi)$. Observăm că $q : \mathcal{M} \longrightarrow \mathcal{M}_\Gamma$ este morfism de modele.

Vom demonstra că $\mathcal{M}_\Gamma \models \Gamma$.

Fie $(\forall X) C$ **if** $H \in \Gamma$ şi $f : T_\Sigma(X) \longrightarrow (N_s, N_\sigma)$ un morfism de algebre cu proprietăţile:

1. $f_s(l) = f_s(r)$ pentru orice $l =_s r \in H$
2. $(f(t_1), f(t_2), \ldots, f(t_n)) \in N_\pi$ pentru orice $\pi(t_1, t_2, \ldots, t_n) \in H$.

Deoarece $T_\Sigma(X)$ este algebră proiectivă există un morfism de algebre

$$g : T_\Sigma(X) \longrightarrow (M_s, M_\sigma)$$

cu proprietatea $g; q = f$. Deducem că:

1. $g_s(l) \sim g_s(r)$ pentru orice $l =_s r \in H$

2. $(q(g(t_1)), q(g(t_2)), \ldots, q(g(t_n))) \in N_\pi$ pentru orice $\pi(t_1, t_2, \ldots, t_n) \in H$,

prin urmare $(\forall h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma)$

1. $(g; h)_s(l) = (g; h)_s(r)$ pentru orice $l =_s r \in H$

2. $((g; h)(t_1), (g; h)(t_2), \ldots, (g; h)(t_n)) \in M'_\pi$ pentru orice $\pi(t_1, t_2, \ldots, t_n) \in H$,

adică $(\forall h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma)$

$$(\mathcal{M}', (g; h)/X) \models H.$$

Ţinând cont că $(\forall X)C$ **if** $H \in \Gamma$ pentru orice $h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma$ deducem că

$$(\mathcal{M}', (g; h)/X) \models C.$$

Daca C este $t =_s t'$ pentru orice $h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma$ deducem că $h_s(g_s(t)) = h_s(g_s(t'))$. Rezultă că $g_s(t) \sim g_s(t')$ deci $f_s(t) = f_s(t')$.

Daca $C = \pi(t_1, t_2, \ldots, t_n)$ pentru orice $h : \mathcal{M} \longrightarrow \mathcal{M}' \models \Gamma$ deducem că

$$(h(g(t_1)), h(g(t_2)), \ldots, h(g(t_n))) \in M'_\pi.$$

Rezultă că $(q(g(t_1)), q(g(t_2)), \ldots, q(g(t_n))) \in N_\pi$ deci $(f(t_1), f(t_2), \ldots, f(t_n)) \in N_\pi$. Deci $\mathcal{M}_\Gamma \models \Gamma$.

Fie $\mathcal{M}'$ un Γ-model şi $h : \mathcal{M} \longrightarrow \mathcal{M}'$ un morfism de modele. Deoarece echivalenţa nucleară a lui h include $\sim$ rezultă existenţa unui unic morfism de algebre $h^\# : \mathcal{M}_\Gamma \longrightarrow \mathcal{M}'$ astfel încât $q; h^\# = h$. Este uşor de arătat că $h^\#$ este morfism de modele.

1 ALGEBRE ASCUNSE

1 Fundament matematic

O **signatură ascunsă** are mulţimea sorturilor S partiţionată în două mulţimi disjuncte

V mulţimea sorturilor vizibile şi

H mulţimea sorturilor ascunse.

In plus se dă $\Gamma \subseteq \Sigma$. Elementele lui Γ se numesc simboluri de **operaţii comportamentale**.

În continuare $\mathcal{C}$ va fi o Γ-algebră şi z o variabilă. Contextele sunt elemente din $\mathcal{C}[z]$ în care z apare o singură dată.

Conceptele de context în algebra $\mathcal{C}$ şi de adâncime a unui context pot fi definite astfel:

1) z este context de adâncime 0

2) Daca $\gamma \in \Gamma_{s_1 s_2 \ldots s_n, s}$, dacă unul dintre elementele $a_1, a_2, \ldots, a_n$ este un context şi restul sunt din $\mathcal{C}$, atunci $\mathcal{C}[z]_\gamma(a_1, a_2, \ldots, a_n)$ este un context de adâncime cu o unitate mai mare decât contextul cu ajutorul căruia a fost format.

Un context de sort vizibil se numeşte **experiment**.

Vom spune că morfismele g şi h definite pe $\mathcal{C}[z]$ sunt $\mathcal{C}$-echivalente dacă $(\forall x \in \mathcal{C}) g(x) = h(x)$.

Von nota cu $U : Alg_\Sigma \longrightarrow Alg_\Gamma$ functorul uituc.

Definiţia 1.1 Intr-o Σ-algebră $\mathcal{A}$ spunem că elementele a şi b de acelaş sort sunt **comportamental echivalente** şi scriem $a \equiv b$ dacă şi numai dacă pentru orice Γ-algebră $\mathcal{C}$, pentru orice pereche $f, g\colon \mathcal{C}[z] \longrightarrow U(\mathcal{A})$ de morfisme $\mathcal{C}$-echivalente, cu proprietăţile $f(z) = a$ şi $f(z) = b$ şi pentru orice experiment c din $\mathcal{C}[z]$ avem $f(c) = g(c)$.

Definiţia uzuală diferă de definiţia de mai sus prin faptul că algebra $\mathcal{C}$ este liberă. Demonstraţia următoare arată că de fapt conceptele coincid.

Fie $\mathcal{C}$ o Γ-algebră, c un experiment din $\mathcal{C}[z]$ şi $f, g\colon \mathcal{C}[z] \longrightarrow U(\mathcal{A})$ o perechede morfisme $\mathcal{C}$-echivalente, cu proprietăţile $f(z) = a$ şi $f(z) = b$.

Fie $e : T_\Sigma(C \cup \{z\}) \longrightarrow \mathcal{C}[z]$ epimorfismul care duce orice element din $C \cup \{z\}$ in el însuşi. Există un experiment $c' \in T_\Sigma(C \cup \{z\})$ cu proprietatea $e(c') = c$.

Deoarece $e; f$ şi $e; g$ au aceeaşi resctricţie la C, deoarece $(e; f)(z) = a$ şi $(e; g)(z) = b$ rezultă că $(e; f)(c') = (e; g)(c')$, deci $f(c) = g(c)$.

Se numeşte **congruenţă ascunsă** într-o Σ-algebră o Γ-congruenţă care pe sorturile vizibile coincide cu egalitatea.

Propoziţie 1.2 *Echivalenţa comportamentală este congruenţă ascunsă.*

Demonstraţie: Dintre proprietăţile de echivalenţa ale lui $\equiv$ probăm doar tranzitivitatea. Fie $a \equiv b$ şi $b \equiv d$. Fie $\mathcal{C}$ o Γ-algebră, $c \in \mathrm{C[z]}_v$ un experiment şi două morfisme $\mathcal{C}$-echivalente $f, g \colon \mathcal{C}[z] \longrightarrow U(\mathcal{A})$ cu $f(z) = a$ şi $g(z) = d$. Fie $h \colon \mathcal{C}[z] \longrightarrow U(\mathcal{A})$ morfismul $\mathcal{C}$-echivalent cu f şi g pentru care $h(z) = b$. Din $a \equiv b$ deducem $f_v(c) = h_v(c)$ şi din $b \equiv d$ deducem $h_v(c) = g_v(c)$. Prin urmare $f_v(c) = g_v(c)$, deci $a \equiv d$.

Fie $a \equiv b$ pentru $a, b \in A_v$ cu $v \in V$. Fie z o variabilă de sort v. Deoarece z este un experiment, alegând morfismele $\mathcal{C}$-echivalente $f, g \colon \mathcal{C}[z] \longrightarrow U(\mathcal{A})$ astfel încât $f(z) = a$ şi $g(z) = b$ deducem $f(z) = g(z)$, deci $a = b$.

Pentru a dovedi că $\equiv$ este congruenţă ascunsă mai trebuie să arătăm că este compatibilă cu operaţiile comportamentale. Având în vedere că $\equiv$ este tranzitivă este suficient să demonstrăm compatibilitatea cu operaţiile comportamentale doar pentru câte un argument al acestora. Fără a micşora generalitatea vom poziţiona acest argument pe ultima poziţie.

Pentru $a \equiv b$ şi $\sigma \in \Gamma_{s_1 s_2 \ldots s_n s', s}$ arătăm că

$$\sigma_A(a_1, \ldots, a_n, a) \equiv \sigma_A(a_1, \ldots, a_n, b).$$

Fie $\mathcal{C}$ o Γ-algebră, $c \in \mathrm{C[z]}_v$ un experiment cu variabila z de sort s şi două morfisme $\mathcal{C}$-echivalente $f, g \colon \mathrm{C[z]} \longrightarrow U(\mathcal{A})$ cu $f_s(z) = \sigma_A(a_1, \ldots, a_n, a)$ şi $g_s(z) = \sigma_A(a_1, \ldots, a_n, b)$.

Fie $x_1, x_2, \ldots, x_n$ şi z' variabile de sorturi respectiv $s_1, s_2, \ldots, s_n$ şi s'. Fie

$$h \colon \mathcal{C}[z] \longrightarrow \mathcal{C}[x_1, \ldots, x_n, z']$$

un morfism cu proprietăţile $(\forall x \in \mathcal{C}) h(x) = x$ şi $h_s(z) = \sigma(x_1, \ldots, x_n, z')$.

Observăm că $h_v(c) \in \mathcal{C}[x_1, \ldots, x_n][z']_v$ este un experiment.

Fie $u, w \colon \mathcal{C}[x_1, \ldots, x_n][z'] \longrightarrow U(\mathcal{A})$ morfismele definite prin
$u(x) = w(x) = f(x)$ pentru orice $x \in \mathcal{C}$,
$u_{s_i}(x_i) = w_{s_i}(x_i) = a_i$ pentru orice $i \in [n]$
$u_{s'}(z') = a$ şi $w_{s'}(z') = b$.

Observăm că $h; u = f$ şi $h; w = g$. Deoarece morfismele u şi w sunt $\mathcal{C}[x_1, \ldots, x_n]$-echivalente din $a \equiv b$ deducem $u_v(h_v(c)) = w_v(h_v(c))$, prin urmare $f_v(c) = g_v(c)$, deci $\sigma_A(a_1, \ldots, a_n, a) \equiv \sigma_A(a_1, \ldots, a_n, b)$. □

Lemă 1.3 *Fie $\sim$ o congruenţă ascunsă, $c \in \mathcal{C}[z]$ şi două morfisme $\mathcal{C}$-echivalente $f, g \colon \mathcal{C}[z] \longrightarrow U(\mathcal{A})$. Dacă $f(z) \sim g(z)$, atunci $f(c) \sim g(c)$.*

Demonstraţie: Prin inducţie structurală după c.

Dacă $c = z$ concluzia este evident adevărată.

Dacă c este din $\mathcal{C}$ avem chiar egalitate.

În caz contrar $c = \mathcal{C}[z]_\gamma(a_1, \ldots, a_n)$ unde $\gamma \in \Gamma_{s_1 \ldots s_n, s}$, $a_i \in C[z]_{s_i}$. Prin ipoteza de inducţie $f(a_i) \sim g(a_i)$ pentru orice $i \in [n]$. Deoarece γ este comportamentală rezultă că

$$A_\gamma(f(a_1), \ldots, f(a_n)) \sim A_\gamma(g(a_1), \ldots, g(a_n))$$

prin urmare

$$f(\mathcal{C}[z]_\gamma(a_1, \ldots, a_n)) \sim g(\mathcal{C}[z]_\gamma(a_1, \ldots, a_n))$$

deci $f(c) \sim g(c)$. □

Teorema 1.4 *Echivalenţa comportamentală este cea mai mare congruenţă ascunsă.*

Demonstraţie: Fie $\sim$ o congruenţă ascunsă şi $a \sim b$.

Probăm că $a \equiv b$. Fie $\mathcal{C}$ o Γ-algebră, $c \in \mathcal{C}[z]_v$ un experiment şi $f, g \colon \mathcal{C}[z] \longrightarrow U(\mathcal{A})$ două morfisme $\mathcal{C}$-echivalente cu $f(z) = a$ şi $g(z) = b$. Deoarece $f(z) \sim g(z)$ din lemă deducem $f_v(c) \sim g_v(c)$, prin urmare $f_v(c) = g_v(c)$ deoarece v este sort vizibil. Deci $a \equiv b$. □

2 COINDUCŢIE

Este o metodă pentru a demonstra că două elemente sunt comportamental echivalente: este suficient să găsim o congruenţă ascunsă care conţine perechea formată din cele două elemente. Corectitudinea acestei metode se bazează pe teorema anterioară.

1 Exemplul cel mai simplu cunoscut

Un singur sort vizibil, sortul boolean, cu operaţiile specifice pe care nu le mai menţionăm.

Un singur sort ascuns: bariera.

Operaţiile comportamentale sunt:

sus : bariera $\longrightarrow$ bariera
jos : bariera $\longrightarrow$ bariera
schimb : bariera $\longrightarrow$ bariera
sus? : bariera $\longrightarrow$ boolean.

Fie S o variabilă de sort bariera. Ecuaţiile prezentării sunt:

sus?(sus S) = true
sus?(jos S) = false
sus?(schimb S) = not(sus? S).

Un model al acestei prezentări se obţine luând ca suport mulţimea numerelor naturale cu următoarele operaţii, unde n este un număr natural arbitrar:

sus(n) = $2 * n$
jos(n) = $2 * n + 1$
schimb(n) = n +1
sus?(n) = dacă (n este par) atunci true altfel false.

In acest model observăm că schimb(schimb(n)) = n + 2 deci ecuaţia

$$(\forall n) schimb(schimb(n)) = n$$

nu este adevărată în orice model. Vom dovedi totuşi că ecuaţia de mai sus este comportamental adevărată în orice model, adică

$$(\forall s) schimb(schimb(s)) \equiv s.$$

Fie $\mathcal{A}$ o algebră care pe sortul boolean este chiar algebra celor două valori clasice ale adevărului şi care este model al prezentării de mai sus.

Probăm că echivalenţa nucleară a funcţiei $sus?_A : A_{bariera} \longrightarrow A_{boolean}$, notată $Ker(sus?_A)$, este o congruenţă ascunsă.
Dacă $sus?_A(s) = sus?_A(s')$, atunci $sus?_A(schimb_A(s)) = not(sus?_A(s)) = not(sus?_A(s')) = sus?_A(schimb(s'))$.
Egalităţile $sus?_A(sus_A(s)) = sus?_A(sus(s'))$ şi $sus?_A(jos_A(s)) = sus?_A(jos(s'))$ completează demonstraţia.

Din teoremă rezultă că $Ker(sus?_A) \subseteq \equiv$. Deoarece

$$(\forall s \in A_{bariera}) sus?_A(schimb(schimb(s))) = not(not(sus?_A(s))) = sus?_A(s)$$

deducem $(\forall s) schimb(schimb(s)) \equiv s$.