Cuprins

Chapter 1

ALGEBRE MULTISORTATE

Textul a fost conceput ca o introducere în programarea declarativă, capitol semnificativ al informaticii matematice(teoretice).

Presupunem că cititorul are cunoştiinţe de teoria cantoriană a mulţimilor şi puţine cunoştiinţe de algebră. Pentru înţelegerea exemplelor se presupun cunoştiinţe privind gramaticile independente de context.

1.1 CONCEPTUL DE ALGEBRĂ MULTISORTATĂ

Conceptul de algebră multisortată apare în jurul anului 1970 prin generalizarea algebrelor universale. Noul concept mai este cunoscut şi sub numele algebre universale eterogene.

Deoarece **datele cu care lucrăm nu sunt toate la fel**, ele sunt clasificate în mai multe tipuri sau sorturi. Acesta este principalul fapt care a dus la apariţia algebrelor multisortate şi în particular a mulţimilor multisortate.

Algebrele multisortate au fost generalizate conducând la algebrele ordonat sortate.

1.1.1 Mulţimi şi funcţii multisortate

Fixăm mulţimea S a sorturilor.

Definiţia 1.1.1 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 M cât şi pentru toate componentele acesteia, M_s unde $s \in S$.

Fie $M = \{M_s\}_{s\in S}$ o mulţime S-sortată. Dacă $s \in S$ şi $m \in M_s$ spunem că elementul m are sortul s sau că s este sortul elementului m.

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

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

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

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

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

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

Definiţia 1.1.2 O **funcţie S-sortată**

$$f : M \longrightarrow N$$

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

Ca şi în cazul mulţimilor S-sortate, operaţiile cu funcţiile S-sortate se fac pe componente. Dacă $f : M \longrightarrow N$ şi $g = \{g_s\}_{s\in S} : N \longrightarrow P$ sunt funcţii S-sortate atunci compunerea lor

$$f; g = \{f_s; g_s\}_{s\in S} : M \longrightarrow P$$

este definită pentru orice $s \in S$ prin

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

Mai detaliat $(f; g)_s(x) = g_s(f_s(x))$ pentru orice $s \in S$ şi $x \in M_s$. Semnul ; folosit pentru compunere este inspirat din limbajele de programare. Mai observăm scrierea diagramatică $f; g$ utilizată pentru compunere în opoziţie cu scrierea clasică $g \circ f$

$$\begin{tikzcd} M \arrow[r, "f"] \arrow[rr, bend right, "f;g"'] & N \arrow[r, "g"] & P \end{tikzcd}$$

Compunerea funcţiilor S-sortate este asociativă. Dacă şi $h = \{h_s\} : P \longrightarrow R$ este funcţie S–sortată, atunci folosind asocitivitatea compunerii funcţiilor uzuale pentru orice $s \in S$

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

Prin urmare $(f; g); h = f; (g; h)$.

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 $s \in S$, unde 1_{M_s} este funcţia identitate a mulţimii M_s. Funcţia identitate are efect neutru la compunere. Pentru orice funcţie S–sortată $f : M \longrightarrow N$ au loc egalităţile $1_M; f = f = f; 1_N$.

Cele două proprietăţi de mai sus ne permit să vorbim de categoria mulţimilor S–sortate(Vezi definiţia categoriei 4.1.1)

1.1.2 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 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

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

faptul ca ea are n argumente de sorturi $s_1, s_2, \ldots, s_n$ şi rezultatul este de sort s. Numele unei operaţii mai este numit şi **simbol de operaţie**.

Terminologia folosită este următoarea:

$s_1 s_2 \ldots s_n$ se numeşte **aritate**,
s este **sortul rezultat** sau sortul rezultatului,
perechea $(s_1 s_2 \ldots s_n, s)$ se numeşte **rang**.

Reamintim că substantivul aritate provine din sufixul "ară" folosit în expresii ca zeroară, unară, binară, ternară, etc.

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

Definiţia 1.1.3 O signatură algebrică

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

este formată dintr-o muţime S ale cărei elemente se numesc sorturi şi o familie de mulţimi

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

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

Menţionăm că mulţimile $\Sigma_{s_1 s_2 \ldots s_n, s}$ pot avea elemente comune, ceea ce permite modelarea supraîncărcării operaţiilor, adică permisiunea ca mai multe operaţii să aibă acelaş nume, sau altfel spus să fie denumite prin acelaşi simbol.

Când nu există pericol de confuzie vom scrie signatură în loc de signatură algebrică şi vom scrie (S, Σ) sau Σ în loc de

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

Cea mai cunoscută signatură multisortată provenită din algebra clasică este cea corespunzătoare conceptelor de spaţiu vectorial sau modul. Simbolurile de operaţii sunt de trei feluri

1. simboluri de operaţii pentru scalari corespunzătoare structurii de corp sau inel
2. simboluri de operaţii pentru vectori corespunzătoare structurii de grup abelian
3. produsul cu scalari : scalar vector $\longrightarrow$ vector.

O consecinţă deosebită a stilului multisortat este faptul că relaţiile pot fi definite ca operaţii cu sortul rezultat boolean. De exemplu

$$\leq\ : \mathit{natural}\ \mathit{natural} \longrightarrow \mathit{boolean}$$

Observăm că $3 \leq 5 =$ adevăr şi $5 \leq 3 =$ fals.

În general prin **relaţie** se înţelege o operaţie al cărei sort rezultat este boolean. Practic, pentru a transforma o relaţie în operaţie se înlocuieşte relaţia cu aşa numita funcţia ei caracteristică. Reamintim că funcţia caracteristică

$$\chi_A : M \longrightarrow \{\text{adevăr, fals}\}$$

a submulţimii A a mulţimii M este definită prin

$$\chi_A(m) = \begin{cases} \text{adevăr} & \text{dacă } m \in A \\ \text{fals} & \text{dacă } m \notin A. \end{cases}$$

Remarcabil este că unele aspecte privind studiul clasic al modelelor unde apar atât operaţii cât şi relaţii poate fi redus la studiul algebrelor multisortate unde apar numai operaţii.

1.1.3 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.

Definiţia 1.1.4 O Σ-algebră $\mathcal{A} = (\{A_s\}_{s \in S}, \{A_\sigma\}_{\sigma \in \Sigma})$ este formată dintr-o mulţime S-sortată, numită suportul algebrei, $A = \{A_s\}_{s \in S}$ 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_σ este o funcţie

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

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). Mai menţionăm că pentru o algebră $\mathcal{A}$ şi suportul acesteia A folosim aceeaşi literă cu grafii diferite.

Dacă nu este pericol de confuzie în loc de Σ-algebră vom scrie mai scurt algebră.

În continuare pentru $s_1, s_2, \ldots, s_n \in S$ vom mai folosi şi notaţia

$$A_{s_1 s_2 \ldots s_n} = A_{s_1} \times A_{s_2} \times \cdots \times A_{s_n}.$$

Din definiţia de mai sus rezultă că dacă $\sigma \in \Sigma_{\lambda,s}$ unde λ este şirul vid din S^*, atunci A_σ este o funcţie definită pe o mulţime cu un element şi cu valori în A_s. Pentru a simplifica scrierea această funcţie este înlocuită cu unica ei valoare, element din A_s adică $A_\sigma \in A_s$. Deci operaţiile fără argumente, numite şi **constante**, sunt elemente ale algebrei de sort corespunzător sortului rezultat al numelui operaţiei.

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

Asemanător algebrei care abordează pe rând diferite structuri algebrce, trebuie să facem acelaş lucru. Adică trebuie să studiem, dar în acelaş timp, structuri algebrice de naturi diferite. Prin urmare, în continuare, fixăm signatura (S, Σ) a algebrelor de care ne ocupăm. De altfel unele concepte, ca de exemplu cel de morfism, nu pot fi definite decât pentru algebre având aceeaşi signatură.

Fixarea signaturii arată că ne ocupăm de o anumită structură algebrică. Faptul că signatura este arbitrară arată ca studiul diferitelor structuri algebrice se face simultan.

1.1.4 Morfisme de algebre multisortate

Un morfism între două algebre multisortate, asemănător oricărui morfism 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. Funcţia $h : A \longrightarrow B$ este morfism $h : (A, *) \longrightarrow (B, \&)$ dacă

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

Să analizăm egalitatea de mai sus. Se evaluează cei doi membri pentru un număr de elemente arbitrare din prima algebră egal cu numărul de argumente al operaţiei şi apoi se egalează rezultatele

- 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 evaluării 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: $B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \dots, h_{s_n}(a_n))$
- se cere ca rezultatul evaluării celor doi membri să fie egali.

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

Definiţia 1.1.5 Funcţia S-sortată $h : A \longrightarrow B$ este un morfism de Σ-algebre multisortate

$$h : \mathcal{A} = (A_s, A_\sigma) \longrightarrow \mathcal{B} = (B_s, B_\sigma)$$

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 $a_1 \in A_{s_1}$, $a_2 \in A_{s_2}$, $\dots$, $a_n \in A_{s_n}$

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

Dacă nu este pericol de confuzie în loc de morfism de Σ-algebre vom scrie morfism de algebre sau chiar morfism. În continuare pentru $s_1, s_2, \dots, s_n \in S$ vom mai folosi şi notaţia

$$h_{s_1 s_2 \dots s_n} = h_{s_1} \times h_{s_2} \times \cdots \times h_{s_n} : A_{s_1 s_2 \dots s_n} \longrightarrow B_{s_1 s_2 \dots s_n}.$$

Cu aceleaşi notaţii mai precizăm că dacă $a_i \in A_{s_i}$ pentru orice $1 \leq i \leq n$

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

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.

Cu notaţiile de mai sus condiţia de morfism pentru operaţia $\sigma : s_1 s_2 \dots s_n \longrightarrow s$ este echivalentă cu

$$A_\sigma; h_s = h_{s_1 s_2 \dots s_n}; B_\sigma.$$

Este deasemenea util să menţionăm diferenţa de notaţie dintre o funcţie S-sortată

$$f : A \longrightarrow B$$

între suporturile a două Σ-algebre $\mathcal{A}$ şi $\mathcal{B}$ şi un morfism între cele două algebre

$$h : \mathcal{A} \longrightarrow \mathcal{B}.$$

Observăm că funcţia identitate 1_A este morfism de Σ-algebre de la $\mathcal{A}$ la $\mathcal{A}$ fapt notat prin $1_{\mathcal{A}} : \mathcal{A} \longrightarrow \mathcal{A}$.

Propoziţie 1.1.6 *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 $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ şi $a_1 \in A_{s_1}$, $a_2 \in A_{s_2}$, $\dots$, $a_n \in A_{s_n}$. Observăm că

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

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

Compunerea morfismelor de Σ-algebre este asociativă.

Morfismul identitate are efect neutru la compunere.

Datorită celor două proprietăţi de mai sus putem vorbi de categoria Σ–algebrelor.

1.1.5 Izomorfisme de algebre multisortate

Definiţia 1.1.7 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}. Prin urmare pentru orice izomorfism $h : \mathcal{A} \longrightarrow \mathcal{B}$ menţionăm egalităţile $h; h^{-1} = 1_{\mathcal{A}}$ şi $h^{-1}; h = 1_{\mathcal{B}}$.

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

Propoziţie 1.1.8 *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, adică 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 pentru orice $s \in S$ că

$$(h; h^{-1})_s = h_s; h_s^{-1} = 1_{A_s} = (1_A)_s \quad \text{şi} \quad (h^{-1}; h)_s = h_s^{-1}; h_s = 1_{B_s} = (1_B)_s$$

deci $h; h^{-1} = 1_A$ 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 $b_1 \in B_{s_1}$, $b_2 \in B_{s_2}$, $\ldots$, $b_n \in B_{s_n}$. Deoarece h este morfism pentru elementele $h_{s_1}^{-1}(b_1), h_{s_2}^{-1}(b_2), \ldots, h_{s_n}^{-1}(b_n)$ din A 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 1.1.9 *Compunerea a două izomorfisme $f : \mathcal{A} \longrightarrow \mathcal{B}$ şi $g : \mathcal{B} \longrightarrow \mathcal{C}$ este un izomorfism $f; g : \mathcal{A} \longrightarrow \mathcal{C}$. În plus*

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

Demonstraţie: Pentru a demonstra că un morfism de la $\mathcal{M}$ la $\mathcal{N}$ este izomorfism având ca invers un alt morfism de la $\mathcal{N}$ la $\mathcal{M}$ este suficient să probăm că prin compunerea celor două morfisme, în ambele sensuri posibile, se obţin identităţi. Prin urmare folosind egalităţile $f; f^{-1} = 1_{\mathcal{A}}$, $f^{-1}; f = 1_{\mathcal{B}}$, $g; g^{-1} = 1_{\mathcal{B}}$ şi $g^{-1}; g = 1_{\mathcal{C}}$ deducem

$$(f; g); (g^{-1}; f^{-1}) = f; (g; g^{-1}); f^{-1} = f; 1_{\mathcal{B}}; f^{-1} = f; f^{-1} = 1_{\mathcal{A}} \text{ şi}$$

$$(g^{-1}; f^{-1}); (f; g) = g^{-1}; (f^{-1}; f); g = g^{-1}; 1_{\mathcal{B}}; g = g^{-1}; g = 1_{\mathcal{C}}$$

ceea ce arată că $f; g$ este izomorfism având inversul $g^{-1}; f^{-1}$.

1.2 ALGEBRE LIBERE - APLICAŢII

După această mică introducere privind algebrele multisortate, trecem la conceptul de algebră liberă datorită importantelor aplicaţii ale acestuia în informatică. Cu el se modelează noţiunile de expresie şi de evaluarea a unei expresii.

Menţionăm ca demonstraţia privind existenţa algebrelor libere prezintă dificultăţi de natură tehnică fapt pentru care este prezentată mai târziu şi recomandată numai celor care posedă spiritul demonstraţiilor matematice.

Ceilalţi se pot mulţumi numai cu explicaţiile de mai jos, acceptând, fără demonstraţie, existenţa algebrelor libere.

1.2.1 Expresii

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. Pentru început 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 1.2.1 Σ-algebra $\mathcal{A} = (A_s, A_\sigma)$ se numeşte **liber generată** de $V \subseteq A$ dacă pentru orice Σ-algebră $\mathcal{D}$ şi pentru orice funcţie sortată $f : V \to D$, există un unic morfism de Σ-algebre $f^\# : \mathcal{A} \to \mathcal{D}$ 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 1.2.2 Se numeşte **expresie** un element dintr-o algebră liberă.

Când privim algebra liberă ca o algebră de expresii, mulţimea V de mai sus reprezintă mulţimea variabilelor. În unele cazuri, elementele lui V sunt numite generatori ai algebrei.

Bineînţeles că acest concept este încă dependent de signatura cu care lucrăm, fapt pentru care atunci când dorim să precizăm signatura vorbim de Σ–expresii în loc de expresii.

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 V a generatorilor.

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 în algebra în care se fac calculele.

Menţionăm că **a da valori variabilelor din mulţimea X în algebra** $\mathcal{D}$ este echivalent cu a da o funcţie

$$v : X \longrightarrow D.$$

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 $v : X \longrightarrow D$ 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 Σ-algebre de la $T_\Sigma(X)$ la $\mathcal{D}$ şi $Set_S(X, D)$ mulţimea funcţiilor S-sortate de la X la D. Fie

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

funcţia restricţie, adică $r(h) : X \longrightarrow D$ este restricţia $h/_X$ a morfismului $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ 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 fără a mai face 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 putem spune 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 1.2.3 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 $h/_X : X \longrightarrow \mathcal{D}$. □

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

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

Semantica instrucţiunii de atribuire

Fie X mulţimea variabilelor utilizate întrun 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 $Sem(x := e) : S \longrightarrow S$, 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

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

Pentru o mai bună înţelegere menţionăm că $s(y)$ este valoarea variabilei y în momentul începerii execuţiei instrucţiunii de atribuire şi că $Sem(x := e)(s)(y)$ este valoarea variabilei y în momentul terminării execuţiei instrucţiunii de atribuire $x := e$.

1.2.2 Unicitatea abstracţie de un izomorfism a algebrelor libere

Dacă A este o submulţime a lui B numim **funcţie incluziune** a lui A în B funcţia $i : A \longrightarrow B$ definită prin $i(a) = a$ pentru orice $a \in A$. În acest caz folosim şi notaţia $i : A \hookrightarrow B$.

Teorema 1.2.4 *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 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 $i_B; g = i_A$.

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, folosind partea de unicitate din definiţia algebrei libere, că $f; g = 1_{\mathcal{A}}$.

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, folosind partea de unicitate din definiţia algebrei libere, că $g; f = 1_{\mathcal{B}}$.

Deci f şi g sunt izomorfisme inverse unul altuia.

Propoziţie 1.2.5 *Orice algebră izomorfă cu o algebră liberă este algebră liberă.*

Demonstraţie: Fie $\mathcal{A}$ o Σ–algebră liber generată de $V \subseteq A$ şi $h : \mathcal{A} \longrightarrow \mathcal{B}$ un izomorfism.

Probăm că Σ–algebra $\mathcal{B}$ este liber generată de $h(V) \subseteq B$. Fie $\mathcal{C}$ o Σ–algebră şi $f : h(V) \longrightarrow C$ o funcţie S–sortată. Fie $g : V \longrightarrow C$ funcţia S–sortată definită prin

$$(\forall v \in V) g(v) = f(h(v)).$$

Deoarece $\mathcal{A}$ este Σ–algebră liber generată de V există un unic Σ–morfism $g^{\#} : \mathcal{A} \longrightarrow \mathcal{C}$ cu proprietatea

$$(\forall v \in V) g^{\#}(v) = g(v).$$

$$\begin{tikzcd}
V \arrow[r, hook] \arrow[d, dotted] \arrow[dd, bend right, "g"'] & \mathcal{A} \arrow[d, "h"'] \arrow[ddl, bend left, "g^{\#}"] \\
h(V) \arrow[r, hook] \arrow[d, "f"] & \mathcal{B} \\
\mathcal{C} &
\end{tikzcd}$$

Calculând restricţia morfismului $h^{-1}; g^{\#} : \mathcal{B} \longrightarrow \mathcal{C}$ la $h(V)$ observăm că

$$(\forall v \in V)(h^{-1}; g^{\#})(h(v)) = g^{\#}(h^{-1}(h(v))) = g^{\#}(v) = f(h(v)),$$

adică Σ–morfismul $h^{-1}; g^{\#}$ este o extindere a funcţiei f.

Vom proba unicitatea acestei extinderi. Fie $t : \mathcal{B} \longrightarrow \mathcal{C}$ un morfism a cărui restricţie la $h(V)$ este f. Pentru orice $v \in V$ observăm că

$$(h; t)(v) = t(h(v)) = f(h(v)) = g(v).$$

Deducem că $h; t = g^{\#}$ deci $t = h^{-1}; g^{\#}$. $\square$

Algebre iniţiale

Definiţia 1.2.6 O Σ-**algebră** $\mathcal{I}$ se numeţe **inţială** 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 cele două propoziţii 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 date.

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.

1.2.3 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 axiome(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 poate fi definit abstract ca fiind semiinelul iniţial.

Observăm că o astfel de definiţie nu spune nimic despre scrierea numerelor. Ele pot fi scrise cu cifre arabe, cu cifre romane, în baza 2 ca în calculatoare sau altfel.

Definiţiile de mai sus, deşi corecte sunt uneori ineficiente, deoarece nu face posibilă execuţia de calcule sau execuţia este ineficientă. Prin urmare dorim alte definiţii echivalente care permit calculatorului să facă calcule care să fie cât mai eficiente. Vom exemplifica pentru numere naturale fără a intra în prea multe detalii.

Tipul abstract al numerelor naturale

Semiinelul este o mulţime M înzestrată cu două operaţii binare notate cu $+$ şi $*$ cu următoarele proprietăţi

1. $(M, +, 0)$ este monoid comutativ,
2. $(M, *, 1)$ este monoid,
3. $*$ este distributivă faţă de $+$ şi
4. $(\forall m \in M) m * 0 = 0 * m = 0$.

Mulţimea numerelor naturale cu adunarea şi inmulţirea uzuale este un semiinel iniţial, adică conceptul de semiinel caracterizează numele naturale ca tip abstract de date.

Pentru a scrie un program se preferă axiomele lui Peano. Chiar dacă Peano nu s-a gândit la programarea prin rescriere, baza programării declarative, se pare să fi scris primul program de acest gen.

Cel căruia i se atribuie prima punere în evidenţă a ideilor abstracte privind numerele naturale este F.W. Lawvere[14].

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 ...

Propoziţia următoare în care conceptul de număr natural este folosit în înţelesul său clasic arată că numerele naturale formează un model al definiţiei abstracte. Acest fapt arată corectitudinea definiţiei abstracte pentru modelul clasic.

Propoziţie 1.2.7 *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

$h(0_N) = 0_A$
$h(n + 1) = s_A(h(n))$ 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 presupunând $g(n) = h(n)$ deducem
$g(n + 1) = g(s_N(n)) = s_A(g(n)) = s_A(h(n)) = h(s_N(n)) = h(n + 1)$. $\square$

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 adaugăm 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 să adune 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 probată mai târziu, în secţiunea 1.9.1.

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 .

Uneori, în programarea prin rescriere, dacă dorim să scriem un program, partea cea mai dificilă este definirea abstractă a tipurilor de date cu care lucrăm. Plecând de la noţiunea intuitivă dată de o algebră $\mathcal{D}$ a datelor, trebuie să găsim signatura şi eventual ecuaţiile pentru care $\mathcal{D}$ devine algebră iniţială.

Exemplul numerelor naturale este doar un început. Vă propunem de exemplu să definiţi relaţia de ordine ca operaţie $<$: nat nat $\longrightarrow$ *bool*. Piatra de încercare va fi însă operaţia de împărţire.

1.3 SUBALGEBRE

Vom continua prezentarea teoriei algebrelor multisortate cu conceptele de parte stabilă şi subalgebră. Deoarece părţile stabile formează o familie Moore, concept echivalent cu cel de operator de închidere vom începe cu o scurtă prezentare a acestor concepte.

1.3.1 Operator de închidere. Familie Moore

Fie $(P, \leq)$ o mulţime parţial ordonată, adică relaţia $\leq$ pe mulţimea P este reflexivă, tranzitivă şi antisimetrică.

Definiţia 1.3.1 Se numeşte **operator de închidere** o funcţie

$$\bullet : P \longrightarrow P$$

cu următoarele proprietăţi:

1. *extensivitate:* $(\forall p \in P)\, p \leq p^\bullet$,
2. *idempotenţă:* $(\forall p \in P)\, p^\bullet = p^{\bullet\bullet}$,
3. *monotonie:* $(\forall p, q \in P)\, p \leq q \Longrightarrow p^\bullet \leq q^\bullet$.

Vom folosi următorul limbaj. $p^\bullet$ este numit **închiderea** elementului p. Un element p cu proprietatea $p^\bullet = p$ se va numi **închis**.

Definiţia 1.3.2 Se numeşte **familie Moore** o submulţime $M \subset P$ astfel încât oricare ar fi $p \in P$ mulţimea $\{m \in M \mid p \leq m\}$ are prim, adică cel mai mic, element.

Vom folosi în continuare pentru orice $p \in P$ notaţia

$$M_p = \{m \in M \mid p \leq m\}.$$

Propoziţie 1.3.3 *Dacă* $^\bullet : P \longrightarrow P$ *este un operator de închidere, atunci mulţimea elementelor închise*

$$M = \{p \in P : p = p^\bullet\}$$

este o familie Moore. În plus $p^\bullet$ *este cel mai mic(prim) element al mulţimii* M_p.

Demonstraţie: Pentru orice $p \in P$ probăm că $p^\bullet$ este cel mai mic(prim) element al mulţimii M_p.

Deoarece $p^\bullet = p^{\bullet\bullet}$ rezultă că $p^\bullet \in M$. Conform extensivităţii $p \leq p^\bullet$ deducem că $p^\bullet$ aparţine mulţimii M_p. Pentru orice $m \in M_p$ din $p \leq m$ deducem $p^\bullet \leq m^\bullet$ şi din $m \in M$ deducem $m^\bullet = m$ deci $p^\bullet \leq m$. $\square$

Observaţia 1.3.4 *Dacă* $\bullet : P \longrightarrow P$ *este un operator de închidere:* $p \leq q^\bullet$ *implică* $p^\bullet \leq q^\bullet$

Demonstraţie: Deoarece $q^\bullet$ este un element închis cu proprietatea $p \leq q^\bullet$ şi $p^\bullet$ este cel mai mic element închis cu proprietatea $p \leq p^\bullet$. $\square$

Propoziţie 1.3.5 *Dacă* M *este o familie Moore şi dacă pentru orice* $p \in P$ *definim* $p^\bullet$ *ca fiind cel mai mic element din* M_p *obţinem un operator de închidere* $^\bullet : P \longrightarrow P$.

Demonstraţie: Deoarece $p^\bullet \in M_p$ deducem $p^\bullet \in M$ şi $p \leq p^\bullet$ pentru orice $p \in P$, adică extensivitatea.

Deoarece $p^{\bullet\bullet}$ este cel mai mic element din mulţimea $\{m \in M \mid p^\bullet \leq m\}$ şi $p^\bullet$ aparţine acestei mulţimi rezultă $p^{\bullet\bullet} \leq p^\bullet$. Folosind extensivitatea $p^\bullet \leq p^{\bullet\bullet}$ deducem $p^\bullet = p^{\bullet\bullet}$.

Dacă $p \leq q$ cum $q^\bullet$ este cel mai mic element din $\{m \in M \mid q \leq m\}$ rezultă că $q^\bullet \in M$ şi $q \leq q^\bullet$, prin urmare $p \leq q^\bullet$. Deoarece $q^\bullet \in \{m \in M \mid p \leq m\}$ deducem $p^\bullet \leq q^\bullet$. $\square$

Propoziţie 1.3.6 *Fie* $'$ *un operator de închidere,* $M = \{p \in P \mid p = p'\}$ *familia Moore asociată lui* $'$ *conform propoziţiei 1.3.3 şi* $^\bullet$ *operatorul de închidere asociat familiei Moore* M *conform propoziţiei 1.3.5. Operatorii de închidere* $'$ *şi* $^\bullet$ *coincid.*

Demonstraţie: Demonstrăm pentru orice $p \in P$ că $p' = p^\bullet$.

Fie $p \in P$. Prin definiţie $p^\bullet$ este cel mai mic element al mulţimii M_p. Dar propoziţia 1.3.3 spune că p' este cel mai mic element al lui M_p, prin urmare $p' = p^\bullet$.

Deci operatorii de închidere $'$ şi $^\bullet$ coincid. $\square$

Propoziţie 1.3.7 *Dacă M este o familie Moore, $^\bullet$ operatorul de închidere asociat lui M conform propoziţiei 1.3.5 şi $M^\bullet = \{q \in P \mid q = q^\bullet\}$ familia Moore asociată lui $^\bullet$ conform propoziţiei 1.3.3, atunci $M = M^\bullet$.*

Demonstraţie: Arătăm pentru orice $p \in P$ că $p \in M$ dacă şi numai dacă $p \in M^\bullet$.

Fie $p \in M^\bullet = \{m \in P \mid m = m^\bullet\}$. Rezultă că $p = p^\bullet$. Din $p^\bullet \in M_p$ deducem că $p^\bullet \in M$ deci $p \in M$.

Dacă $p \in M$ atunci p este cel mai mic element din mulţimea $\{m \in M \mid p \leq m\}$ deci $p = p^\bullet$ adică $p \in M^\bullet$. $\square$

Din propoziţiile de mai sus se observă echivalenţa celor două concepte. Fixând mulţimea parţial ordonată $(P, \leq)$, considerăm multimea $\mathcal{O}$ a operatorilor de închidere şi mulţimea $\mathcal{M}$ a familiilor Moore din P. Propoziţia 1.3.3 ne dă o funcţie de la $\mathcal{O}$ la $\mathcal{M}$. Propoziţia 1.3.5 ne dă o funcţie de la $\mathcal{M}$ la $\mathcal{O}$. Celelalte două propoziţii ne spun că aceste functii sunt bijecţii inverse una alteia.

Bazându-ne pe motivaţia de mai sus nu vom mai face nici o distinţie între cele două concepte, apariţia în discuţie a unuia dintre ele introducând automat şi pe celălalt.

Următoarea propoziţie este utilă în multe exemple.

Propoziţie 1.3.8 *Într-o latice completă L, M este familie Moore dacă şi numai dacă pentru orice $A \subseteq M$, $infA \in M$.*

Demonstraţie: Reamintim că prin definiţie orice submulţime A dintr-o latice completă are infimum(cel mai mare minorant), pe care în continuare îl notăm $infA$.

Presupunem că M este familie Moore. Pentru $A \subseteq M$ probăm $infA \in M$. Operatorul de închidere asociat fiind extensiv, deducem $infA \leq (infA)^\bullet$. Probăm că $(infA)^\bullet$ este minorant pentru A. Dacă $a \in A$ atunci $infA \leq a$ de unde, deoarece operatorul de închidere este crescător deducem $(infA)^\bullet \leq a^\bullet$. Deoarece $a \in M$ deducem $a^\bullet = a$ prin urmare $(infA)^\bullet \leq a$. Aşadar $(infA)^\bullet$ este minorant pentru A, prin urmare $(infA)^\bullet \leq infA$. Rezultă că $(infA)^\bullet = infA$, deci $infA$ este element închis şi $infA \in M$.

Pentru cealaltă implicaţie, presupunem că orice submulţime a lui M are infimum în M. Probăm că M este familie Moore. Fie $p \in L$ şi notăm $M_p = \{m \in M \mid p \leq m\}$. Din ipoteză, $infM_p \in M$. Deoarece p este minorant pentru M_p rezultă că $p \leq infM_p$ prin urmare $infM_p \in M_p$ este primul element al mulţimii M_p. $\square$

1.3.2 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 1.3.9 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 $s_1 s_2 \ldots s_n \in S^*$, pentru orice $s \in S$, pentru orice $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$, pentru orice $a_1 \in P_{s_1}$, $a_2 \in P_{s_2}$,…,$a_n \in P_{s_n}$ elementul $A_\sigma(a_1, a_2, \ldots, a_n)$ este în P_s.

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

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 şi aplica apoi propoziţia de mai sus. Lăsăm acest fapt ca exerciţiu. Vom prefera o cale mai dificilă, utilizând direct definiţia, 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
$X_s^{n+1} = X_s^n \cup \{A_\sigma(a) : w \in S^*, \sigma \in \Sigma_{w,s}, a \in X_w^n\}$ pentru orice $n \in N$ şi orice $s \in S$.

Observăm că şirul $\{X^n\}_{n \in N}$ este crescător. Mai observăm că mulţimea X^{n+1} se obţine adăugând la X^n rezultatele aplicării unei singure operaţii, o singură dată, unor elemente din X^n.

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

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

Propoziţie 1.3.10 $\overline{X}$ *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 evidentă deoarece $X = X^0 \subseteq \overline{X}$.

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

Definiţia 1.3.11 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 sau că X este o mulţime de generatori ai algebrei $\mathcal{A}$.

Operatorul de închidere asociat familiei Moore a părţilor stabile mai are şi 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.

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 $\mathbf{P}$ o proprietate referitoare la elementele algebrei $\mathcal{A}$. Pentru a dovedi că toate elementele algebrei $\mathcal{A}$ au proprietatea $\mathbf{P}$ este suficient ca să dovedim că

1. orice element din X are proprietatea **P,**

2. pentru orice $s_1s_2\dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1s_2\dots s_n,s}$ dacă $a_1 \in A_{s_1}$, $a_2 \in A_{s_2}$, $\dots$, $a_n \in A_{s_n}$ sunt elemente arbitrare cu proprietatea **P**, atunci $A_\sigma(a_1, a_2, \dots, a_n)$ are proprietatea **P.**

Să ne convingem 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.

Deoarece $\overline{X}$ este cea mai mică parte stabilă care include X din cele două proprietăţi deducem $\overline{X} \subseteq B$. Dar $\overline{X} = A$ deoarece X generează algebra $\mathcal{A}$, prin urmare $A \subseteq B$, deci orice element din A are proprietatea **P.**

Subalgebre

Conceptul de subalgebră este foarte apropiat de cel de parte stabilă. Diferenţa principală constă în faptul că subalgebra este o algebră iar partea stabilă 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 $s_1s_2\dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1s_2\dots s_n,s}$ şi $b_1 \in B_{s_1}$, $b_2 \in B_{s_2}$, $\dots$, $b_n \in B_{s_n}$.

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 $s_1s_2\dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1s_2\dots s_n,s}$ şi $b_1 \in B_{s_1}$, $b_2 \in B_{s_2}$, $\dots$, $b_n \in B_{s_n}$.

1.3.3 Morfisme şi părţi stabile

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

Definiţia 1.3.12 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 1.3.13 *Nucleul de săgeată dublă este o parte stabilă.*

Demonstraţie: Fie $s_1s_2\dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1s_2\dots s_n,s}$ şi $a_1 \in Ker(f, g)_{s_1}$, $a_2 \in Ker(f, g)_{s_2}$, $\dots$, $a_n \in Ker(f, g)_{s_n}$. Pentru orice $1 \le i \le n$ 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 $\quad A_\sigma(a_1, a_2, \dots, a_n) \in Ker(f, g)_s$. $\square$

Corolar 1.3.14 *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.*

Demonstraţie: *Deoarece prin ipoteză $(\forall x \in X) f(x) = g(x)$ deducem $X \subseteq Ker(f, g)$. Deoarece $Ker(f, g)$ este parte stabilă şi $\overline{X}$ este cea mai mică parte stabilă care include X deducem $\overline{X} \subseteq Ker(f, g)$. Deci $(\forall x \in \overline{X}) f(x) = g(x)$, adică restricţiile lui f şi g la $\overline{X}$ coincid.* $\square$

Corolar 1.3.15 *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 $f = g$.*

Demonstraţie: *Fie X o mulţime de generatori ai algebrei $\mathcal{A}$, adică $\overline{X} = A$. Presupunem că restricţiile lui f şi g la X sunt egale. Din corolarul anterior deducem că restricţiile lui f şi g la $\overline{X}$ coincid. Dar $\overline{X} = A$ implică $f = g$.*

Propoziţie 1.3.16 *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 $s_1s_2\ldots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1s_2\ldots s_n,s}$ şi $b_1 \in h_{s_1}(P_{s_1})$, $b_2 \in h_{s_2}(P_{s_2})$, $\ldots$, $b_n \in h_{s_n}(P_{s_n})$. 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 $A_\sigma(p_1, p_2, \ldots, p_n) \in P_s$. Observăm că

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

Prin urmare $B_s(b_1, b_2, \ldots, b_n) \in h_s(P_s)$, 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 $s_1s_2\ldots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1s_2\ldots s_n,s}$ 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 $h_{s_1}(a_1) \in Q_{s_1}$, $h_{s_2}(a_2) \in Q_{s_2}$, $\ldots$ $h_{s_n}(a_n) \in Q_{s_n}$ deducem $B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \ldots, h_{s_n}(a_n)) \in Q_s$. Deoarece

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

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

3. Din $X \subseteq \overline{X}$, deoarece imaginea directă este crescătoare, deducem $h(X) \subseteq h(\overline{X})$. Deoarece membrul drept este o parte stabilă a lui $\mathcal{B}$ 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 a lui $\mathcal{A}$ 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.

1.4 EXISTENŢA ALGEBRELOR LIBERE

1.4.1 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 1.4.1 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 1.4.2 Pentru orice $m \in M$ există şi sunt unice numărul natural n şi elementele $b_1 \in B$, $b_2 \in B$, $\ldots$, $b_n \in B$ cu proprietatea $m = b_1b_2\ldots b_n$.

Să observăm diferenţa esenţială dintre cele două definiţii echivalente. Observăm că în definiţia 1.4.1 nu apar de loc elemente, apărând numai concepte din afara monoidului M. Definiţia 1.4.2 în schimb lucrează numai cu elemente din interiorul monoidului.

Comparând definiţia dată algebrelor libere cu cele două de mai sus constatăm că definiţia algebrelor libere este asemănătoare cu definiţia 1.4.1. Sunt atât de asemănătoare încât pot fi generalizate la cel mai înalt nivel de abstractizare, cel al teoriei categoriilor prin conceptul de săgeată universală.

Conceptul asemănător celui din definiţia 1.4.2 este cel de algebră Peano, concept echivalent, aşa cum am spus mai sus, 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 a existenţei algebrei Peano. Deoarece textul se adresează în primul rând 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 ai nodului.

Existenţa algebrelor libere - varianta scurtă

Definiţia 1.4.3 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 funcţie $f : X \longrightarrow S$. 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.

In această variantă expresiile sunt scrise folosind scrierea uzuală cu paranteze şi virgule.
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 $s_1 s_2 \ldots s_n \in S^*$, pentru orice $s \in S$, pentru orice $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$, dacă $t_i \in T_\Sigma(X)_{s_i}$ pentru orice $1 \leq i \leq n$, atunci $\sigma(t_1, t_2, \ldots, t_n) \in T_\Sigma(X)_s$.

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 $n \geq 1$, orice $s_1 s_2 \ldots s_n \in S^*$, orice $s \in S$, 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 din cololarul 1.3.15 că $h = f^{\#}$.

Unde-i greşeala?

1.4.2 Algebre Peano

Definiţia 1.4.4 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 1.4.5 *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 $s \in S$

$$X_s^{n+1} = X_s^n \cup \{A_\sigma(a) : w \in S^*,\ \sigma \in \Sigma_{w,s} \text{ şi } a \in X_w^n\}.$$

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

$$h_s^{n+1}(a) = \begin{cases} h_s^n(a) & \text{dacă } a \in X_s^n \\ B_\sigma(h_w^n(a')) & \text{dacă } a = A_\sigma(a') \notin X_s^n \text{ unde } w \in S^*,\ \sigma \in \Sigma_{w,s} \text{ şi } a' \in X_w^n \end{cases}$$

Corectitudinea acestei definiţii rezultă din condiţia 3 din definiţia algebrei Peano, care ne asigură că scrierea lui a sub forma $a = A_\sigma(a')$ este unică.

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

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

Observăm că pentru orice $s \in S$ şi orice $a \in A_s$ deoarece $A = \bigcup_{n \in N} X^n$ există $n \in N$ cu proprietatea $a \in X_s^n$.

Observăm că $g_s(a) = h_s^m(a)$ pentru orice număr natural m cu proprietatea $a \in X_s^m$.

Probăm că $g : \mathcal{A} \longrightarrow \mathcal{B}$ este morfism de algebre. Fie $\sigma \in \Sigma_{w,s}$ şi $a \in A_w$.
Deoarece conform condiţiei 2 din definiţia algebrei Peano $A_\sigma(a) \notin X_s^0$ există n cel mai mic număr natural cu proprietatea $A_\sigma(a) \in X_s^{n+1} - X_s^n$. Rezultă că

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

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 conform condiţiei 3 din definiţia algebrei Peano 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 $\mathcal{A}$. $\square$

Am putea spune că această teoremă este partea comună a tuturor demonstraţiilor privind existenţa algebrelor libere. Rolul acestei teoreme este de a reduce demonstraţia existenţei algebrelor libere la existenţa algebrelor Peano. Cu alte cuvinte demonstraţia de natură semantică este redusă la aspectele ei sintactice.

Menţionăm că propoziţia 1.4.7 este reciproca teoremei de mai sus.

1.4.3 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 $(n, t_0 n_1 t_1 \ldots n_k t_k) \in P$ unde literele n sunt neterminale şi literele t sunt cuvinte cu litere terminale, atunci

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

adică operaţia corespunzătoare lui $(n, t_0 n_1 t_1 \ldots n_k t_k)$ 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.

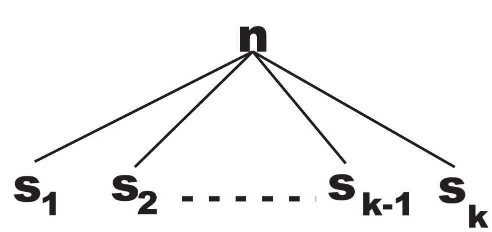
Un arbore de derivare într-o gramatică independentă de context are următoarele proprietăţi:

1. este un arbore finit şi local ordonat, adică succesorii fiecărui nod sunt într-o ordine totală;

2. are noduri etichetate cu terminale sau neterminale, rădăcina fiind etichetată cu un neterminal;

3. orice nod etichetat cu un terminal nu are nici un succesor;

4. pentru orice nod, dacă este etichetat cu un neterminal n, atunci perechea formată din n şi cuvântul format de etichetele succesorilor formează o producţie $(n, s_1 s_2 \ldots s_{k-1} s_k) \in P$.

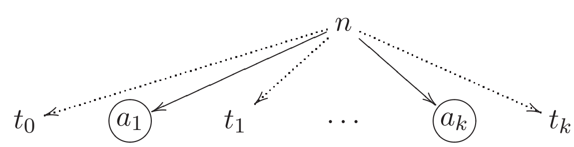


Arborii de derivare ai unei gramatici independente de context G pot fi organizaţi ca o G-algebră $\mathcal{A}$ după cum urmează:

- pentru orice sort n, adică pentru orice neterminal n, mulţimea A_n este formată din totalitatea arborilor de derivare care au rădăcina etichetată cu n.

- pentru producţia $p = (n, t_0n_1t_1 \ldots n_kt_k)$ şi arborii $a_i \in A_{n_i}$ arborele $A_p(a_1, a_2, \ldots, a_k)$ este format astfel: rădăcina este etichetată cu n, etichetele succesorilor rădăcinii în număr egal cu numărul literelor din $t_0n_1t_1 \ldots n_kt_k$ sunt chiar literele acestui cuvânt şi subarborele fiecărui nod etichetat cu n_i este chiar a_i.

$A_p(a_1, a_2, \ldots, a_k) =$



Teorema 1.4.6 *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. Aici este esenţială finitudinea arborilor de derivare deoarece aceasta implică existenţa adâncimii oricărui 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 $p = (n, t_0n_1t_1 \ldots n_kt_k)$ cu proprietatea

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

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

Observăm că arborii $a_1, a_2, \ldots, a_k$ 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 $A_p(a_1, a_2, \ldots, a_k) \in P_s$, deci $a \in P_s$.

A doua condiţie din definiţia algebrelor Peano

$$A_p(a_1, a_2, \ldots, a_k) \notin \emptyset$$

este evident adevărată.

Trecem la ultima condiţie. Fie $p = (n, t_0n_1t_1 \ldots n_kt_k)$ şi $q = (n, t'_0s_1t'_1 \ldots s_{k'}t'_{k'})$ 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

$$t_0n_1t_1 \ldots n_kt_k = t'_0s_1t'_1 \ldots s_{k'}t'_{k'}.$$

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

$$t_1n_2t_2 \ldots n_kt_k = t'_1s_2t'_2 \ldots s_kt'_k.$$

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 a primului nivel din arborii egali $A_p(a_1, a_2, \ldots, a_k)$ şi $A_q(b_1, b_2, \ldots, b_{k'})$ rezultă că $a_i = b_i$ pentru orice $1 \leq i \leq k$. $\square$

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 $p = (n, \text{cuvântul vid})$ este producţie, deci $a = A_p$. În concluzie $A_p \in P_n$ deoarece P este parte stabilă, deci $a \in P_n$.

1.4.4 Existenţa algebrelor iniţiale

Urmărim să arătăm că pentru orice signatură $(S, \{\Sigma_{w,s}\}_{w \in S^*, s \in S})$ există o Σ-algebră iniţială pe care în cele ce urmează o vom nota cu T_Σ.

Construcţia algebrei iniţiale care urmează este bazată pe scrierea poloneză a expresiilor fără variabile, adică un şir de semne de operaţii în care semnul de operaţie este plasat în faţa argumentelor sale care trebuie sa-l urmeze. Pentru generarea acestor expresii vom folosi o gramatică independentă de context. Producţiile de forma $(s, \sigma s_1 s_2 \ldots s_n)$ unde $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ 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.

Fără a micşora generalitatea vom presupune ca S şi Σ sunt disjuncte.

Considerăm G gramatica independentă de context definită prin

1. Mulţimea neterminalelor este S,
2. Mulţimea terminalelor este Σ şi
3. Mulţimea producţiilor este

$$\{(s, \sigma w) \,|\, w \in S^*, s \in S \text{ şi } \sigma \in \Sigma_{w,s}\}.$$

Notăm cu $\mathcal{A} = (A_s, \{A_{(s,\sigma w)}\}_{\sigma \in \Sigma_{w,s}})$ algebra arborilor ei de derivare. Ea este, conform teoremei 1.4.6, algebră Peano peste mulţimea vidă. Conform teoremei 1.4.5, $\mathcal{A}$ este G-algebră liber generată de mulţimea vidă, adică G-algebră iniţială.

Nu ne mai rămâne decât să observăm că noţiunile de G-algebră şi Σ-algebră coincid. Pentru aceasta pentru $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ operaţia corespunzătoare producţiei ataşate $(s, \sigma s_1 s_2 \ldots s_n)$ este o funcţie

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

1.4.5 Existenţa algebrelor libere

Urmărim să arătăm că pentru orice signatură (S, 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 $T_\Sigma(X)$. Reamintim că mulţimea S-sortată X are componentele disjuncte două câte două. Fără a micşora generalitatea vom presupune că Σ şi X sunt disjuncte.

Construcţia algebrei libere generate de o mulţime de variabile se bazează pe existenţa algebrelor inţiale. Comparând formarea, în stil intuitiv, al expresiilor fără variabile cu a expresiilor cu variabile observăm că diferenţa este doar la inceput. Pentru construcţia expresiilor fără variabile se pleacă de la costante, operaţiile fără argumente. Pentru construcţia expresiilor cu variabile se pleacă de la costante şi variabile. În continuare, în ambele cazuri se continuă cu aplicarea operaţiilor neconstante în toate modurile posibile. Se ajunge, în cazul expresiilor fără variabile, la algebra iniţială şi în cazul expresiilor cu variabile din mulţimea S-sortată X la algebra liber generată de X.

Trecem la formalizarea ideilor de mai sus. Considerăm signatura algebrică $\Sigma_X = \Sigma \cup X$ unde orice variabilă $x \in X_s$ devine o constantă de sort s.

Fie

$$\mathcal{I} = (\{I_s\}_{s\in S}, \{I_\sigma\}_{\sigma\in\Sigma}, \{I_x\}_{x\in X})$$

o Σ_X–algebră iniţială. Vom proba că Σ–algebra $(\{I_s\}_{s\in S}, \{I_\sigma\}_{\sigma\in\Sigma})$ impreună cu funcţia injectivă $f : X \longrightarrow \{I_s\}_{s\in S}$ defintă prin $f_s(x) = I_x$ pentru orice $s \in S$ şi $x \in X_s$ sunt o Σ–algebră liber generată de X.

Fie $\mathcal{A} = (A_s, A_\sigma)$ o Σ–algebră şi $g : X \longrightarrow A$ o funcţie S–sortată. Considerăm Σ_X–algebra $(A_s, A_\sigma, \{g(x)\}_{x\in X})$ şi unicul morfism de Σ_X–algebre existent

$$h : \mathcal{I} \longrightarrow (A_s, A_\sigma, \{g(x)\}_{x\in X}).$$

Este suficient să mai remarcăm că $h : \mathcal{I} \longrightarrow (A_s, A_\sigma, \{f(x)\}_{x\in X})$ este morfism de Σ_X–algebra dacă şi numai dacă $h : (\{I_s\}_{s\in S}, \{I_\sigma\}_{\sigma\in\Sigma}) \longrightarrow \mathcal{A}$ este morfism de Σ–algebre şi $g; h = f$. $\square$

Propoziţie 1.4.7 *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 $x \in X$. Deoarece orice algebră izomorfă cu o algebră Peano este algebră Peano rezultă că $\mathcal{L}$ este algebră Peano peste X.

1.5 SEMANTICA ALGEBREI INIŢIALE

Menţionăm aici pe marele informatician, regretatul Joseph Goguen, profesorul şi prietenul multor români, care spunea că "semantica algebrei iniţiale" este una dintre cele mai frumoase idei ale sale [10].

Metoda semanticii algebrei initiale este o simplificare a metodei mai clasice a semanticii denotaţionale, numită şi semantică matematică.

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 muţimea 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** $\mathcal{S} = (\{S_n\}_{n\in N}, \{S_p\}_{p\in P})$, numită în continuare **algebră semantică**.

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.

1.5.1 Semantica unui şir de cifre ca număr natural

Vom considera o gramatică G care generează şirurile finite nevide 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. 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 $< cifra >< nat >$?

1.5.2 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
$+$ $\times$ pentru adunare şi înmulţire

M pentru unica celulă de memorie a calculatorului
( ) pentru parantezele necesare în scrierea expresiilor
IF , pentru o anumită expresie condiţionată
E pentru comanda de evaluarea a unei expresii

Remarcăm că butoanele diferite de ON, OFF şi E sunt folosite pentru introducerea expresiilor care vor fi evaluate la apăsarea butonului E. Sintaxa expresiilor este definită de producţiile r1-r6 ale gramaticii de mai jos. Expresiile sunt construite pornind de la şirurile finite de cifre zecimale reprezentând numere naturale(producţia r1), variabila M(producţia r2) şi recursiv, alte expresii cuprinse între paranteze(producţia r6). Pentru construţia expresiilor sunt folosite numai operaţiile de adunare şi înmulţire(producţiile r3 şi r4). Producţia r5 permite construirea unor expresii condiţionate unde condiţia reprezentată de prima expresie permite o alegere între celelalte două expresii.

Vom considera o gramatică G care generează limbajul programelor care pot fi executate de minicalculator. Ea extinde gramatica din exemplul precedent. Numele butoanelor de mai sus sunt chiar terminalele gramaticii.

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 pe care l-am scris la începutul rândului, în faţa producţiei.

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> $\times$ <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 rezultatul evaluării, adică 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, generată de producţiile r7 şi r8, 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

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

Un program, generat de producţia r9, 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 $S_{<program>} = N^+$.

Folosind 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

definiţiile pentru operaţii, fără a le 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 $e3(m)$
$S_{r6}(e) = e$
$S_{r7}(e)(m) = e(m)$
$S_{r8}(e, \imath)(m) = e(m)\imath(e(m))$
$S_{r9}(\imath) = \imath(0)$

1.5.3 Arbori de derivare

Reamintim proprietăţile unui arbore de derivare dintr-o gramatică independentă de context:
1. este un arbore finit şi local ordonat, adică succesorii fiecărui nod sunt într-o ordine totală;
2. are noduri etichetate cu terminale sau neterminale, rădăcina fiind etichetată cu un neterminal;
3. orice nod etichetat cu un terminal nu are nici un succesor;
4. pentru orice nod, dacă este etichetat cu un neterminal n, atunci perechea formată din n şi cuvântul format de etichetele $s_1, s_2, \ldots, s_k$ ale succesorilor săi formează o producţie $(n, s_1 s_2 \ldots s_{k-1} s_k) \in P$.

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.

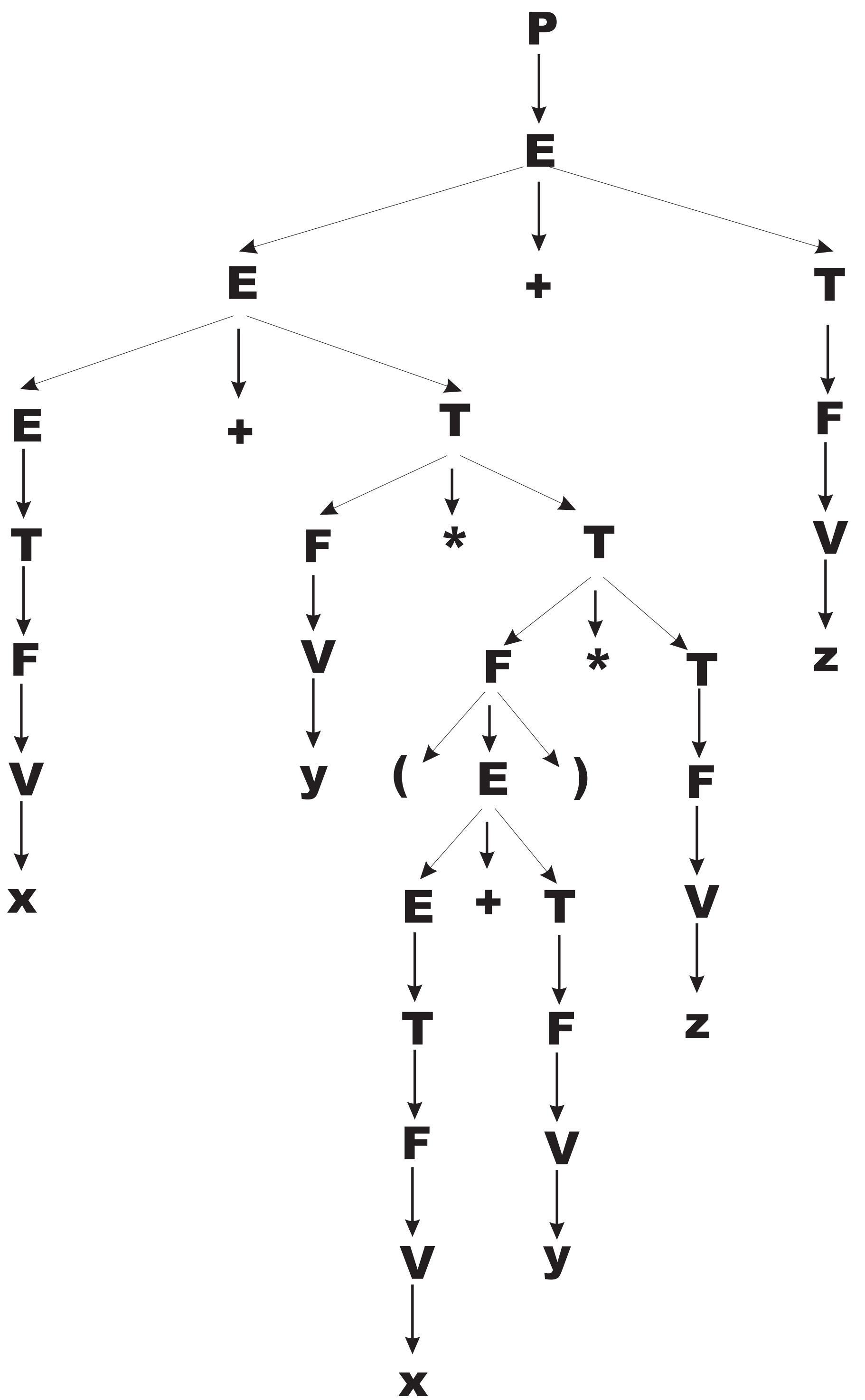
Continuăm cu mulţimea **regulilor(producţiilor)** gramaticii:

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.

Un exemplu

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



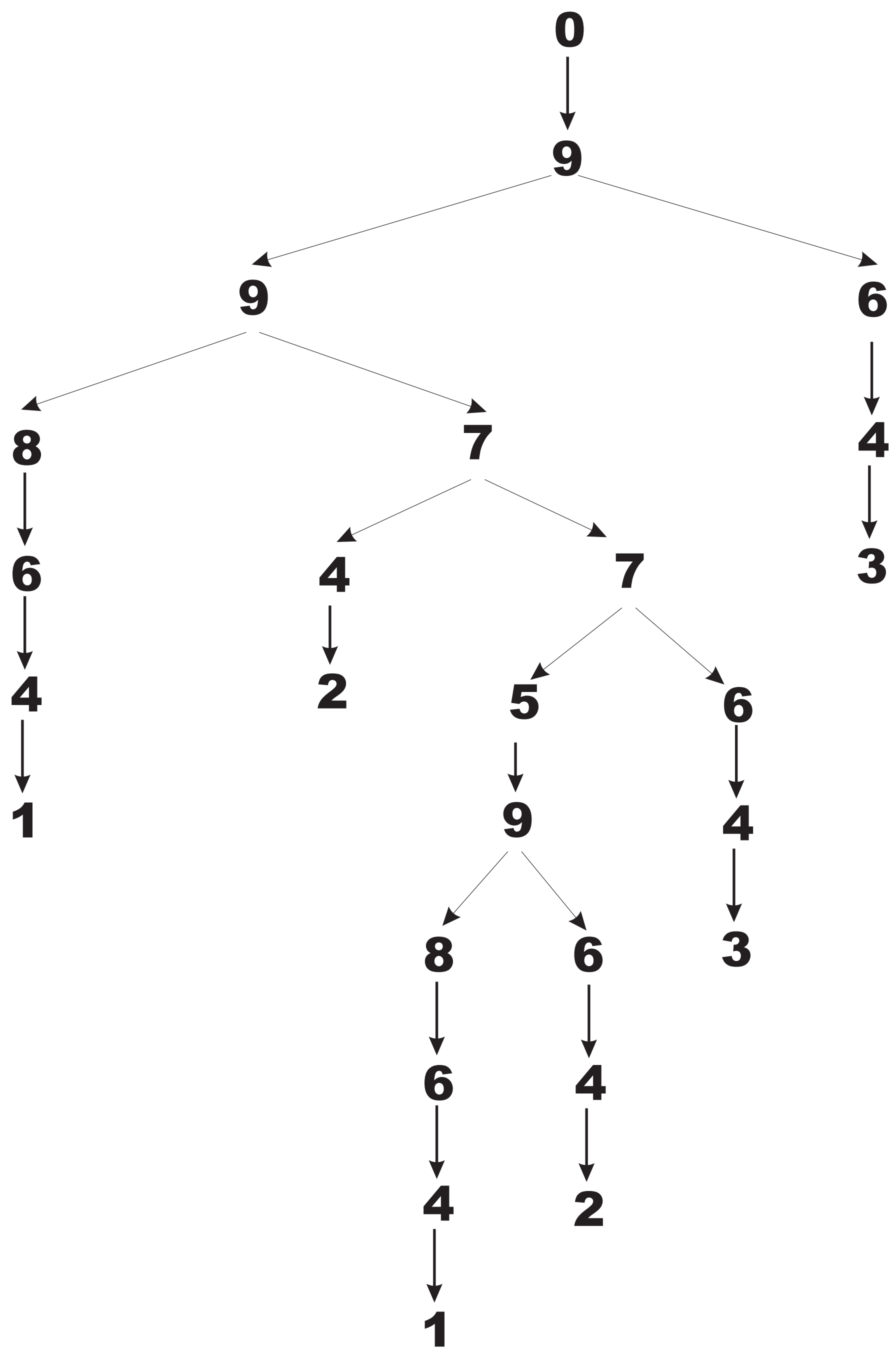
Pentru a reface expresia, arborele se parcurge în inordine.

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 $b = 6(4(3))$.



1.5.4 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 $\{x, y, z, +, *\}$, sau mai popular mulţimea şirurilor finite nevide de semne din $\{x, y, z, +, *\}$. Operaţiile sunt următoarele:

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

$1 = x$, $2 = y$ şi $3 = z$,

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

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 rescrierii expresiei în scrierea poloneză inversă.

Conform metodei semanticii algebrei iniţiale rezultă că C(A) este scrierea poloneză înversă a expresiei $x + y * (x + y) * z + z$.

Calculele sunt următoarele

C(a) $= 8(6[4(1)]) = x$ şi C(b) $= 6(4(3)) = z$

C(A) $= C$(0(9[9{a, 7(4[2], 7[5{9(a,6[4(2)])},b]) },b])) = 9[9{x, 7(y , 7[9(x,y) ,z]) },z] =

9[9{x, 7(y , 7[xy+,z]) },z] = 9[9{x, 7(y,$xy + z*$) },z] = 9[9{x, $yxy + z**$ },z] = 9[$xyxy + z**+$,z] =

$xyxy + z** + z+$.

Rezultatul final confirmă corectitudinea şi eficacitatea metodei.

1.5.5 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. Ori de câte ori se pune un element în stivă pointerul creşte automat cu o unitate. Ori de câte ori se scoate un element din stivă pointerul scade automat cu o unitate. Presupunem că la începutul execuţiei programului pointerul indică primul loc liber din stivă.

Presupunem că limbajul de asamblare include următoarele instrucţiuni:

Ad R	scoate valoarea din vârful stivei, o adună cu conţinutul lui R şi pune rezultatul în R
Mu R	scoate valoarea din vârful stivei, o inmulţeşte cu conţinutul lui R şi pune rezultatul în R
ld R	scoate valoarea din vârful stivei şi o pune în R
st R	valoarea din R este pusă în vârful stivei
print	tipăreşte valoarea din vârful stivei

Semantica unei expresii va fi programul care evaluează expresia şi imprimă rezultatul evaluării. Programul care evaluează o expresie are ca efect plasarea în vârful stivei a rezultatului evaluării acesteia şi avansarea pointerului. 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 vor fi notate cu litere greceşti.

Cele 10 operaţii ale algebrei semantice, corespunzătoare producţiilor sunt:

1_S	=	st x
2_S	=	st y
3_S	=	st z
$4_S(\alpha)$	=	α
$5_S(\alpha)$	=	α
$6_S(\alpha)$	=	α
$7_S(\alpha,\beta)$	=	α ; β ; ld R ; Mu R ; st R
$8_S(\alpha)$	=	α
$9_S(\alpha,\beta)$	=	α ; β ; ld R ; Ad R ; st R
$0_S(\alpha)$	=	α ; 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(A) este programul care evaluează şi tipăreşte rezultatul pentru expresia $x + y * (x + y) * z + z$.

Program

Prezentăm calculele care dovedesc afirmaţia de mai sus.

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

Notând c = 7[5{9(a,6[4(2)])},b] deducem

C(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 ; ld R ; Mu R ; st R =
1_S ; 2_S ; ld R ; Ad R ; st R ; 3_S ; ld R ; Mu R ; st R =
st x ; st y ; ld R ; Ad R ; st R ; st z ; ld R ; Mu R ; st R

Deoarece

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

deducem

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

st x ; st y ; st x ; st y ; ld R ; Ad R ; st R ; st z ; ld R ; Mu R ;
st R ; ld R ; Mu R ; st R ; ld R ; Ad R ; st R ; st z ; ld R ; Ad R ; st R ; print

Optimizare cod

Programul de mai sus poate fi simplificat. Observăm că grupul de instrucţiuni "st R ; ld R" are efect cumulat nul ceea ce conduce la programul

st x ; st y ; st x ; st y ; ld R ; Ad R ; st R ; st z ; ld R ; Mu R ;
Mu R ; Ad R ; st R ; st z ; ld R ; Ad R ; st R ; print

1.6 CONGRUENŢE şi ALGEBRE CÂT

Dacă $f : A \longrightarrow B$ este o funcţie relaţia $Ker(f)$ definită prin

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

se numeşte **echivalenţa nucleară** a lui f, sau mai pe scurt nucleul lui f.

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

1.6.1 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 1.6.1 O congruenţă $\sim = \{\sim_s\}_{s\in S}$ 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 $a_i \sim_{s_i} b_i$ în A_{s_i} pentru orice $1 \leq i \leq n$, atunci

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

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

Propoziţie 1.6.2 *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*

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

aşadar $A_\sigma(a_1, a_2, \dots, a_n)\ Ker(h_s)\ A_\sigma(b_1, b_2, \dots, b_n)$. Deci $Ker(h)$ este congruenţă.

Propoziţie 1.6.3 *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 $1 \leq i \leq n$. Rezultă că $a_i \sim^k b_i$ pentru orice $k \in K$ şi 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

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

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

Congruenţe de grupuri - numai pentru cei ştiu ce este un subgrup normal

Grupurile sunt privite ca algebre cu trei operatii: una binară $\times$, una unara * şi o constantă e. De ce? Există cel puţin două motive.

a)Motivul principal este dispariţia cuantificatorului existenţial din axiomele conceptului de grup

$$(\forall x \forall y \forall z) x \times (y \times z) = (x \times y) \times z$$

$$(\forall x) x \times e = x$$

$$(\forall x) x \times x* = e.$$

Aceasta plasează conceptul de grup printre structurile algebrice ecuaţional axiomatizabile, concept bine studiat în algebră.

b) Conceptele generale din teoria algebrelor multisortate dau prin particularizare conceptele uzuale din teoria grupurilor.

Conceptul de parte stabilă coincide cu cel de subgrup.

Conceptul de morfism de Σ-algebră coincide cu cel de morfism de grup.

Două condiţii se impun pentru conceptul de congruenţă

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: presupunând $a \sim b$, 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 1.6.4 *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.

3. Cele două treceri, de la un concept la celălalt şi reciproc, stabilesc o bijecţie între mulţimea congruenţelor unui grup şi mulţimea subgrupurilor sale normale. □

Congruenţe de inele - numai pentru cei care ştiu ce este un ideal

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.

Conceptul de parte stabilă coincide cu cel de subinel.

Conceptul de morfism de Σ-algebră coincide cu cel de morfism de inel.

Având în vedere cazul grupurilor este suficient să punem condiţia de congruenţă numai pentru cele două operaţii binare.

Propoziţie 1.6.5 *În orice inel comutativ conceptul de congruenţă este echivalent cu cel de ideal.*

Demonstraţie:

1 Dată o congruenţă clasa elementului neutru pentru adunare 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.

3. Cele două treceri, de la un concept la celălalt şi reciproc, stabilesc o bijecţie între mulţimea congruenţelor unui inel şi mulţimea idealelor sale. □

Concluzie

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

1.6.2 Algebre cât

Proprietatea de universalitate a mulţimii cât.

Propoziţie 1.6.6 *Fie $\sim$ o relaţie de echivalenţa în mulţimea A. Fie $A/\sim$ mulţimea cât şi $\rho : A \longrightarrow A/\sim$ 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 $\rho; f^{\#} = f$.

Demonstraţie: Definim funcţia $f^{\#}$ pentru orice $a \in A$ prin

$$f^{\#}(\rho(a)) = f(a).$$

Definiţia este corectă deoarece $b \sim a$ implică, folosind ipoteza $\sim \subseteq Ker(f)$, că $bKer(f)a$, adică $f(a) = f(b)$.
Egalitatea din enunţ $\rho; f^{\#} = f$ rezultă direct din definiţie.
Unicitatea rezultă din surjectivitatea funcţiei ρ de factorizare. Fie $g : A/\sim \longrightarrow B$ o funcţie cu proprietatea $\rho; g = f$. Rezultă că pentru orice $a \in A$

$$g(\rho(a)) = f(a).$$

Prin urmare $g(\rho(a)) = f^{\#}(\rho(a))$ pentru orice $a \in A$. Deoarece orice element din $A/\sim$ este de forma $\rho(a)$ cu $a \in A$ rezultă că $g = f^{\#}$. $\square$

Proprietatea de universalitate a algebrei cât.

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

$$A/\sim_\sigma(\rho(a_1), \rho(a_2), \ldots, \rho(a_n)) = \rho(A_\sigma(a_1, a_2, \ldots, a_n)).$$

Trebuie să probăm corectitudinea acestei definiţii. Cu notaţiile de mai sus presupunem pentru orice $1 \le i \le n$ că $b_i \in A_{s_i}$ şi $\rho(a_i) = \rho(b_i)$. Cu aceste ipoteze este suficient să probăm că $\rho(A_\sigma(a_1, a_2, \ldots, a_n)) = \rho(A_\sigma(b_1, b_2, \ldots, b_n))$. Pentru orice $1 \le i \le n$ din $\rho(a_i) = \rho(b_i)$ deducem $a_i \sim b_i$. Deoarece $\sim$ este congruenţă deducem $A_\sigma(a_1, a_2, \ldots, a_n) \sim A_\sigma(b_1, b_2, \ldots, b_n)$, deci $\rho(A_\sigma(a_1, a_2, \ldots, a_n)) = \rho(A_\sigma(b_1, b_2, \ldots, b_n))$.

Teorema 1.6.7 *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 $\rho; f^{\#} = f$.*

Demonstraţie: Din proprietatea de universalitate a mulţimii cât deducem existenţa unei unice funcţii $f^{\#} : A/\sim \longrightarrow B$ cu proprietatea $\rho; f^{\#} = f$. Observăm că $f_s^{\#}(\rho(a)) = f_s(a)$ pentru orice $s \in S$ şi $a \in A_s$. Mai trebuie să demonstrăm că funcţia $f^{\#}$ este un morfism de Σ-algebre.
Fie $s_1 s_2 \ldots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$ şi $a_i \in A_{s_i}$ pentru orice $1 \le i \le n$. Observăm că

$$f_s^{\#}(A/\sim_\sigma(\rho(a_1), \rho(a_2), \ldots, \rho(a_n))) = f_s^{\#}(\rho(A_\sigma(a_1, a_2, \ldots, a_n))) = f_s(A_\sigma(a_1, a_2, \ldots, a_n)) =$$

$$B_\sigma(f_{s_1}(a_1), f_{s_2}(a_2), \ldots, f_{s_n}(a_n)) = B_\sigma(f_{s_1}^{\#}(\rho(a_1)), f_{s_2}^{\#}(\rho(a_2)), \ldots, f_{s_n}^{\#}(\rho(a_n))).$$

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

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.

1.7 ALGEBRE PROIECTIVE

1.7.1 Proiectivitatea algebrelor libere

Definiţia 1.7.1 O Σ-algebră $\mathcal{P}$ se numeşte *proiectivă* dacă pentru orice Σ-morfism cu toate componentele surjective $e : \mathcal{A} \longrightarrow \mathcal{B}$ si pentru orice morfism $f : \mathcal{P} \longrightarrow \mathcal{B}$ există un morfism $g : \mathcal{P} \longrightarrow \mathcal{A}$ astfel încât $g; e = f$.

Propoziţie 1.7.2 *Orice Σ-algebră liberă este proiectivă.*

Demonstraţie: Fie X o mulţime S-sortată de variabile. Vom proba că Σ-algebra $T_\Sigma(X)$ liber generată de X este proiectivă.

Fie $\mathcal{A}$, $\mathcal{B}$ două Σ-algebre şi $e : \mathcal{A} \longrightarrow \mathcal{B}$ un morfism cu toate componentele surjective. Dacă $f : T_\Sigma(X) \longrightarrow \mathcal{B}$ 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 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. $\square$

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

1.7.2 Alte proprietăţi

Cei care nu sunt interesaţi de toate detaliile pot sări această secţiune, 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 1.7.3 *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 Alg_Σ 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 $m_s = n_s$. 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) $C\sigma(b1, \ldots, bn) = B\sigma(b1, \ldots, bn)$ dacă $bi \in B_{si}$ pentru orice $i \in [n]$;
2) $C\sigma(r_{s1}(b1), \ldots, r_{sn}(bn)) = r_s(B\sigma(b1, \ldots, bn))$ dacă $bi \in B_{si}$ pentru orice $i \in [n]$;
3) în celelalte situaţii operaţia $C\sigma$ 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 $B\sigma(b1, \dots, bn) \in h_s(A_s)$ şi deci $B\sigma(b1, \dots, bn) = r_s(B\sigma(b1, \dots, bn))$.

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 ι, $r : \mathcal{B} \longrightarrow \mathcal{C}$. 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 $s \in S$. $\Box$

Această propoziţie ne asigură că algebrele proiective corespund conceptului categorial cu acelaşi nume.

Lemă 1.7.4 *Orice subalgebră a unei algebre libere este algebră liberă.*

Demonstraţie: Într-o algebră definim relaţia

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

Observăm că în orice algebră liberă relaţia $\vdash$ este bine fondată, adică nu există şiruri infinite $\{a_n\}_{n\in N}$ cu proprietatea $a_n \vdash a_{n+1}$ pentru orice n natural.

Fie $\mathcal{P}$ o subalgebră a unei algebre libere $\mathcal{L}$. Relaţia $\vdash$ este bine fondată 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 $\mathcal{P}$. Prin urmare pentru orice $s \in S$

$$X_s = \{x \in P_s | (\forall\sigma)(\forall p_1, p_2, \dots, p_n) x \neq P_\sigma(p_1, p_2, \dots, p_n)\}$$

Vom dovedi ca $\overline{X}$, subalgebra generată de X în $\mathcal{P}$, 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 bine fondată 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. $\Box$

Propoziţie 1.7.5 (Andrei Popescu) *In $\mathcal{A}lg_\Sigma$ 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 $g; \epsilon = 1_P$. Obserevăm 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.

1.8 SPRE ABSTRACTIZAREA TIPURILOR DE DATE

1.8.1 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

$$(\forall X)l \stackrel{\circ}{=} r$$

unde l şi r sunt elemente din algebra liber generată de mulţimea X de variabile. Semnul $\stackrel{\circ}{=}$ este utilizat în loc de $=$, între două expresii, pentru a marca faptul că egalitatea poate fi şi falsă, fapt pentru care vom folosi denumirea de **egalitate formală** în loc de egalitate. În cazul multisortat, cei doi membri ai unei egalităţi formale trebuie să fie de acelaş sort. Uneori scriem $l \stackrel{\circ}{=}_s r$ pentru a marca sortul comun s al celor doi membri.

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. Ţinând cont că a da valori variabilelor din X în algebra $\mathcal{D}$ este echivalent cu a da un morfism de la $T_\Sigma(X)$ la $\mathcal{D}$, această idee intuitivă conduce la:

Definiţia 1.8.1 Axioma $(\forall X)l \stackrel{\circ}{=} r$ este satisfăcută în algebra $\mathcal{D}$ dacă şi numai dacă pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ este adevărată egalitatea $h(l) = h(r)$.

Dacă $(\forall X)l \stackrel{\circ}{=} r$ este satisfăcută în algebra $\mathcal{D}$, vom folosi notaţia

$$\mathcal{D} \models_\Sigma (\forall X)l \stackrel{\circ}{=} r$$

şi mai spunem că algebra $\mathcal{D}$ satisface $(\forall X)l \stackrel{\circ}{=} r$. $\square$

În continuare vom folosi pentru $(\forall X)l \stackrel{\circ}{=} r$ 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.

1.8.2 Ecuaţii condiţionate

În logica ecuaţională o axiomă poate fi 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

$$(\forall X)\ a \stackrel{\circ}{=}_s c \textbf{ if } H$$

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 $(\forall X)a \stackrel{\circ}{=}_s c$ **if** H fapt notat prin

$$\mathcal{D} \models_\Sigma (\forall X)\ a \stackrel{\circ}{=}_s c \textbf{ if } H$$

dacă şi numai dacă

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

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

1.8.3 Necesitatea utilizării cuantificatorilor în ecuaţii

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

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

$$\begin{tikzcd} a \arrow[r, "g"] & \mathit{bool} & \arrow[l, shift right, "F"'] \arrow[l, shift left, "T"] \end{tikzcd}$$

unde se vede că sortul rezultat al celor trei simboluri de operaţie g, F şi T este *bool*; operaţiile F şi T nu au argumente şi g are un singur argument de sort a.

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 $\{x\}$ pentru sortul a şi {g(x),F,T} pentru sortul *bool*.

Are loc relaţia $\mathcal{I} \models_\Sigma F \stackrel{\circ}{=} T$? Sau mai intuitiv: este egalitatea formală $F \stackrel{\circ}{=} T$ 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. $\mathcal{I} \models_\Sigma (\forall\emptyset) T \stackrel{\circ}{=} F$ dacă şi numai dacă pentru orice morfism $h : \mathcal{I} \longrightarrow \mathcal{I}$, $h_{\mathit{bool}}(F) = h_{\mathit{bool}}(T)$, ceea ce este **fals** deoarece $h_{\mathit{bool}}(F) = F \neq T = h_{\mathit{bool}}(T)$.

2. $\mathcal{I} \models_\Sigma (\forall x) T \stackrel{\circ}{=} F$ dacă şi numai dacă pentru orice morfism $h : T_\Sigma(\{x\}, \emptyset) \longrightarrow \mathcal{I}$, $h_{\mathit{bool}}(F) = h_{\mathit{bool}}(T)$, ceea ce este **adevărat** deoarece nu există nici un morfism $h : T_\Sigma(\{x\}, \emptyset) \longrightarrow \mathcal{I}$.

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

Contradicţia obţinută prin omiterea cuantificatorilor din faţa egalităţii formale $F \stackrel{\circ}{=} T$ arată că în logica ecuaţională multisortată prezenţa cuantificatorilor în ecuaţii este necesară.

In multe cazuri, de exemplu în limbajele de programare PROLOG, OBJ, Maude, CafeOBJ, etc cuantificatorii din faţa ecuaţiilor sunt omişi dar se presupune implicit că ecuaţiile sunt cuantificate prin mulţimea variabilelor care apar în ecuaţie. Egalitatea formală $T \stackrel{\circ}{=} F$ este implicit interpretată $(\forall\emptyset) T \stackrel{\circ}{=} F$.

1.8.4 In primul rând semantica

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

$$\mathit{Sen}(\mathcal{D}) = \{a \stackrel{\circ}{=}_s c \mid s \in S,\ a, c \in D_s\}$$

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

Să observăm că $\mathit{Sen}(\mathcal{D})$ 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 $\mathcal{D}$.

Având în vedere cele de mai sus nu vom mai face nici o distincţie între relaţiile lui $\mathcal{D}$ şi mulţimile de egalităţi formale între elementele lui $\mathcal{D}$.

Definiţia 1.8.2 O *ecuaţie condiţionată* este

$$(\forall X) l \stackrel{\circ}{=}_s r \text{ **if** } H$$

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 $(\forall X) l \stackrel{\circ}{=}_s r$ în loc de $(\forall X) l \stackrel{\circ}{=}_s r$ **if** $\emptyset$.

Definiţia 1.8.3 Algebra $\mathcal{D}$ *satisface* ecuaţia condiţionată $(\forall X) l \stackrel{\circ}{=}_s r$ **if** H, fapt notat prin

$$\mathcal{D} \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r \textbf{ if } H$$

dacă pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ pentru care $h_{s'}(u) = h_{s'}(v)$ pentru orice $u \stackrel{\circ}{=}_{s'} v \in H$, are loc egalitatea $h_s(l) = h_s(r)$. □

În cele ce urmează indicele Σ din $\models_\Sigma$ va fi omis. El va fi menţionat atunci când este pericol de confuzie. Observăm că $\mathcal{D} \models (\forall X) l \stackrel{\circ}{=}_s r$ 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 1.8.4 Spunem că algebra $\mathcal{D}$ satisface Γ sau că $\mathcal{D}$ este 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. □

1.8.5 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 această carte. În cazul global o propoziţie din $\mathcal{D}$ va fi scrisă $(\forall \mathcal{D}) a \stackrel{\circ}{=}_s c$ în loc de $a \stackrel{\circ}{=}_s c$. 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 [1, 16, 17]. Nici una nu prezintă stilul local.

Menţionăm că algebra $\mathcal{A}$ nu are legătură cu algebra $T_\Sigma(X)$ folosită în vreo axiomă $(\forall X) l \stackrel{\circ}{=}_s r$ **if** H din Γ. Menţionăm şi faptul 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.

1.8.6 Congruenţa semantică

Privind Γ ca o mulţime de axiome, ne interesează toate consecinţele ei semantice. Prin consecinţă semantică se înţelege orice fapt adevărat în orice Γ–algebră. Logica ecuaţională este interesată de consecinţele semantice care sunt scrise ca egalităţi formale.

Prin urmare ecuaţia $(\forall X) l \stackrel{\circ}{=}_s r$ este o consecinţă semantică a lui Γ, denumită şi tautologie sau propoziţie valabilă sau validă a logicii ecuaţionale, dacă ea este adevărată în orice Γ–algebră, adică $h_s(l) = h_s(r)$ pentru orice Γ–algebră $\mathcal{M}$ şi pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{M}$.

Pentru locarizare fixăm algebra $T_\Sigma(X)$ şi pentru generalizare o înlocuim cu o Σ–algebră arbitrară $\mathcal{A}$.

Prin urmare vom lucra într-o Σ-algebră $\mathcal{A}$ şi vom grupa într-o relaţie toate tautologiile(propoziţiile valide) logicii ecuaţionale din algebra $\mathcal{A}$.

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 $(\forall X)\, l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$ şi orice morfism $h :\ T_\Sigma(X) \longrightarrow \mathcal{A}$
$(\forall u \overset{\circ}{=}_t v \in H) h_t(u) \overset{\circ}{=}_t h_t(v)$ implică $h_s(l) \overset{\circ}{=}_s h_s(r)$.

Definiţia 1.8.5 O submulţime M a lui $Sen(\mathcal{A})$ se numeşte **închisă la substituţie** sau închisă la **Sub$_\Gamma$** dacă pentru orice $(\forall X) l \overset{\circ}{=}_s r$ **if** H în Γ şi orice $h : T_\Sigma(X) \longrightarrow \mathcal{A}$
$(\forall u \overset{\circ}{=}_t v \in H) h_t(u) \overset{\circ}{=}_t h_t(v) \in M$ implică $h_s(l) \overset{\circ}{=}_s h_s(r) \in M$.

Observăm că intersecţia unor mulţimi închise la substituţie este tot o mulţime închisă la substituţie.

Lemă 1.8.6 *Pentru orice morfism $f : \mathcal{A} \longrightarrow \mathcal{M} \models \Gamma$, $Ker(f)$ este închis la* **Sub$_\Gamma$**.

Demonstraţie: Fie $(\forall X) l \overset{\circ}{=}_s r$ **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 $u \overset{\circ}{=}_t v \in H$. Prin urmare $(h; f)_t(u) = (h; f)_t(v)$ pentru orice $u \overset{\circ}{=} v \in H$. Deoarece $h; f :\ T_\Sigma(X) \longrightarrow \mathcal{M} \models (\forall X) l \overset{\circ}{=}_s r$ **if** H rezultă că $(h; f)_s(l) = (h; f)_s(r)$, deci $h_s(l)\ Ker(f)\ h_s(r)$.□

Propoziţie 1.8.7 *Congruenţa 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 1.8.8 *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 $(\forall X) l \overset{\circ}{=}_s r$ **if** H în Γ. Fie $h : T_\Sigma(X) \longrightarrow \mathcal{A}/\sim$ un morfism astfel încât $h_t(u) = h_t(v)$ pentru orice $u \overset{\circ}{=}_t v \in H$. Deoarece orice algebră liberă este proiectivă rezultă că $T_\Sigma(X)$ este algebră proiectivă, prin urmare există un morfism $f : T_\Sigma(X) \longrightarrow \mathcal{A}$ astfel încât $f; \rho = h$. Pentru orice $u \overset{\circ}{=}_t v \in H$ 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 $h_s(l) = h_s(r)$. □

Arătaţi că $\equiv_\Gamma$ este cea mai mică congruenţă închisă la substituţie.

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

Teorema 1.8.9 $\mathcal{A}_\Gamma \models \Gamma$

Demonstraţie: Se aplică propoziţiile 1.8.7 şi 1.8.8. □

Teorema 1.8.10 *Pentru orice* Γ*-algebră* $\mathcal{B}$ *şi pentru orice morfism* $h: \mathcal{A} \longrightarrow \mathcal{B}$ *există şi este unic un morfism* $h^{\#}: \mathcal{A}_\Gamma \longrightarrow \mathcal{B}$ *astfel încât* $\eta; h^{\#} = h$.

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

Corolar 1.8.11 *Dacă* $\mathcal{A}$ *este* Σ*-algebră iniţială, atunci* $\mathcal{A}_\Gamma$ *este* Γ*-algebră iniţială.*

Corolar 1.8.12 *Pentru orice signatură* Σ *şi pentru orice mulţime* Γ *de ecuaţii condiţionate există o* Γ*-algebră iniţială.*

Demonstraţie: Se foloseşte existenţa Σ–algebrei initiale şi corolarul precedent.

1.8.7 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.

1.9 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.

1.9.1 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 : $\longrightarrow$ 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 1.9.1 *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. (vezi şi propoziţia 1.2.7)*

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 deocamdată să adune şi să înmulţească.

Adunarea poate fi introdusă prin declaraţia op _+_ : nat nat $\longrightarrow$ nat .

dar aceasta, singură, nu face decât să strice ceea ce am construit deja, deoarece de exemplu calculatorul interpretează $0+0$, deoarece nu ştie să adune ca un nou element diferit de 0, ceea ce evident nu dorim.

La cele de mai sus adăugăm prea cunoscutele axiome ale lui Peano.

op _+_ : nat nat $\longrightarrow$ nat .
var X Y : nat .
eq X + 0 = 0 .
eq X + s(Y) = s(X+Y) .
op _*_ : nat nat $\longrightarrow$ nat .
eq X * 0 = 0 .
eq X * s(Y) = (X*Y) + X .

Parantezele din ultimul rând sunt puse pentru ca maşina să înţeleagă că înmulţirea se efectuează inaintea adunării. Există metode mai eficiente care fac maşina să înţeleagă acest fapt, dar acesta nu este scopul prezentului text. În demonstraţia de mai jos nu mai punem aceste paranteze deoarece textul nu se mai adresează calculatorului.

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, adică se aplică numai de la stânga la dreapta.

Propoziţie 1.9.2 *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
$h(n *_N (m+1)) = h(n *_N m +_N n) = h(n *_N m) +_A h(n) = (h(n) *_A h(m)) +_A h(n) = h(n) *_A s_A(h(m)) = h(n) *_A h(m+1)$. $\square$

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.

Cred că este util să menţionăm aici că dat un tip de date nu este suficient să găsim o structură algebrică care să-l definească ca tip abstract de date. De exemplu ştim din algebră că numerele naturale formează un semiinel iniţial. Caracterizarea numerelor naturale ca semiinel iniţial nu ne mulţumeşte deoarece axiomele conceptului de semiinel nu ne dau un program(calculatorul este incapabil să facă operaţii cu numerele naturale folosind numai axiomele semiinelului). Axiomele lui Peano dau un program în programarea prin rescriere.

Chapter 2

RESCRIERI

2.1 TEORII DEDUCTIVE À la MOISIL

Fie E o mulţime de propoziţii. O *regulă de deducţie* pe E este o pereche (H, e) unde H este o submulţime *finită* a lui E (şi se numeşte ipoteza regulii) şi $e \in E$ (şi se numeşte concluzia regulii). Poate mai sugestiv o regulă (H, e) ar fi trebuit scrisă $H \longrightarrow e$ deoarece semnificaţia ei este "propoziţia e este o consecinţa a propoziţiilor din H". Mai menţionăm că în cazul particular $H = \emptyset$ regula $(\emptyset, e)$ spune că e este o axiomă.

O submulţime D a lui E se numeşte **închisă** la regula (H, e) dacă $H \subseteq D$ implică $e \in D$. In acest caz se mai spune că regula (H, e) este corectă pentru D deoarece regula nu ne scoate din D.

Fie R o mulţime de reguli de deducţie. O submulţime D a lui E se numeşte inchisă la R dacă şi numai dacă D este inchisă la orice regulă din R.

Definiţia închiderii la substituţie, dată în secţiunea 1.8.6, este un caz particular al definiţiei de mai sus.

Observaţia 2.1.1 *Dacă $D_i \subseteq E$ este inchisă la R pentru fiecare $i \in I$, atunci $\bigcap_{i \in I} D_i$ este închisă la R.*

Demonstraţie: Fie $(H, e) \in R$ astfel încât $H \subseteq \bigcap_{i \in I} D_i$. Pentru orice $i \in I$, deoarece D_i este inchisă la R şi $H \subset D_i$ deducem $e \in D_i$. Deci $e \in \bigcap_{i \in I} D_i$. $\Box$

Prin definiţie $[n] = \{1, 2, \ldots, n\}$ pentru orice număr natural n.

O propoziţie $e \in E$ poate fi demonstrată folosind R dacă şi numai dacă există o secvenţă finită de propoziţii $e_1 e_2 \ldots e_n$ astfel încât $e_n = e$ şi pentru orice $i \in [n]$, există $(H, e_i) \in R$ astfel încât $H \subseteq \{e_1, e_2, \ldots, e_{i-1}\}$. Vom spune că $e_1 e_2 \ldots e_n$ este o *demonstraţie* pentru $e_n = e$.

Observaţia 2.1.2 *Dacă α şi β sunt demonstraţii atunci $\alpha\beta$ este demonstraţie.*

Notăm cu $\mathbf{Teor}(R)$ mulţimea propoziţiilor care pot fi demonstrate folosind R.

Lemă 2.1.3 *Fie $D \subseteq E$. Dacă D este inchisă la R atunci $\mathbf{Teor}(R) \subseteq D$.*

Demonstraţie: Fie $e \in \mathbf{Teor}(R)$. Prin urmare există o secvenţă finită de propoziţii $e_1 e_2 \ldots e_n$ astfel încât $e_n = e$ şi pentru oricare i din [n] există $(H, e_i) \in R$ astfel încât $H \subseteq \{e_1, e_2, \ldots, e_{i-1}\}$.

Probăm prin inducţie că $e_i \in D$ pentru orice i din [n]. Ipoteza de inducţie este $(\forall k < i)\ e_k \in D$. Deoarece şirul $e_1 e_2 \ldots e_n$ este o demonstraţie există $(H, e_i) \in R$ cu $H \subseteq \{e_1, e_2, \ldots, e_{i-1}\}$. Din ipoteza de inducţie deducem $H \subseteq D$. Deoarece D este închisă la R rezultă că $e_i \in D$.

În particular pentru $i = n$ deducem $e \in D$.

Dar e a fost luată arbitrar în $\mathbf{Teor}(R)$, deci $\mathbf{Teor}(R) \subseteq D$. $\Box$

Această lemă ne furnizează o metodă prin care putem arăta că toate propoziţiile demonstrabile folosind R au o anumită proprietate şi anume este suficient să arătăm că mulţimea propoziţiilor cu proprietatea dată este închisă la R.

Teorema 2.1.4 **Teor**(R) *este cea mai mică mulţime(în raport cu* $\subseteq$*) care este închisă la* R*.*

Demonstraţie: Demonstrăm că **Teor**(R) este închisă la R. În acest scop presupunem $(H, e) \in R$, unde $H = \{f_1, \ldots, f_n\}$ şi $H \subseteq$ **Teor**(R). Pentru orice $i \in [n]$ din $f_i \in$ **Teor**(R) deducem existenţa unei demonstraţii $\alpha_i f_i$ unde α_i este un şir de propoziţii din E. Atunci $\alpha_1 f_1 \ldots \alpha_n f_n$ este o demonstraţie deoarece este o concatenare de demonstraţii şi pentru că $(\{f_1, \ldots, f_n\}, e) \in R$ deducem că $\alpha_1 f_1 \ldots \alpha_n f_n e$ este o demonstraţie pentru e. Deci $e \in$ **Teor**(R). Rezultă că **Teor**(R) este închisă la R. Din lema 2.1.3 rezultă că **Teor**(R) este cea mai mică în sensul incluziunii care este închisă la R. $\square$

Observaţia 2.1.1 ne spune că mulţimile închise la R formează, conform propoziţiei 1.3.8, o familie Moore. Pentru a pune în evidenţă operatorul de închidere asociat trebuie să ne referim la demonstraţii din ipoteze. Mai precis pentru $X \subseteq D$ adăugăm elementele lui X ca axiome $X' = \{(\emptyset, e) : e \in X\}$. Închiderea lui X este $Teor(R \cup X')$.

2.2 LOGICĂ ECUAŢIONALA

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

$$\equiv_\Gamma = \{a \stackrel{\circ}{=}_s b \in Sen(A) : \ (\forall M \models \Gamma)(\forall f : \ A \longrightarrow M) f_s(a) = f_s(b)\}.$$

Conform tradiţiei mai scriem

$$\models a \stackrel{\circ}{=}_s b \text{ dacă şi numai dacă } a \equiv_\Gamma b$$

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

2.2.1 Reguli de deducţie, corectitudine

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

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

Conform tradiţiei notăm prin $\vdash a \stackrel{\circ}{=}_s b$ faptul că egalitatea formală $a \stackrel{\circ}{=}_s b$ este demonstrabilă cu regulile de mai sus.

Observaţia 2.2.1 *Observăm că o mulţime de egalităţi formale este închisă la regula* **R** *dacă şi numai dacă este o relaţie reflexivă.*

Din această cauză **R** este numită regula reflexivităţii.

Observaţia 2.2.2 *Observăm că o mulţime de egalităţi formale este închisă la regula* **S** *dacă şi numai dacă este o relaţie simetrică.*

Din această cauză **S** este numită regula simetriei.

Observaţia 2.2.3 *Observăm că o mulţime de egalităţi formale este închisă la regula* **T** *dacă şi numai dacă este o relaţie tranzitivă.*

Din această cauză **T** este numită regula tranzitivităţii.

Observaţia 2.2.4 *Observăm că o mulţime de egalităţi formale este închisă la regula* $\mathbf{C}\Sigma$ *dacă şi numai dacă este o relaţie compatibilă cu toate operaţiile algebrei* $\mathcal{A}$.

Din această cauză $\mathbf{C}\Sigma$ este numită regula compatibilităţii cu operaţiile din Σ.

Teorema 2.2.5 *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 în conformitate cu propoziţia 1.8.7. $\square$

Corolar 2.2.6 $\vdash a \overset{\circ}{=}_s b$ *implică* $\models a \overset{\circ}{=}_s b$. $\square$

Demonstraţie: Congruenţa semantică, adică mulţimea egalităţilor formale cu proprietatea $\models a \overset{\circ}{=}_s b$, este închisă la $\mathbf{R_E}$ conform teoremei 2.2.5

Mulţimea egalităţilor formale demonstrabile, adică mulţimea egalităţilor formale cu proprietatea $\vdash a \overset{\circ}{=}_s b$, este cea mai mică mulţime închisă la $\mathbf{R_E}$ conform teoremei 2.1.4.

Prin urmare mulţimea egalităţilor formale cu proprietatea $\vdash a \overset{\circ}{=}_s b$ este inclusă în mulţimea egalităţilor formale cu proprietatea $\models a \overset{\circ}{=}_s b$, de unde rezultă concluzia.

2.2.2 Completitudine

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

$$a \sim_\Gamma b \Longleftrightarrow \vdash a \overset{\circ}{=}_s b.$$

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

Fie $\mathcal{A}_\Gamma$ câtul lui $\mathcal{A}$ prin $\sim_\Gamma$ şi fie $\eta_\mathcal{A}: \ \mathcal{A} \longrightarrow \mathcal{A}_\Gamma$ Σ-morfismul canonic de factorizare.

Observaţia 2.2.7 $\mathcal{A}_\Gamma \models \Gamma$

Demonstraţie: Deoarece $\mathbf{Sub}_\Gamma$ este în $\mathbf{R_E}$ rezultă că $\sim_\Gamma$ este închisă la substituţii prin urmare aplicând propoziţia 1.8.8 rezultă concluzia.

Teorema 2.2.8 *Teoremă de completitudine:* $\models a \overset{\circ}{=}_s b$ *implică* $\vdash a \overset{\circ}{=}_s b$.

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

În concluzie congruenţa $\sim_\Gamma$ şi congruenţa semantică $\equiv_\Gamma$ coincid. Prin urmare algebra $\mathcal{A}_\Gamma$ şi morfismul $\eta_\mathcal{A}$ coincid cu cele introduse într-un mod numai aparent diferit în secţiunea 1.8, teoremele 1.8.9 şi 1.8.10.

Proprietatea de universalitate a algebrei $\mathcal{A}_\Gamma$ demonstrată atunci, teorema 1.8.10, rămâne adevărată şi în noul context.

Următorul pas după o teoremă de completitudine care ne asigur u a existenţa unei demonstraţii pentru orice tautologie este de a găsi aceste demonstraţii. Informaticienii sunt ceva mai pretenţioşi deoarece doresc ca aceste demonstraţii să fie găsite de un calculator. Pentru acest scop rescrierea este foarte utilă.

2.3 RESCRIERE LOCALĂ

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 $2 * \bullet$, unde $\bullet$ este un semn special care arată locul în care se face rescrierea. Deoarece a doua rescriere se face la vârf contextul este $\bullet$.

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.

2.3.1 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

$$nr_x(\alpha) \text{ numărul de apariţii ale lui } x \text{ în } \alpha.$$

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, $\bullet$ o variabilă de sort s, $\bullet \notin A_s$. Considerăm algebra liber generată de $A \cup \{\bullet\}$, şi anume $T_\Sigma(A \cup \{\bullet\})$. Vom presupune fără a micşora generalitatea că

$$T_\Sigma(A) \subseteq T_\Sigma(A \cup \{\bullet\}).$$

Un element c din $T_\Sigma(A \cup \{\bullet\})$ se numeşte *context* dacă $nr_\bullet(c) = 1$. 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 $\bullet \leftarrow d : T_\Sigma(A \cup \{\bullet\}) \longrightarrow \mathcal{A}$ unicul morfism de Σ–algebre cu proprietatea $(\bullet \leftarrow d)(\bullet) = d$ şi $(\bullet \leftarrow d)(a) = a$ pentru orice $a \in A$. Menţionăm că restricţia lui $\bullet \leftarrow d$ la $T_\Sigma(A)$ este un morfism care evaluează în $\mathcal{A}$ o expresie în care elementele din A erau considerate variabile.

Pentru orice t din $T_\Sigma(A \cup \{\bullet\})$ şi $a \in A_s$, vom prefera să scriem

$$t[a]$$

în loc de $(\bullet \leftarrow a)(t)$, deoarece $(\bullet \leftarrow a)(t)$ este de fapt rezultatul înlocuirii lui $\bullet$ prin a în t şi al evalării în $\mathcal{A}$ al expresiei obţinute în $T_\Sigma(A)$.

Dacă în t nu apare $\bullet$, adică t este din Σ-algebra liber generată de A, atunci $t[a] = t[d]$ pentru orice $a, d \in A_s$, deoarece ambele sunt egale cu evaluarea în $\mathcal{A}$ a lui t. Egalitatea poate fi demonstrată printr-o simplă inducţie structurală.

Compunerea relaţiilor $R \subseteq A \times B$ şi $Q \subseteq B \times C$ este relaţia $R; Q \subseteq A \times C$ definită prin

$$R; Q = \{(a, c) \mid (\exists b \in B)(a, b) \in R \text{ şi } (b, c) \in Q\}.$$

Compunerea relaţiilor este asociativă.

Relaţiile între elementele aceleeaşi mulţimi A, adică părţile lui $A \times A$, formează cu operaţia de compunere un monoid al cărui element neutru este relaţia de egalitate. Pentru astfel de relaţii sunt definite puterile lor naturale. Prin definiţie R^0 este chiar relaţia de egalitate. Prin definiţie

$$R^* = \bigcup_{n \geq 0} R^n$$

este închiderea reflexivă şi tranzitivă a relaţiei R, adică R^* este cea mai mică relaţie reflexivă şi tranzitivă care include R. Menţionăm că * este operator de închidere în mulţimea părţilor lui $A \times A$.

Pentru o relaţie notată $\longrightarrow$ în loc de notaţia $\longrightarrow^*$ se preferă notaţia $\stackrel{*}{\longrightarrow}$.

2.3.2 Închiderea la contexte

Definiţia 2.3.1 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 $c[a]\,Q\,c[d]$.

Observăm că relaţia de egalitate este închisă la contexte.

Lemă 2.3.2 *Compunerea relaţiilor închise la contexte este închisă la contexte.*

Demonstraţie: Presupunem că relaţiile Q şi R sunt închise la contexte. Fie c un context.

Presupunem $a\,Q; R\,b$. Prin urmare există d astfel încât $a\,Q\,d$ şi $d\,R\,b$. Deoarece Q şi R sunt închise la contexte deducem $c[a]\,Q\,c[d]$ şi $c[d]\,R\,c[b]$, prin urmare $c[a]\,Q; R\,c[b]$. □

Propoziţie 2.3.3 *Dacă relaţia R este închisă la contexte, atunci închiderea ei reflexivă şi tranzitivă R^* este închisă la contexte.*

Demonstraţie: Presupunem că R este închisă la contexte. Folosind observaţia şi lema de mai sus deducem că pentru orice n natural relaţia R^n este închisă la contexte.

Fie c un context şi $a\,R^*\,b$. Prin urmare există n natural cu proprietatea $a\,R^n\,b$. Deducem că $c[a]\,R^n\,c[b]$, deci $c[a]\,R^*\,c[b]$. □

Introducem regula de deducţie

CAΣ $a \stackrel{\circ}{=}_{s_i} d$ implică $A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) \stackrel{\circ}{=}_s A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n)$ $(\forall \sigma \in \Sigma_{s_1 \ldots s_n, s})$, 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 2.3.4 O relaţie $\rho \subset A \times A$ se numeşte **compatibilă pe argumente cu operaţiile** algebrei $\mathcal{A}$ dacă este închisă la **CA**Σ.

Propoziţie 2.3.5 *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 contextul $c = \sigma(a_1, \ldots, a_{i-1}, \bullet, a_{i+1}, \ldots, a_n)$. Presupunem $(a, d) \in Q$ şi deducem din închiderea la contexte că $(c[a], c[d]) \in Q$. Observăm că

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

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

Reciproca se arată prin inducţie structurală în $T_\Sigma(A \cup \{\bullet\})$.

Pasul 0: $c = \bullet$. 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 $c' \in T_\Sigma(A \cup \{\bullet\})$ este un context şi $a_i \in T_\Sigma(A)$

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

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 2.3.6 Dacă Q este o relaţie pe A vom nota

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

Propoziţie 2.3.7 *$\longrightarrow_Q$ este inchiderea la contexte a lui Q.*

Demonstraţie: Trebuie să dovedim că $\longrightarrow_Q$ este cea mai mică relaţie închisă la contexte care include Q.

Pentru a dovedi că $\longrightarrow_Q$ este inchisă la contexte, vom prefera să arătăm că este compatibilă pe argumente cu oprerațiile. Fie $(c[a], c[d])$ în $\longrightarrow_Q$ unde $(a, d) \in Q$, $\sigma \in \Sigma_{t_1 t_2 \ldots t_n, t}$ un simbol de operaţie şi $a_i \in A_{t_i}$ 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 $\bullet$.

Presupunem $Q \subseteq R$ şi că R este închisă la contexte. Probăm că $\longrightarrow_Q \subseteq R$. Fie $(c[a], c[d])$ în $\longrightarrow_Q$ unde $(a, d) \in Q$. Incluziunea $Q \subseteq R$ implică $(a, d) \in R$. Deoarece R este închisă la contexte deducem că $(c[a], c[d])$ este în R. $\square$

Propoziţia precedentă ne asigură că $\longrightarrow_Q$ este operator de închidere.

2.3.3 Închiderea la preordini compatibile cu operaţiile

Preordine semnifică o relaţie reflexivă şi tranzitivă.

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

Remarcăm că **CΣ** şi **R** implică **CAΣ**.
Reciproca **CAΣ**, **R** şi **T** implică **CΣ** este demonstrată de lema următoare.

Lemă 2.3.8 *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* **CΣ**.

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 ipoteza **CA**Σ rezultă $A_\sigma(a_1)\,\rho_s\,A_\sigma(b_1)$.

Dacă $n \geq 2$ aplicând succesiv ipoteza **CA**Σ 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 $A_\sigma(a_1,a_2,\ldots,a_n)\,\rho_s\,A_\sigma(b_1,b_2,\ldots,b_n)$. $\Box$

Pentru a rezuma desenăm schema

$$\mathbf{C}\Sigma \underset{\mathbf{R,T}}{\overset{\mathbf{R}}{\rightleftarrows}} \mathbf{CA}\Sigma \Leftrightarrow \text{ închiderea la contexte}$$

Un comentariu privind regulile de deducţie **C**Σ şi **CA**Σ este bine venit. Regula **C**Σ este utilizată în logica ecuaţională. La rescrieri este preferabil să lucrăm cu regula **CA**Σ, deoarece ea este echivalentă cu închiderea la contexte. Din cele de mai sus rezultă că în prezenţa reflexibilităţii şi a tranzitivităţii regulile **CA**Σ şi **C**Σ sunt echivalente. Deoarece reflexivitatea şi tranzitivitatea sunt acceptate atât de logica ecuaţională cât şi de rescrieri, cele două reguli au aceeaşi putere.

Propoziţie 2.3.9 $\stackrel{*}{\longrightarrow}_Q$ *este cea mai mică preordine compatibilă cu operaţiile care include* Q.

Demonstraţie: Relaţia $\stackrel{*}{\longrightarrow}_Q$ este prin definiţie închiderea reflexivă şi tranzitivă a relaţiei $\longrightarrow_Q$. Ea este, prin definiţie, reflexivă şi tranzitivă. Este si închisă la contexte, conform propoziţiei 2.3.3 deoarece $\longrightarrow_Q$ este închisă la contexte.

Evident $Q \subseteq \longrightarrow_Q \subseteq \stackrel{*}{\longrightarrow}_Q$.

Fie R o preordine compatibilă cu operaţiile care include Q. Deoarece **C**Σ şi **R** implică **CA**Σ relaţia R este compatibilă pe argumente cu operaţiile, aşadar relaţia R este închisă la contexte, prin urmare aplicând propoziţia 2.3.7, deducem $\longrightarrow_Q \subseteq R$. Rezultă că $\stackrel{*}{\longrightarrow}_Q \subseteq R$ deoarece R este reflexivă şi tranzitivă. $\Box$

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

Notăm $u \Downarrow_Q v$ dacă există $a \in \mathcal{A}$ astfel încât $u \stackrel{*}{\longrightarrow}_Q a$ şi $v \stackrel{*}{\longrightarrow}_Q a$.

2.3.4 Γ–rescriere

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

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

SRew$_\Gamma$ Pentru orice $(\forall X)\, l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$ şi orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{A}$
$(\forall u \stackrel{\circ}{=}_t v \in H)(\exists d \in A_t) h_t(u) \stackrel{\circ}{=}_t d$ şi $h_t(v) \stackrel{\circ}{=}_t d$ implică
$c[h_s(l)] \stackrel{\circ}{=}_{s'} c[h_s(r)]$ pentru orice context $c \in T_\Sigma(A \cup \{\bullet\})_{s'}$.

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 $c = \bullet$.

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 $(\forall X)\, l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$ şi un morfism $h: T_\Sigma(X) \longrightarrow \mathcal{A}$ cu proprietatea că pentru orice $u \stackrel{\circ}{=}_{s'} v \in H$ există $d \in A_{s'}$ cu $h_{s'}(u) \stackrel{\circ}{=}_{s'} d$ şi $h_{s'}(v) \stackrel{\circ}{=}_{s'} d$. Pentru orice $u \stackrel{\circ}{=}_{s'} v \in H$ cu **S** deducem $d \stackrel{\circ}{=}_{s,} h_{s'}(v)$ şi apoi cu **T** obţinem $h_{s'}(u) \stackrel{\circ}{=}_{s'} h_{s'}(v)$. In final aplicăm $\mathbf{Sub}_\Gamma$ şi deducem $h_s(l) \stackrel{\circ}{=}_s h_s(r)$.

Pentru a rezuma desenăm schema

$$\begin{tikzcd} \mathbf{Sub}_\Gamma \arrow[r, shift left, "\mathbf{R}"] & \mathbf{Rew}_\Gamma \arrow[l, shift left, "\mathbf{S,T}"] \arrow[r, shift right, "\mathbf{CA\Sigma}"'] & \mathbf{SRew}_\Gamma \arrow[l, shift right] \end{tikzcd}$$

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

$Q_0 = \emptyset$

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

Pentru a ne convinge că şirul de mai sus este crescător putem demonstra prin inducţie că $(\forall n) Q_n \subseteq Q_{n+1}$. Evident $Q_0 = \emptyset \subseteq Q_1$. Presupunem $Q_n \subseteq Q_{n+1}$ şi probăm $Q_{n+1} \subseteq Q_{n+2}$. Fie $h_s(l) \stackrel{\circ}{=}_s h_s(r)$ în Q_{n+1}, adică există $(\forall Y) l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$ şi $h : T_\Sigma(Y) \longrightarrow \mathcal{A}$ cu proprietatea $(\forall u \stackrel{\circ}{=}_t v \in H) h_t(u) \Downarrow_{Q_n} h_t(v)$. Deoarece şirul $\Downarrow_{Q_n}$ este crescător deducem $h_t(u) \Downarrow_{Q_{n+1}} h_t(v)$, deci $h_s(l) \stackrel{\circ}{=}_s h_s(r)$ este în Q_{n+2}.

Deoarece $\longrightarrow_\bullet$ şi $\stackrel{*}{\longrightarrow}_\bullet$ sunt operatori de închidere deducem că şirurile $\longrightarrow_{Q_n}$ şi $\stackrel{*}{\longrightarrow}_{Q_n}$ sunt crescatoare.

Observaţia 2.3.10 *Dacă Γ nu conţine decât ecuaţii, adică în Γ nu există nici o ecuaţie condiţionată, observăm că şirul $\{Q_n\}_{n\geq 1}$ este constant, deci $Q = Q_1$.*

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 $a \stackrel{*}{\longrightarrow}_Q d$ implică existenţa unui n natural cu proprietatea $a \stackrel{*}{\longrightarrow}_{Q_n} d$.

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 2.3.11 $\stackrel{*}{\Longrightarrow}_\Gamma$ *este închisă la* $\mathbf{SRew}_\Gamma$.

Demonstraţie: Fie $(\forall Y) l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$, $h : T_\Sigma(Y) \longrightarrow \mathcal{A}$ morfism cu proprietatea $h_t(u) \Downarrow_Q h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$ şi un context $c \in T_\Sigma(A \cup \{\bullet\})_{s'}$. 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 $u \stackrel{\circ}{=}_t v \in H$ este finit există un n natural astfel încât $h_t(u) \Downarrow_{Q_n} h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$. Prin urmare $h_s(l) \stackrel{\circ}{=}_s h_s(r) \in Q_{n+1}$. Deoarece $h_s(l) \stackrel{\circ}{=}_s h_s(r) \in Q$ deducem $c[h_s(l)] \stackrel{*}{\Longrightarrow}_Q c[h_s(r)]$, adică $c[h_s(l)] \stackrel{*}{\Longrightarrow}_\Gamma c[h_s(r)]$. $\Box$

Propoziţie 2.3.12 $\stackrel{*}{\Longrightarrow}_\Gamma$ *este cea mai mică relaţie închisă la* **R**, **T**, $\mathbf{C\Sigma}$ *şi* $\mathbf{Rew}_\Gamma$.

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

Fie W o relaţie închisă la **R**, **T**, $\mathbf{C\Sigma}$ şi $\mathbf{Rew}_\Gamma$. Demonstrăm prin inducţie după n că $Q_n \subseteq W$.
Dacă $n = 0$ avem $Q_0 = \emptyset \subseteq W$.
Pentru $n \geq 1$ fie $h_s(l) \stackrel{\circ}{=}_s h_s(r) \in Q_n$, unde $(\forall Y) l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$, $h : T_\Sigma(Y) \longrightarrow \mathcal{A}$ şi $h_t(u) \Downarrow_{Q_{n-1}} h_t(v)$

pentru orice $u \overset{\circ}{=}_t v \in H$. Din ipoteza de inducţie $Q_{n-1} \subseteq W$. Cum W este închisă la $\mathbf{C\Sigma}$ şi $\mathbf{R}$ rezultă că W este închisă la $\mathbf{CA\Sigma}$, adică la contexte, deci $\longrightarrow_{Q_{n-1}} \subseteq W$. Folosind faptul că W este închisă la $\mathbf{R}$ şi $\mathbf{T}$ rezultă că $\overset{*}{\longrightarrow}_{Q_{n-1}} \subseteq W$, prin urmare $\Downarrow_{Q_{n-1}} \subseteq \Downarrow_W$.
Din $(\forall u \overset{\circ}{=}_t v \in H) h_t(u) \Downarrow_{Q_{n-1}} h_t(v)$ obţinem $(\forall u \overset{\circ}{=}_t v \in H) h_t(u) \Downarrow_W h_t(v)$. Folosind închiderea lui W la $\mathbf{Rew}_\Gamma$ rezultă că $h_s(l) \overset{\circ}{=}_s h_s(r) \in W$, deci $Q_n \subseteq W$.

Prin urmare $Q \subseteq W$ şi folosind propoziţia 2.3.9 deducem că $\overset{*}{\longrightarrow}_Q \subseteq W$.

2.3.5 Corectitudinea rescrierii

Teorema 2.3.13 *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. $\square$

Corolar 2.3.14 $\overset{*}{\Longrightarrow} \subseteq \equiv_\Gamma$. *(Corectitudinea rescrierii pentru congruenţa semantică)*

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

2.4 Relaţia de întâlnire, Forme normale

2.4.1 Confluenţă

Definiţia 2.4.1 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 2.4.2 *Dacă $\succ$ este o preordine confluentă pe mulţimea A, atunci relaţia $\downarrow$ definită prin*

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

şi denumită "întâlnire prin $\succ$" este cea mai mică echivalenţă pe A care include $\succ$.

Demonstraţie:

- Deoarece din $a \succ a$ şi $a \succ a$ deducem $a \downarrow a$. Deci relaţia $\downarrow$ ese reflexivă.
- 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 prin tranzitivitate 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 2.4.3 *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 cu $\mathbf{C}\Sigma$ deducem

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

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

2.4.2 Completitudinea întâlnirii prin recriere

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

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

Relaţia $\downarrow_\Gamma$ este denumită întâlnire prin rescriere.

Propoziţie 2.4.4 $\downarrow_\Gamma \subseteq \equiv_\Gamma$.

Demonstraţie: Din $a \downarrow_\Gamma d$ deducem existenţa lui c astfel încât: $a \overset{*}{\Rightarrow}_\Gamma c$ şi $d \overset{*}{\Rightarrow}_\Gamma c$. Din aceasta utilizând corectitudinea rescrierii faţă de congruenţa semantică deducem că $a \equiv_\Gamma c$ şi $d \equiv_\Gamma c$ deci $a \equiv_\Gamma d$. $\Box$

Aceasta dovedeşte corectitudinea lui $\downarrow_\Gamma$ în raport cu congruenţa semantică. Pentru a demonstra şi completitudinea sa, adică incluziunea inversă, avem nevoie să presupunem că $\overset{*}{\Rightarrow}_\Gamma$ este confluentă. Folosind observaţia 2.4.3 deducem că $\downarrow_\Gamma$ este o congruenţă.

Lemă 2.4.5 *Dacă $\overset{*}{\Rightarrow}_\Gamma$ este confluentă, atunci congruenţa $\downarrow_\Gamma$ este închisă la substituţie.*

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

Fie $\mathcal{A}^\Gamma$ factorizarea lui $\mathcal{A}$ prin $\downarrow_\Gamma$ şi fie $\tau : \mathcal{A} \longrightarrow \mathcal{A}^\Gamma$ morfismul de factorizare. Doarece congruenţa $\downarrow_\Gamma$ este închisă la substituţie deducem din propoziţia 1.8.8 că $\mathcal{A}^\Gamma \models \Gamma$.

Teorema 2.4.6 *Dacă $\overset{*}{\Rightarrow}_\Gamma$ este confluentă, atunci $\downarrow_\Gamma = \equiv_\Gamma$.*

Demonstraţie: Folosind propoziţia 2.4.4 avem de demonstrat doar $\equiv_\Gamma \subseteq \downarrow_\Gamma$.

Fie $a \equiv_\Gamma c$. Deoarece $\mathcal{A}^\Gamma \models \Gamma$ şi $\tau : \mathcal{A} \longrightarrow \mathcal{A}^\Gamma$ 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 $\downarrow_\Gamma$. $\Box$

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 $a \downarrow_\Gamma c$?"

2.4.3 Forme normale

La mulţimi

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

Definiţia 2.4.7 Elementul $n \in A$ se numeşte **o formă normală** pentru relaţia $\succ$ pe mulţimea A dacă

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

In limbajul teoriei mulţimilor partial ordonate, o formă normală este numită element minimal. Unele texte folosesc "ireductibil" sau formă "canonică" în loc de formă "normală".

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 2.4.8 *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 2.4.9 *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 2.4.10 *Funcţia $fn: A \longrightarrow N$ 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 $fn(a) = fn(d)$. $\square$

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 complect şi independent de reprezentanţi pentru $\downarrow$. Independent înseamnă că nu există două forme normale echivalente şi diferite, adică $n \downarrow n'$ implică $n = n'$, oricare ar fi formele normale n şi n'. Complect înseamnă că orice element este echivalent cu o formă normală, adică pentru orice $a \in A$ există o formă normală $n \in N$ cu proprietatea $a \downarrow n$.

Observaţia 2.4.11 *Există o unică bijecţie $b: A/_\downarrow \longrightarrow N$ astfel încât $\rho; b = fn$ unde $\rho : A \longrightarrow A/_\downarrow$ este funcţia naturală de factorizare.*

Demonstraţie: Din proprietatea 1.6.6 de universalitate a mulţimii cât rezultă existenţa şi unicitatea funcţiei $b : A/_\downarrow \longrightarrow N$ cu proprietatea $\rho; b = fn$.

Probăm că funcţia b este surjectivă. Fie $n \in N$. Deoarece $fn(n) = n$ deducem $b(\rho(n)) = n$, deci funcţia b este surjectivă.

Probăm că este injectivă. Presupunem că $b(\rho(a)) = b(\rho(c))$. Rezultă că $fn(a) = fn(c)$. Prin urmare $a \downarrow c$. Deci $\rho(a) = \rho(c)$. $\square$

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 $1 \le i \le k$.

Observaţia 2.4.12 *$fn: \mathcal{A} \longrightarrow \mathcal{N}$ 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

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

Propoziţie 2.4.13 *Dacă în Σ-algebra $\mathcal{A} = (A_s, A_\sigma)$, $\succ$ este o preordine multi-sortată compatibilă cu operaţiile algebrei şi cu proprietatea* **FN!**, *atunci algebrele $\mathcal{A}/_\downarrow$ şi $\mathcal{N}$ sunt izomorfe.*

Demonstraţie: Notăm ca mai sus cu $\rho : \mathcal{A} \longrightarrow \mathcal{A}$ morfismul de factorizare. Din proprietatea de universalitate a algebrei cât 1.6.7 rezultă existenţa şi unicitatea morfismului $h : \mathcal{A}/_\downarrow \longrightarrow \mathcal{N}$ cu proprietatea $\rho; h = fn$. Ca în cazul mulţimilor rezultă că morfismul h este bijectiv. Din propoziţia 1.1.8 rezultă că h este izomorfism.

Teorema 2.4.14 *Dacă $\overset{*}{\Rightarrow}_\Gamma$ satisface axioma* **FN!**, *atunci algebra formelor normale este isomorfă cu $\mathcal{A}^\Gamma$.*

2.4.4 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 2.4.15 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 $a_n \neq a_{n+1}$. $\Box$

Remarcăm că terminarea, concept venit din informatică, coincide cu buna fondare, concept provenit din lumea relaţiilor sau din teoria axiomatică a mulţimilor.

Observaţia 2.4.16 *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. $\Box$

Definiţia 2.4.17 Dacă preordinea $\succ$ este confluentă şi are proprietatea de terminare, atunci $< A, \succ >$ se numeşte **canonică**. $\Box$

Propoziţie 2.4.18 *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 $n' = n"$. $\Box$

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 $a \equiv_\Gamma c$, adică $a \downarrow_\Gamma c$, este suficient conform observaţiei 2.4.10 să rescriem a si c in forma lor normală $fn(a)$ şi $fn(c)$ pentru ca apoi să verificăm egalitatea $fn(a) = fn(c)$.*

2.5 RESCRIERE ÎN SUBTERMENI

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.

2.5.1 Completitudinea rescrierii în subtermeni

Propoziţiile 2.3.11 şi 2.3.12 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 2.5.1 *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 definiţia lui R_Γ deducem că $R_\Gamma \subseteq \overset{*}{\Rightarrow}_\Gamma$.

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 că R_Γ este închisă la $\mathbf{CA}\Sigma$, adică pentru fiecare $\sigma \in \Sigma_{s_1\ldots s_n,s'}$ şi $a_i \in A_{s_i}$ că $a\, R_\Gamma\, d$ implică

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

Demonstrăm că mulţimea

$$D = \{a\, R_\Gamma\, d \mid \sigma \in \Sigma_{s_1 s_2\ldots s_n,s}; (\forall a_j \in A_{s_j}) A_\sigma(a_1,\ldots,a_{i-1},a,a_{i+1},\ldots,a_n)\, R_\Gamma\, A_\sigma(a_1,\ldots,a_{i-1},d,a_{i+1},\ldots,a_n)\}$$

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 $A_\sigma(a_1,\ldots,a_{i-1},a,a_{i+1},\ldots,a_n)\ R_\Gamma\ A_\sigma(a_1,\ldots,a_{i-1},u,a_{i+1},\ldots,a_n)$ şi $A_\sigma(a_1,\ldots,a_{i-1},u,a_{i+1},\ldots,a_n)\ R_\Gamma\ A_\sigma(a_1,\ldots,a_{i-1},d,a_{i+1},\ldots,a_n)$ deducem $A_\sigma(a_1,\ldots,a_{i_1},a,a_{i+1},\ldots,a_n)\, R_\Gamma\, A_\sigma(a_1,\ldots,a_{i-1},d,a_{i+1},\ldots,a_n)$. Prin urmare $(a,d) \in D$. Deci D este tranzitivă.

Probăm că D este închisă la $\mathbf{SRew}_\Gamma$. Fie $(\forall X) l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$, un morfism $h : T_\Sigma(X) \to \mathcal{A}$ cu proprietatea că pentru orice $u \overset{\circ}{=}_t v \in H$. 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 $c \in T_\Sigma(A \cup \{\bullet\})_{s'}$ 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 $u \overset{\circ}{=}_t v \in H$ 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 $c' = \sigma(a_1,\ldots,a_{i-1},c,a_{i+1},\ldots,a_n)$ este un context.
Din $c'[h_s(l)] = A_\sigma(a_1,\ldots,a_{i-1},c[h_s(l)],a_{i+1}\ldots,a_n)$ şi $c'[h_s(r)] = A_\sigma(a_1,\ldots,a_{i-1},c[h_s(r)],a_{i+1},\ldots,a_n)$ obţinem

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

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

În concluzie $R_\Gamma = \overset{*}{\Rightarrow}_\Gamma$. $\square$

2.5.2 Rescriere într-un pas

Fie c în $T_\Sigma(A \cup \{\bullet\})_{s'}$ un context. Fie $(\forall X) l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$ şi fie $h : T_\Sigma(X) \to \mathcal{A}$ un morfism astfel încât pentru orice $u \stackrel{\circ}{=}_t v \in H$ 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 2.5.2 *Rescrierea este închiderea reflexivă şi 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 $(\forall X) l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$, $h : T_\Sigma(X) \to \mathcal{A}$ un morfism astfel încât pentru orice $u \stackrel{\circ}{=}_t v \in H$ 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 $u \stackrel{\circ}{=}_t v \in H$ 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 $c[h(l)] \Rightarrow c[h(r)]$, deci $c[h(l)] \stackrel{*}{\Rightarrow} c[h(r)]$ 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 $\Rightarrow$.

2.5.3 Consideraţii metodologice

Pentru a demonstra că $a \equiv_\Gamma b$ este suficient, în ipotezele de mai sus, 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

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

Pentru a face rescrierea este suficient să arătăm că $h_t(u) \downarrow_\Gamma h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$. 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 alte rescrieri, ca în cazul general. Acest fapt explică de ce, în acest caz metoda backtracking nu este folosită în implementarea rescrierii.

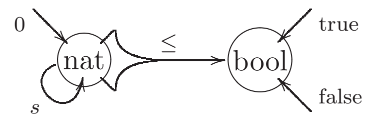
Remarcăm diferenţa esenţială dintre o mulţime Γ de axiome şi un program. Un program este o mulţime de axiome care verifică nişte condiţii şi care sunt scrise respectând sintaxa unui limbaj de programare declarativă.

Am pus deja în evidenţă nişte condiţii suficiente pe care trebuie să le îndeplinească o mulţime Γ de axiome pentru a deveni program: rescrierea generată trebuie să fie confluentă şi să aibă proprietatea de terminare.

Practic, de exemplu, întro ecuaţie din Γ este interzis ca în membrul drept să apară o variabilă care nu apare în membrul stâng, deoarece în caz contrar rescrierea nu are proprietatea de terminare.

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 $s(s(0)) \leq s(s(z)) \Rightarrow_\Gamma true$.

2.6 UNIFICARE

In semantica operaţională a limbajelor de specificaţie rescrierea joacă un rol primordial. Pentru a rescrie un subtermen $a \in T_\Sigma(Y)$ al termenului de rescris cu ajutorul unei reguli se pune problema identificării lui a cu imaginea membrului stâng $l \in T_\Sigma(X)$ al unei reguli din Γ printr-o substituţie(morfism) de la $T_\Sigma(X)$ la $T_\Sigma(Y)$. Această problemă de potrivire(matching în engleză) se poate rezolva cu ajutorul algoritmului de unificare. Mai precis se caută a se unifica l cu a în care toate variabilele din a sunt considerate constante. Dacă în l şi a există variabile comune, aceste variabile comune pot fi înlocuite în axioma utilizată din Γ prin variabile noi. Acest fapt nu restrânge generalitatea deoarece variabilele din axiome sunt cuantificate universal fapt ce permite redenumirea lor.

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

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

De fapt mulţimea finită de egalităţii formale este privită ca un sistem de ecuaţii. Şi de aici poate proveni denumirea de ecuaţie folosită ca un sinonim pentru egalităţile formale. Acest subcapitol descrie modul de rezolvare a unui astfel de sistem chiar în algebra liberă din care provin termenii ecuaţiilor. Cuvântul unificator înseamnă de fapt soluţie în algebra liberă din care provin termenii ecuaţiilor.

Algoritmul prezentat aici este ales pentru simplcitatea sa. Menţionăm existenţa unor algoritmi mai eficienţi.

2.6.1 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 $t \stackrel{\circ}{=} t$.

2. **Descompune.** Orice ecuaţie din lista de rezolvat de forma

$$f(a_1, a_2, \ldots, a_n) \stackrel{\circ}{=} f(b_1, b_2, \ldots, b_n)$$

se înlocuieşte cu ecuaţiile $a_1 \stackrel{\circ}{=} b_1$, $a_2 \stackrel{\circ}{=} b_2$, ..., $a_n \stackrel{\circ}{=} b_n$,

3. **Elimină.** Orice ecuaţie din lista de rezolvat, de forma $x \stackrel{\circ}{=} t$ sau $t \stackrel{\circ}{=} x$ în care x este o variabilă care nu apare în t este mutată sub forma $x \stackrel{\circ}{=} t$ 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 $f(a_1, a_2, \ldots, a_n) \stackrel{\circ}{=} g(b_1, b_2, \ldots, b_m)$ cu $f \neq g$,

2) Existenţa în lista de rezolvat a unei ecuaţii de forma $x \stackrel{\circ}{=} t$ sau $t \stackrel{\circ}{=} x$ 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 ne furnizează cel mai general unificator.

2.6.2 Terminare

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

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

2.6.3 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

$$s(f(a_1, a_2, \ldots, a_n)) \stackrel{\circ}{=} s(f(b_1, b_2, \ldots, b_n))$$

este echivalentă cu

$$f(s(a_1), s(a_2), \ldots, s(a_n)) \stackrel{\circ}{=} f(s(b_1), s(b_2), \ldots, s(b_n))$$

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

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

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

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

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

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

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

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

Faptul poate fi dovedit prin inducţie.

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

Fie $x \stackrel{\circ}{=} t$ 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. Prin urmare, variabilele din membrul stâng al noii liste soluţie sunt diferite între ele.

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 $x_i \stackrel{\circ}{=} t_i$ 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 $U; U = U$.

Notând $V = X - \{x_1, x_2, \dots, x_k\}$ observăm că $V \subseteq X$ şi că

$$U : T_\Sigma(X) \longrightarrow T_\Sigma(V)$$

este un Σ–morfism cu proprietatea $U(v) = v$ pentru orice $v \in V$.

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

2.7 LOCAL CONFLUENŢĂ

Dacă terminarea unui program poate fi chiar şi o problemă deschisă, problema confluenţei are soluţie, adică se pot scrie programe care pentru o mulţime dată de reguli Γ să constate, în ipoteza terminării, dacă Γ-rescrierea este confluentă. De fapt se probează local confluenţa care împreună cu terminarea implică confluenţa.

Reamintim că relaţia de rescriere este confluentă dacă

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

Definiţia 2.7.1 Spunem că rescrierea este **local confluentă** dacă

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

Reamintim 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 2.7.2 *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 $a \stackrel{*}{\Longrightarrow} n$. Vom demonstra unicitatea formei normale fapt ce implică confluenţa. Notăm cu N mulţimea formelor normale. Fie M mulţimea elementelor care se pot rescrie în cel puţin două forme normale.

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

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

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

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

Probăm unicitatea formei normale. Raţionând prin absurd presupunem 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. □

2.8 RESCRIERE MODULO ECUATII

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 necondiţionate care în loc de a fi folosite ca axiome vor fi utilizate pentru rescrierea modulo ecuaţii.

2.8.1 Motivare semantică

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

Propoziţie 2.8.1 *Fie $m : \mathcal{A} \longrightarrow \mathcal{B}$ un morfism din categoria Σ–algebrelor cu toate componentele injective. Dacă $\mathcal{B} \models \Gamma$, atunci $\mathcal{A} \models \Gamma$.*

Demonstraţie: Fie $(\forall X) l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$. Fie $h : T_\Sigma(X) \longrightarrow \mathcal{A}$ un morfism astfel încât $h_t(u) = h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$. Atunci $h; m : T_\Sigma(X) \longrightarrow \mathcal{B}$ este un morfism şi

$$(h; m)_t(u) = m_t(h_t(u)) = m_t(h_t(v)) = (h; m)_t(v)$$

pentru orice $u \stackrel{\circ}{=}_t v \in H$. 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 componenta m_s a morfismului m este injectivă, deci $h_s(l) = h_s(r)$.

Aşadar $\mathcal{A}$ satisface orice $(\forall X) l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$, adică $\mathcal{A} \models \Gamma$. □

Corolar 2.8.2 *Orice subalgebră a unei Γ–algebre este Γ–algebră.*

Propoziţie 2.8.3 *Fie $e : \mathcal{A} \longrightarrow \mathcal{B}$ un morfism din categoria Σ–algebrelor cu toate componentele surjective şi E o mulţime de ecuaţii necondiţionate. Dacă $\mathcal{A} \models E$, atunci $\mathcal{B} \models E$.*

Demonstraţie: Fie $(\forall X) l \stackrel{\circ}{=}_s r \in E$ şi $h : T_\Sigma(X) \longrightarrow \mathcal{B}$ un morfism. Deoarece Σ-algebra liberă $T_\Sigma(X)$ este *proiectivă*, conform propoziţiei 1.7.2, rezultă că există un morfism $g : T_\Sigma(X) \longrightarrow \mathcal{A}$ astfel încât $g; e = h$. Atunci

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

Deci $\mathcal{B}$ satisface orice ecuaţie $(\forall X) l \stackrel{\circ}{=}_s r$ din E, adică $\mathcal{B}$ satisface E. $\Box$

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 2.8.4 *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 morfismul $j : \mathcal{A}_E \longrightarrow \mathcal{A}_{E,\Gamma}$ are toate componentele surjective şi $\mathcal{A}_E \models E$ rezultă din propoziţia 2.8.3 că $\mathcal{A}_{E,\Gamma} \models E$, deci $\mathcal{A}_{E,\Gamma} \models \Gamma \cup E$.

Pentru restul demonstraţiei puteţi utiliza figura 2.1.

$$\begin{tikzcd}
\mathcal{A} \arrow[r, "\rho"] \arrow[dr, "h"'] & \mathcal{A}_E \arrow[r, "j"] \arrow[d, "h'"] & \mathcal{A}_{E,\Gamma} \arrow[dl, "h''"] \\
 & \mathcal{B} &
\end{tikzcd}$$

Figure 2.1:

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 $\rho; j$. $\Box$

Propoziţie 2.8.5 $a \equiv_{\Gamma \cup E} c \;\Leftrightarrow\; \rho_s(a) \equiv_\Gamma \rho_s(c)$ *pentru oricare $a, c \in A_s$.*

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 $\rho; j = t; i^{-1}$ deducem $j_s(\rho_s(a)) = j_s(\rho_s(c))$ deci $\rho_s(a) \equiv_\Gamma \rho_s(c)$. $\Box$

Propoziţia 2.8.5 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 2.8.5 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 $\mathcal{A}_E$.

2.8.2 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 $a \overset{\circ}{=}_s b$

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.

Preliminarii

Fie un morfism $h: \mathcal{A} \longrightarrow \mathcal{B}$ şi $\bullet$ o variabilă de sort s.
Fie $h^\bullet: T_\Sigma(A \cup \{\bullet\}) \longrightarrow T_\Sigma(B \cup \{\bullet\})$ unicul morfism definit pentru orice $x \in A \cup \{\bullet\}$ prin

$$h^\bullet(x) = \begin{cases} \bullet & \text{dacă } x = \bullet \\ h(x) & \text{dacă } x \in A \end{cases}.$$

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

Pentru oricare a din $\mathcal{A}$

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

Pentru orice c din $T_\Sigma(A \cup \{\bullet\})$ are loc egalitatea

$$h(c[a]) = h^\bullet(c)[h(a)].$$

Dacă în plus morfismul $h: \mathcal{A} \longrightarrow \mathcal{B}$ este surjectiv pe componente, atunci morfismul $h^\bullet: T_\Sigma(A \cup \{\bullet\}) \longrightarrow T_\Sigma(B \cup \{\bullet\})$ este şi el surjectiv pe componente fapt care se demonstrează prin inducţie structurală probând că pentru orice $c' \in T_\Sigma(B \cup \{\bullet\})$ există $c \in T_\Sigma(A \cup \{\bullet\})$ cu proprietatea $h^\bullet(c) = c'$. In plus pentru fiecare context $c' \in T_\Sigma(B \cup \{\bullet\})$ există un context $c \in T_\Sigma(A \cup \{\bullet\})$ astfel încît $h^\bullet(c) = c'$.

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 $h: \mathcal{A} \longrightarrow \mathcal{B}$.

Propoziţie 2.8.6 *Fie un morfism $h: \mathcal{A} \longrightarrow \mathcal{B}$. Dacă $a \overset{h}{\Rightarrow} d$ atunci $h(a) \overset{*}{\Rightarrow}{}^{\mathcal{B}}_\Gamma h(d)$.*

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

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

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

$\mathbf{Cl}_{Ker(h)}$. Fie $aKer(h)_s d$. Prin urmare $h_s(a) = h_s(d)$. Din reflexivitate $h_s(a) \overset{*}{\Rightarrow}{}^{\mathcal{B}}_{\Gamma} h_s(d)$, deci $a \overset{\circ}{=} d \in D$.

T. Fie $a \overset{\circ}{=}_s b \in D$ şi $b \overset{\circ}{=}_s c \in D$. Prin urmare $h_s(a) \overset{*}{\Rightarrow}{}^{\mathcal{B}}_{\Gamma} h_s(b)$ şi $h_s(b) \overset{*}{\Rightarrow}{}^{\mathcal{B}}_{\Gamma} h_s(c)$. Datorită tranzitivităţii $h_s(a) \overset{*}{\Rightarrow}{}^{\mathcal{B}}_{\Gamma} h_s(c)$, deci $a \overset{\circ}{=} c \in D$.

$\mathbf{SRew}_\Gamma$. Fie $(\forall X)l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$ şi $f:\ T_\Sigma(X) \longrightarrow \mathcal{A}$ un morfism astfel încît pentru orice $u \overset{\circ}{=}_t v$ există $a_{uv} \in A_t$ cu proprietăţile $f_t(u) \overset{\circ}{=}_t a_{uv} \in D$ şi $f_t(v) \overset{\circ}{=}_t a_{uv} \in D$. 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 $T_\Sigma(A \cup \{\bullet\})$, deoarece $\overset{*}{\Rightarrow}{}^{\mathcal{B}}_{\Gamma}$ este închis la $\mathbf{SRew}_\Gamma$, folosind contextul $h^\bullet(c)$ şi morfismul $f;h : T_\Sigma(X) \longrightarrow \mathcal{B}$ obţinem

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

Deoarece

$$h^\bullet(c)[(f;h)_s(l)] = h^\bullet(c)[h_s(f_s(l))] = h(c[f_s(l)])$$

şi analog

$$h^\bullet(c)[(f;h)_s(r)] = h^\bullet(c)[h_s(f_s(r))] = h(c[f_s(r)])$$

rezultă că

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

ceea ce încheie demonstraţia, deoarece arată că $c[f_s(l)] \overset{\circ}{=}_s c[f_s(r)] \in D$.

Deoarece D este închisă la regulile $\mathbf{Cl}_{Ker(h)}$, **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 2.8.7 *Dacă* $a \overset{*}{\Rightarrow}_\Gamma d$ *atunci* $h(a) \overset{*}{\Rightarrow}{}^{\mathcal{B}}_{\Gamma} h(d)$.

Demonstraţie: Putem aplica propoziţia folosind $\overset{*}{\Rightarrow}_\Gamma \subseteq \overset{h}{\Rightarrow}$.

Completitudinea

Propoziţie 2.8.8 *Fie* $h:\ \mathcal{A} \longrightarrow \mathcal{B}$ *un morfism surjectiv pe componente. Dacă* $h(a) \overset{*}{\Rightarrow}{}^{\mathcal{B}}_{\Gamma} h(d)$ *atunci* $a \overset{h}{\Rightarrow} d$.

Demonstraţie: Fie mulţimea

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

Observăm că $h_s(u) \overset{\circ}{=} h_s(v) \in D$ 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 **R**,**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 $h_s(a) \overset{\circ}{=}_s h_s(a) \in D$, prin urmare $d \overset{\circ}{=}_s d \in D$. Deci D este reflexivă.

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

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

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

$$c[f_s(l)] \stackrel{\circ}{=} c[f_s(r)] \in D.$$

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

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

Observăm că pentru orice $u \stackrel{\circ}{=}_t v \in H$, din

$$h_t(g_t(u)) \stackrel{\circ}{=} h_t(a_{uv}) \in D \text{ şi } h_t(g_t(u)) \stackrel{\circ}{=} h_t(a_{uv}) \in D$$

deducem

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

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

Prin urmare

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

Dar,

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

şi

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

Deci $c[f_s(l)] \stackrel{\circ}{=} c[f_s(r)] \in D$. $\square$

Corolar 2.8.9 *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 aceaşi teorie privind acest alt tip de rescriere. În concluzie rescrierile într-o algebra unifică rescrieriele într-o algebră liberă cu rescrierile modulo ecuaţii.

2.9 DEMONSTRAREA ECUAŢIILOR CONDIŢIONATE

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 subcapitol vom da o cale prin care se poate demonstra că o ecuaţie condiţionată γ este adevărată în orice Γ-algebră, adică $\Gamma \models_\Sigma \gamma$.

2.9.1 Preliminarii

Se arată uşor că

$$\mathcal{B} \models_\Sigma (\forall X)l \overset{\circ}{=}_s r \textbf{ if } H \text{ şi } \mathcal{B} \models_\Sigma \{(\forall X)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=}_t v \in H\} \text{ implică } \mathcal{B} \models_\Sigma (\forall X)l \overset{\circ}{=}_s r.$$

Întradevăr pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{B}$ din $\mathcal{B} \models_\Sigma \{(\forall X)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=}_t v \in H\}$ deducem $h_t(u) = h_t(v)$ pentru orice $u \overset{\circ}{=}_t v \in H$. Prin urmare din $\mathcal{B} \models_\Sigma (\forall X)l \overset{\circ}{=}_s r$ **if** H rezultă că $h_s(l) = h_s(r)$. Am probat că pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{B}$ are loc egalitatea $h_s(l) = h_s(r)$, deci $\mathcal{B} \models_\Sigma (\forall X)l \overset{\circ}{=}_s r$.

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

Lemă 2.9.1

$$\mathcal{B} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r \textbf{ if } H \textit{ dacă şi numai dacă } [\mathcal{B} \models_\Sigma \{(\forall\emptyset)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=}_t v \in H\} \textit{ implică } \mathcal{B} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r].$$

Demonstraţie: Presupunem că $\mathcal{B} \models_\Sigma \{(\forall\emptyset)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=}_t v \in H\}$ implică $\mathcal{B} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r$.
Probăm că $\mathcal{B} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r$ **if** H.

Fie $h :\ T_\Sigma \longrightarrow \mathcal{B}$ cu $h_t(u) = h_t(v)$ pentru orice $u \overset{\circ}{=}_t v \in H$.

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

Din ipoteză deducem $\mathcal{B} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r$, deci $h_s(l) = h_s(r)$. $\square$

Corolar 2.9.2

$$\Gamma \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r \textbf{ if } H \textit{ dacă şi numai dacă } \Gamma \cup \{(\forall\emptyset)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=} v \in H\} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r.$$

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

$$\Gamma \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r \textbf{ if } H$$

care prin definiţie este echivalentă cu

$$(\forall\mathcal{B} \models_\Sigma \Gamma)\mathcal{B} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r \textbf{ if } H.$$

Conform lemei afirmaţia de mai sus este echivalentă cu

$$(\forall\mathcal{B} \models_\Sigma \Gamma)(\mathcal{B} \models_\Sigma \{(\forall\emptyset)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=} v \in H\} \text{ implică } \mathcal{B} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r)$$

care este echivalentă cu

$$(\forall\mathcal{B} \models_\Sigma (\Gamma \cup \{(\forall\emptyset)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=} v \in H\}))\ \mathcal{B} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r$$

adică cu

$$\Gamma \cup \{(\forall\emptyset)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=} v \in H\} \models_\Sigma (\forall\emptyset)l \overset{\circ}{=}_s r. \square$$

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.

2.9.2 Schimbarea signaturii

Prima etapă constă în înlocuirea ecuaţiei condiţionate 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). Această tehnică de demonstraţie a mai fost folosită în subcapitolul 1.4.5.

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. Gândiţi-vă la el ca la o funcţie incluziune de mulţimi deoarece conceptul de morfism de signaturi va fi introdus mult mai târziu. Vom nota functorul uituc cu

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

Deoarece conceptul categorial de functor nu este definit aici puneţi mai mult accent pe cuvântul uituc. Un alt exemplu de functor uitic, de la inele la grupuri, este cel care uită înmulţirea inelului păstrând numai structura aditivă a grupului. $Mod(j)$ uită constantele corespunzătoare lui X din fiecare Σ_X–algebră păstrând numai structura de Σ–algebră.

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 $a; h = b$. Explicit egalitatea $a; h = b$ spune că

$$h_s(a_s(x)) = b_s(x)$$

pentru orice $s \in S$ şi $x \in X_s$.

Având in vedere notaţiile de mai sus observăm că

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

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

Un cititor mai putin experimentat poate considera cele două egalităţi de mai sus drept definiţii. Prima egalitate reflectă uitarea constantelor corespunzătoare variabilelor din X. A doua egalitate spune că orice morfism de Σ_X–algebre este morfism de Σ–algebre

Schimbarea signaturii Σ cu $\Sigma_X$impune translatarea Σ–axiomelor în Σ_X–axiome.

2.9.3 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 2.9.3 *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 $T_{\Sigma}(Y)$.

Probăm surjectivitatea lui F. Fie Σ-morfismul $u : T_{\Sigma}(Y) \longrightarrow \mathcal{M}$. Definim $g : T_{\Sigma}(X \cup Y) \longrightarrow \mathcal{M}$ unicul morfism cu proprietăţile

$$g(z) = \begin{cases} u(z) & \text{dacă } z \in Y, \\ m(z) & \text{dacă } z \in X. \end{cases}$$

Deoarece $g(x) = m(x)$ pentru orice $x \in X$, observăm că $g : T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m)$ este Σ_X-morfism. Deoarece $g(y) = u(y)$ pentru orice $y \in Y$ deducem că restricţia lui g la $T_{\Sigma}(Y)$ coincide cu u.

Probăm injectivitatea lui F. Fie $h_1 : T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m)$ şi $h_2 : T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m)$ două Σ_X–morfisme cu proprietatea $F(h_1) = F(h_2)$. Deoarece h_1 şi h_2 sunt Σ_X–morfisme deducem $h_1(x) = m(x) = h_2(x)$ pentru orice $x \in X$. Din $F(h_1) = F(h_2)$ deducem $h_1(y) = h_2(y)$ pentru orice $y \in Y$. Deoarece h_1 şi h_2 sunt morfisme definite pe $T_{\Sigma}(X \cup Y)$ deducem că $h_1 = h_2$. $\square$

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

$$\gamma = (\forall Y) l \stackrel{\circ}{=}_s r \textbf{ if } H$$

îi ataşăm o altă Σ_X-ecuaţie condiţionată

$$J(\gamma) = (\forall Y) i_s(l) \stackrel{\circ}{=}_s i_s(r) \textbf{ if } \{i_t(u) \stackrel{\circ}{=} i_t(v) \mid u \stackrel{\circ}{=}_t v \in H\}.$$

Funcţia J duce o Σ-ecuaţie necondiţionată într-o Σ_X-ecuaţie necondiţionată.

Lemă 2.9.4 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

$$\forall h :\ T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M},\ m) \text{ dacă } h_t(i_t(u)) = h_t(i_t(v)) \text{ pentru orice } u \stackrel{\circ}{=}_t v, \text{ atunci } h_s(i_s(l)) = h_s(i_s(r)),$$

prin urmare echivalentă cu

$$\forall h :\ T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M},\ m) \text{ dacă } (i; Mod(j)(h))_t(u) = (i; Mod(j)(h))_t(v) \text{ pentru orice } u \stackrel{\circ}{=}_t v \in H, \text{ atunci } (i; Mod(j)(h))_s(l) = (i; Mod(j)(h))_s(r);$$

adică echivalentă cu

$$\forall h :\ T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M},\ m) \text{ dacă } F(h)_t(u) = F(h)_t(v) \text{ pentru orice } u \stackrel{\circ}{=}_t v \in H, \text{ atunci } F(h)_s(l) = F(h)_s(r).$$

Folosind bijecţia de mai sus deducem că $(\mathcal{M}, m) \models_{\Sigma_X} J(\gamma)$ este echivalentă cu

$$\forall g :\ T_{\Sigma}(Y) \longrightarrow \mathcal{M} \text{ dacă } g_t(u) = g_t(v) \text{ pentru orice } u \stackrel{\circ}{=}_t v \in H, \text{ atunci } g_s(l) = g_s(r)$$

adică cu $\mathcal{M} \models_{\Sigma} \gamma$. $\square$

Pentru o mulţime Γ de Σ-ecuaţii condiţionate notăm $J(\Gamma) = \{J(\gamma) \mid \gamma \in \Gamma\}$.

Corolar 2.9.5 *Pentru orice Σ_X-algebră $(\mathcal{M}, m)$ şi orice mulţime Γ de Σ-ecuaţii condiţionate*

$$(\mathcal{M}, m) \models_{\Sigma_X} J(\Gamma) \textit{ dacă şi numai dacă } \mathcal{M} \models_{\Sigma} \Gamma. \ \square$$

2.9.4 Teorema constantelor

Teorema 2.9.6 Teorema constantelor:

$$\Gamma \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r \textbf{ if } H \ \ \textit{dacă şi numai dacă} \ \ J(\Gamma) \models_{\Sigma_X} (\forall \emptyset) l \stackrel{\circ}{=}_s r \textbf{ if } H.$$

Demonstraţie: La început vom menţiona forme echivalente pentru cei doi membri ai echivalenţei din enunţ.

Membrul stâng

$$\Gamma \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r \textbf{ if } H$$

este echivalent succesiv cu:

1. Oricare ar fi Σ-algebra $\mathcal{A}$, dacă $\mathcal{A} \models_\Sigma \Gamma$, atunci $\mathcal{A} \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r$ **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 $u \stackrel{\circ}{=}_t v \in H$, atunci $h_s(l) = h_s(r)$.

Membrul drept

$$J(\Gamma) \models_{\Sigma_X} (\forall \emptyset) l \stackrel{\circ}{=}_s r \textbf{ if } H$$

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 $h : \ T_{\Sigma_X} \longrightarrow (\mathcal{M},\ m) \models_{\Sigma_X} J(\Gamma)$
   dacă $h_t(u) = h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$, atunci $h_s(l) = h_s(r)$.

Presupunem că $\Gamma \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r$ **if** H. Fie $h : \ T_{\Sigma_X} \longrightarrow (\mathcal{M},\ m) \models_{\Sigma_X} J(\Gamma)$ cu $h_t(u) = h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$.

Deoarece $\mathcal{M} \models_\Sigma \Gamma$ rezultă că $\mathcal{M} \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r$ **if** H prin urmare din $h : \ T_\Sigma(X) \longrightarrow \mathcal{M}$ şi $h_t(u) = h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$ deducem $h_s(l) = h_s(r)$.

Reciproc, presupunem că $J(\Gamma) \models_{\Sigma_X} (\forall \emptyset) l \stackrel{\circ}{=}_s r$ **if** H. Fie $h : \ T_\Sigma(X) \longrightarrow \mathcal{M} \models_\Sigma \Gamma$ cu $h_t(u) = h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$.

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 orice $u \stackrel{\circ}{=}_t v \in H$ din ipoteză deducem $h_s(l) = h_s(r)$. $\Box$

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 2.9.7 Teorema deducţiei:

$$\Gamma \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r \textbf{ if } H \ \ \textit{dacă şi numai dacă} \ \ J(\Gamma) \cup \{(\forall \emptyset) u \stackrel{\circ}{=}_t v \mid u \stackrel{\circ}{=}_t v \in H\} \models_{\Sigma_X} (\forall \emptyset) l \stackrel{\circ}{=}_s r.$$

Demonstraţie: Conform teoremei constantelor

$$\Gamma \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r \textbf{ if } H$$

este echivalentă cu

$$J(\Gamma) \models_{\Sigma_X} (\forall \emptyset) l \stackrel{\circ}{=}_s r \textbf{ if } H$$

care conform corolarului 2.9.2 este echivalentă cu

$$J(\Gamma) \cup \{(\forall \emptyset) u \stackrel{\circ}{=}_t v \in H\} \models_{\Sigma_X} (\forall \emptyset) l \stackrel{\circ}{=}_s r. \ \Box$$

In concluzie pentru demonstrarea Σ-ecuaţiei condiţionate $(\forall X) l \stackrel{\circ}{=}_s r$ **if** H putem proceda astfel

- schimbăm signatura Σ cu signatura Σ_X, translatând axiomele din Γ în noua signatură
- adăugăm $\{(\forall \emptyset) u \stackrel{\circ}{=}_t v \mid u \stackrel{\circ}{=}_t v \in H\}$ la axiome şi
- demonstrăm $(\forall \emptyset) l \stackrel{\circ}{=}_s r$ în noul context.

Credem că este bine să semnalăm întrebarea. Dacă Γ este program, este şi $J(\Gamma) \cup \{(\forall \emptyset) u \stackrel{\circ}{=}_t v \in H\}$ un program?

2.10 PERECHI CRITITCE

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 mai poate fi rescrisă, adică este o formă normală.

Lemă 2.10.1 *Pentru orice $c \in T_\Sigma(A \cup \{\bullet\})$ dacă $a \stackrel{*}{\Rightarrow}_\Gamma d$, atunci $c[a] \stackrel{*}{\Rightarrow}_\Gamma c[d]$.*

Demonstraţie: Menţionăm că lema arată mai mult decât închiderea la contexte, căci în enunţ nu se presupune nimic despre c.

Probăm prin inducţie structurală după c.

Dacă $c = \bullet$ 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 **C**Σ deducem

$$A_\sigma(c_1[a], c_2[a], \ldots, c_n[a]) \stackrel{*}{\Rightarrow}_\Gamma A_\sigma(c_1[d], c_2[d], \ldots, c_n[d]),$$

prin urmare deoarece

$A_\sigma(c_1[a], c_2[a], \ldots, c_n[a]) = A_\sigma(\bullet \leftarrow a(c_1), \bullet \leftarrow a(c_2), \ldots, \bullet \leftarrow a(c_n)) =$
$= \bullet \leftarrow a(\sigma(c_1, c_2, \ldots, c_n)) = \bullet \leftarrow a(c) = c[a]$ şi
$A_\sigma(c_1[d], c_2[d], \ldots, c_n[d]) = A_\sigma(\bullet \leftarrow d(c_1), \bullet \leftarrow d(c_2), \ldots, \bullet \leftarrow d(c_n)) =$
$= \bullet \leftarrow d(\sigma(c_1, c_2, \ldots, c_n)) = \bullet \leftarrow d(c) = c[d]$ tragem concluzia
$c[a] \stackrel{*}{\Rightarrow}_\Gamma c[d]$. $\Box$

Lemă 2.10.2 *Pentru orice $c \in T_\Sigma(A \cup \{\bullet\})$ 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. $\Box$

Lemă 2.10.3 *Dacă y nu apare în $c \in T_\Sigma(A \cup \{\bullet\})$, atunci $(y \longrightarrow b)(c[a]) = c[(y \longrightarrow b)(a)]$.*

Demonstraţie: Se face prin inducţie structurală după c. $\Box$

Definiţia 2.10.4 Fie $(\forall X_1) l_1 \stackrel{\circ}{=}_{s'} r_1$ şi $(\forall X_2) l_2 \stackrel{\circ}{=}_{s''} r_2$ două axiome din Γ fără variabile comune, ipoteză care nu restrânge generalitatea. Fie c în $T_\Sigma(A \cup \{\bullet\}_{s'})$ un context astfel încât

$$l_1 = c[d]$$

unde d <u>nu este o variabilă</u>. 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**. $\Box$

Pe scurt notăm mulţimea perechilor critice

$$\mathbf{PC}_\Gamma = \{(u(r_1), u(c[r_2]))\ :\ (\forall X_1)l_1 \overset{\circ}{=}_{s'} r_1, (\forall X_2)l_2 \overset{\circ}{=}_{s''} r_2 \in \Gamma,\ l_1 = c[d],\ c \text{ este context, } d \text{ nu este variabilă şi } u = \mathbf{CGU}\{d, l_2\}\ \}$$

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 2.10.5 *Dacă rescrierea este local confluentă, atunci* $\mathbf{PC}_\Gamma \subseteq \downarrow_\Gamma$.

Demonstraţie: Observăm că

$$u(l_1) \Rightarrow_\Gamma u(r_1) \text{ şi}$$

$$u(l_1) = u^\bullet(c)[u(d)] = u^\bullet(c)[u(l_2)] \Rightarrow_\Gamma u^\bullet(c)[u(r_2)] = u(c[r_2]).$$

Deoarece rescrierea este local confluentă deducem că

$$u(r_1) \downarrow_\Gamma u(c[r_2]).\ \square$$

Teorema 2.10.6 *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 $(\forall X_i)l_i \overset{\circ}{=} r_i \in \Gamma$, 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 $h_1(l_1)$.

Cazul 1(fără suprapunere). Intuitiv cele două rescrieri se fac în zone disjuncte din a, adică $h_1(l_1)$ şi $h_2(l_2)$ sunt subarbori disjuncţi ai lui a. Rescrierea $a \Rightarrow a_1$ se face înlocuind $h_1(l_1)$ cu $h_1(r_1)$, subarborele $h_2(l_2)$ rămânând neschimbat. Rescrierea $a \Rightarrow a_2$ se face înlocuind $h_2(l_2)$ cu $h_2(r_2)$, subarborele $h_1(l_1)$ rămânând neschimbat. Observăm că a_1 şi a_2 pot fi rescrişi în acelaşi element care se obţine fie înlocuind în a_1 pe $h_2(l_2)$ cu $h_2(r_2)$, fie înlocuind în a_2 pe $h_1(l_1)$ cu $h_1(r_1)$.

Formal înlocuind în a subarborii $h_1(l_1)$ şi $h_2(l_2)$ cu două variabile noi z_1 şi z_2 există $c \in T_\Sigma(A \cup \{z_1, z_2\})$ cu câte o apariţie a variabilelor 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

$$a_1 = c_1[h_1(r_1)] = (z_1 \leftarrow h_1(r_1))(c_1) = (z_1 \leftarrow h_1(r_1))[(z_2 \leftarrow h_2(l_2))(c)] = (z_2 \leftarrow h_2(l_2))[(z_1 \leftarrow h_1(r_1))(c)] = \{(z_1 \leftarrow h_1(r_1))(c)\}[h_2(l_2)]$$

prin urmare

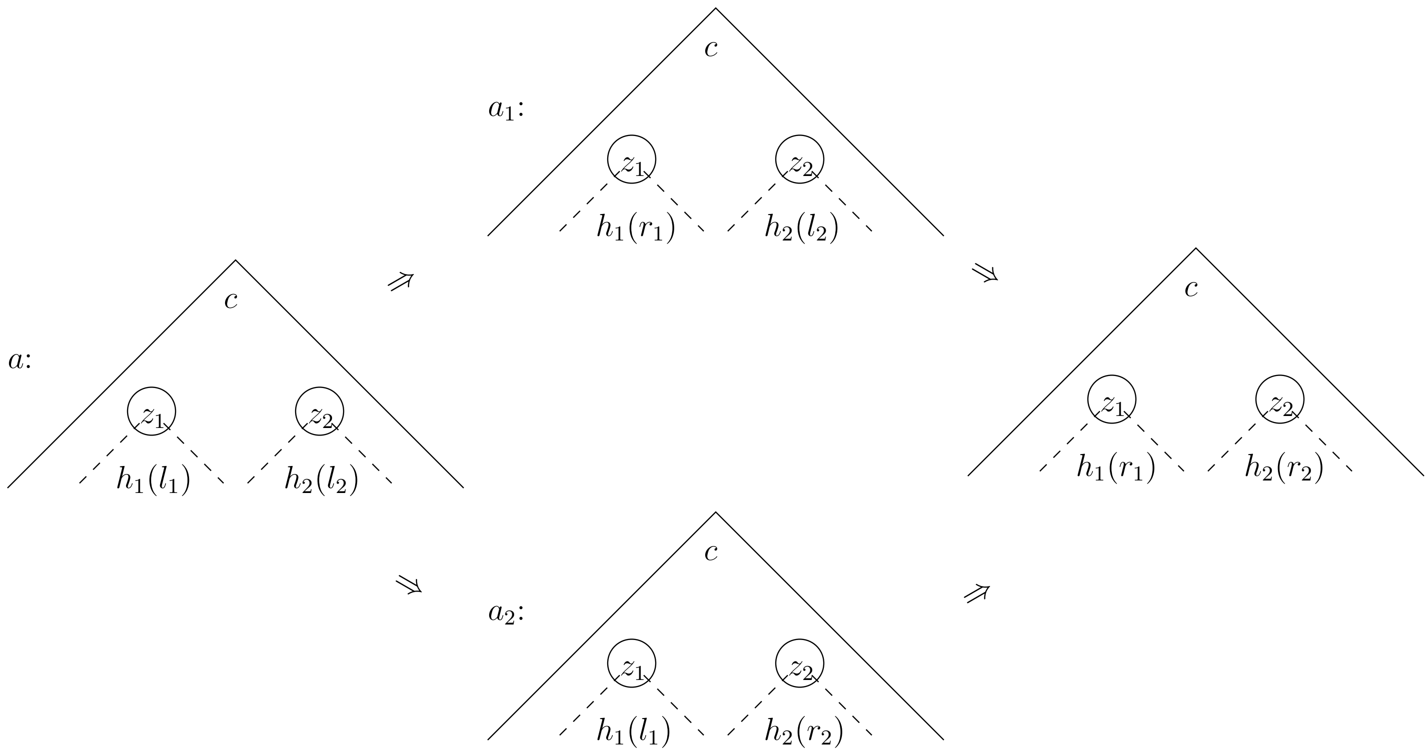


Figure 2.2: Cazul 1. Subarbori disjuncţi

$$a_1 \Rightarrow (z_1 \leftarrow h_1(r_1))(c)[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))[(z_1 \leftarrow h_1(r_1))(c)].$$

Observăm că

$$a_2 = c_2[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))(c_2) = (z_2 \leftarrow h_2(r_2))([z_1 \leftarrow h_1(l_1)](c)) = \{(z_2 \leftarrow h_2(r_2))(c)\}[h_1(l_1)]$$

prin urmare

$$a_2 \Rightarrow \{(z_2 \leftarrow h_2(r_2))(c)[h_1(r_1)] = (z_1 \leftarrow h_1(r_1))[(z_2 \leftarrow h_2(r_2))(c)] = [z_2 \leftarrow h_2(r_2)][(z_1 \leftarrow h_1(r_1))(c)],$$

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).

Deoarece $h_2(l_2)$ este o subexpresie a lui $h_1(l_1)$ rezultă existenţa unei subexpresii d a lui l_1 astfel încât

$$h_2(l_2) = h_1(d).$$

Notăm cu c contextul cu variabila distinsă z_2 în care d apare în l_1, adică

$$l_1 = c[d].$$

Deoarece

$$a = c_1[h_1(l_1)] = c_1[h_1(c[d])] = c_1[h_1^{z_2}(c)[h_1(d)]] = c_1[h_1^{z_2}(c)[h_2(l_2)]] = c_1[(z_2 \leftarrow h_2(l_2))(h_1^{z_2}(c))].$$

Deoarece z_2 nu apare în c_1

$$a = (z_2 \leftarrow h_2(l_2))(c_1[h_1^{z_2}(c)]) = c_1[h_1^{z_2}(c)][h_2(l_2)]$$

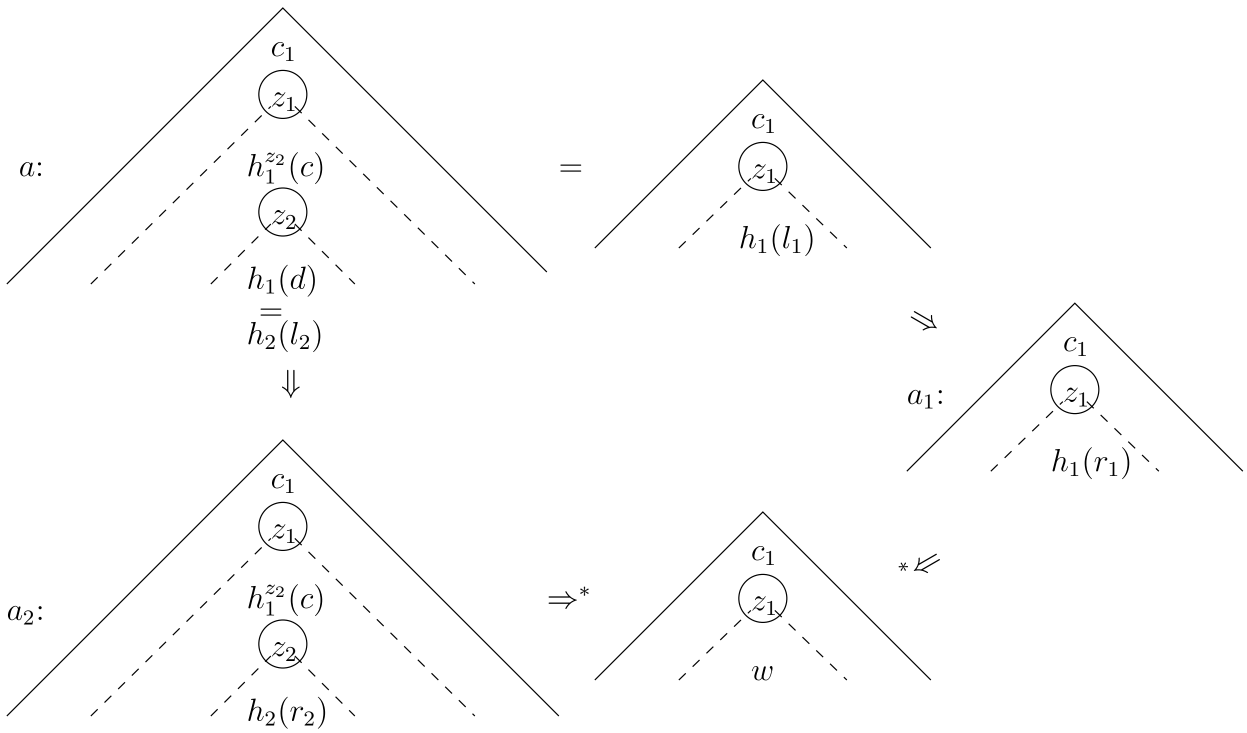


Figure 2.3: Cazul 2. Cu suprapunere critică

prin urmare

$$c_2 = c_1[h_1^{z_2}(c)],$$

deci

$$a = c_1[h_1^{z_2}(c)][h_2(l_2)]$$

şi

$$a_2 = c_1[h_1^{z_2}(c)][h_2(r_2)] = (z_2 \leftarrow h_2(r_2))(c_1[h_1^{z_2}(c)]).$$

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

$$\begin{aligned} a_2 &= (z_2 \leftarrow h_2(r_2))((z_1 \leftarrow h_1^{z_2}(c))(c_1)) \text{ deoarece } z_2 \text{ nu apare } c_1 \\ &= (z_1 \leftarrow \{z_2 \leftarrow h_2(r_2)(h_1^{z_2}(c))\})(c_1) = (z_1 \leftarrow \{h_1^{z_2}(c)[h_2(r_2)]\})(c_1) = (z_1 \leftarrow h(c[r_2]))(c_1) = \\ &= c_1[h(c[r_2])] \stackrel{*}{\Rightarrow}_\Gamma c_1[w], \end{aligned}$$

deci $a_1 \downarrow_\Gamma a_2$.

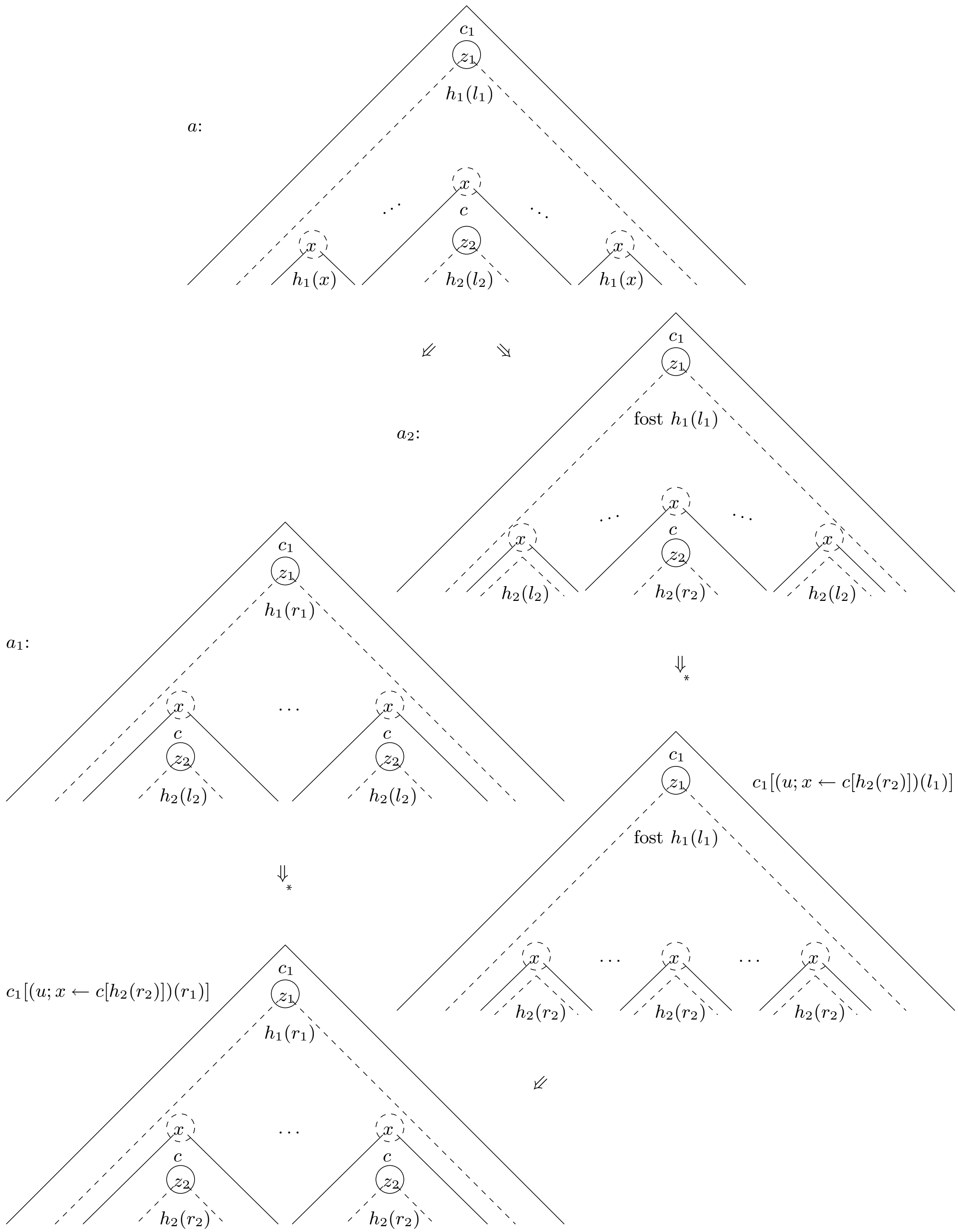


Figure 2.4: Cazul 3. Cu suprapunere necritică

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 $c \in T_\Sigma(A \cup \{z_2\})$ contextul în care se găseşte $h_2(l_2)$ în $h_1(x)$, prin urmare

$$h_1(x) = c[h_2(l_2)] \tag{2.1}$$

Fie

$$u : T_\Sigma(X_1) \longrightarrow T_\Sigma(A \cup \{x\})$$

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 $x \leftarrow h_1(x) = x \leftarrow c[h_2(l_2)] = x \leftarrow c\,; z_2 \leftarrow h_2(l_2)$, 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

$$a_1 \overset{*}{\Rightarrow}_\Gamma c_1[(u; x \leftarrow c)(r_1)])[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))(c_1[(u; x \leftarrow c)(r_1)]) = c_1[(z_2 \leftarrow h_2(r_2))((u; x \leftarrow c)(r_1))]$$

deci

$$a_1 \overset{*}{\Rightarrow}_\Gamma c_1[(u; x \leftarrow c; z_2 \leftarrow h_2(r_2))(r_1)] = c_1[(u; x \leftarrow c[h_2(r_2)])(r_1)].$$

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

$$w; z_2 \leftarrow h_2(r_2) \;=\; p; x \leftarrow h_1(x).$$

$$a_2 = c_2[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))(c_1[w(l)]) = c_1[(w; z_2 \leftarrow h_2(r_2))(l)] = c_1[(p; x \leftarrow h_1(x))(l)].$$

Deoarece $x \leftarrow h_1(x) \;=\; x \leftarrow c; z_2 \leftarrow h_2(l_2)$

$a_2 = c_1[(z_2 \leftarrow h_2(l_2))((p; x \leftarrow c)(l))] = (z_2 \leftarrow h_2(l_2))(c_1[(p; x \leftarrow c)(l)]) = (c_1[(p; x \leftarrow c)(l)])[h_2(l_2)]$.
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

$$a_1 \downarrow_\Gamma a_2. \ \Box$$

2.11 TERMINAREA şi PROCEDURA KNUTH-BENDIX

2.11.1 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. Menţionăm că terminarea este denumirea folosită în informatică pentru buna fondare.

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 $n \in N$.

Teorema 2.11.1 *Următoarele proprietăţi sunt echivalente*

1. *Rescrierea are proprietatea de terminare,*
2. *închiderea tranzitivă a rescrierii întrun pas* $\stackrel{+}{\Rightarrow}_\Gamma$ *este o relaţie de ordine strictă şi bine fondată,*
3. *există o relaţie de ordine* $\geq$ *pe* A *compatibilă cu structura algebrică cu proprietăţile:*
   (a) *varianta ei strictă* $>$ *este bine fondată,*
   (b) *pentru orice* $(\forall X) l =_s r$ **if** $H \in \Gamma$ *şi orice* $h : T_\Sigma(X) \to \mathcal{A}$ *dacă* $h(H) \subseteq \downarrow_\Gamma$*, atunci* $h(l) \geq h(r)$
4. *există o relaţie bine fondată* $\vdash$ *pe o mulţime* M *şi o funcţie* $\rho : \ A \longrightarrow M$ *cu proprietăţile*
   (a) *pentru orice* $(\forall X) l =_s r$ **if** $H \in \Gamma$ *şi orice* $h : T_\Sigma(X) \to \mathcal{A}$ *dacă* $h(H) \subseteq \downarrow_\Gamma$ *şi* $h(l) \neq h(r)$*, atunci* $\rho(h(l)) \vdash \rho(h(r))$
   (b) *pentru orice* $n \geq 1$ *şi orice* $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$*,* $\rho(a) \vdash \rho(d)$ *şi* $A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) \neq A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n)$ *implică* $\rho(A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n)) \vdash \rho(A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n))$*,*
5. *există o relaţie bine fondată* $\vdash$ *pe o mulţime* M *şi* $\rho : \ A \longrightarrow M$ *o funcţie cu proprietăţile*
   (a) *pentru orice* $(\forall X) l =_s r$ **if** $H \in \Gamma$ *şi orice* $h : T_\Sigma(X) \to \mathcal{A}$ *dacă* $h(H) \subseteq \downarrow_\Gamma$ *şi* $h(l) \neq h(r)$*, atunci* $\rho(h(l)) \vdash \rho(h(r))$*,*
   (b) *pentru orice context* $c \in \mathcal{A}[z]_{s'}$ *şi pentru orice* $a, d \in A_s$ *dacă* $\rho(a) \vdash \rho(d)$ *şi* $c[a] \neq c[d]$*, atunci* $\rho(c[a]) \vdash \rho(c[d])$*,*
6. *există o relaţie bine fondată* $\vdash$ *pe mulţimea* M *şi* $\rho : \ A \longrightarrow M$ *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. Probăm (b) prin inducţie structurală după contextul c. Observăm că elementele algebrei $\mathcal{A}$ nu sunt contexte.

Dacă $c = \bullet$, deoarece $c[a] = a$ şi $c[d] = d$ concluzia rezultă direct din ipoteză.

Dacă $c = \sigma(a_1, a_2, \ldots, a_n)$ 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 sunt din A. Presupunem $c[a] \neq c[d]$. Deoarece $c[a] = A_\sigma(a_1, \ldots, a_{i-1}, a_i[a], a_{i+1}, \ldots, a_n)$ şi $c[d] = A_\sigma(a_1, \ldots, a_{i-1}, a_i[d], a_{i+1}, \ldots, a_n)$ rezultă că $a_i[a] \neq a_i[d]$. Din ipoteza de inducţie deducem că $\rho(a_i[a]) \vdash \rho(a_i[d])$.

Deoarece $A_\sigma(a_1, \ldots, a_{i-1}, a_i[a], a_{i+1}, \ldots, a_n) \neq A_\sigma(a_1, \ldots, a_{i-1}, a_i[d], a_{i+1}, \ldots, a_n)$, din ipoteză rezultă că

$$\rho(A_\sigma(a_1, \ldots, a_{i-1}, a_i[a], a_{i+1}, \ldots, a_n)) \vdash \rho(A_\sigma(a_1, \ldots, a_{i-1}, a_i[d], a_{i+1}, \ldots, a_n)).$$

5⟶6. Presupunem că $a \Rightarrow_\Gamma d$. Prin urmare există $(\forall X) l =_s r$ **if** $H \in \Gamma$, morfismul $h : T_\Sigma(X) \to \mathcal{A}$ şi contextul $c \in T_\Sigma(A \cup \{\bullet\})_{s'}$ cu proprietăţile $h(H) \subseteq \downarrow_\Gamma$, $a = c[h_s(l)]$ şi $d = c[h_s(r)]$.

În plus $a \neq d$ implică $h_s(l) \neq h_s(r)$.

Din prima ipoteză deducem $\rho(h_s(l)) \vdash \rho(h_s(r))$.

Folosim $c[h_s(l)] \neq c[h_s(r)]$ şi aplicând a doua ipoteză rezultă că $\rho(c[h_s(l)]) \vdash \rho(c[h_s(r)])$. Prin urmare $\rho(a) \vdash \rho(d)$.

6⟶1. Presupunrm prin absurd că $\stackrel{*}{\Rightarrow}_\Gamma$ 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 bine fondarea relaţiei $\vdash$. $\square$

Se poate observa diferenţa esenţială dintre cele două ipoteze ale variantelor 4 sau 5 din teorema 2.11.1. 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.

2.11.2 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 bine fondată 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:

$$\rho(e) = 2, \ \ \rho(x) = 2 \text{ pentru orice variabilă } x$$

$$\rho(t') = 2^{\rho(t)} \text{ şi } \rho(t * s) = \rho(t)^2 \rho(s).$$

Observăm că $\rho(t) \geq 2$ pentru orice termen t.

Verificăm prima ipoteză a echivalenţelor 4 sau 5 din teorema 2.11.1.

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.

2.11.3 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 2.11.1. În aceste condiţii atenţia se va concentra asupra veridicităţii condiţiei (a) din aceleaşi condiţii echivalente.

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.

$$e * A = A \tag{2.2}$$

$$A' * A = e \tag{2.3}$$

$$(A * B) * C = A * (B * C) \tag{2.4}$$

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

$$u(l_1) \Rightarrow u(r_1) \quad \text{şi} \quad u(l_1) = u^\bullet(c)[u(l_2)] \Rightarrow u^\bullet(c)[u(r_2)] = u(c[r_2])$$

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 2.11.2, pentru a alege între cele două egalităţi şi anume dacă $\rho(n_1) > \rho(n_2)$ alegem $n_1 \overset{\circ}{=} n_2$ pentru a o adăuga celorlalte reguli.

Unificând membrul stâng al lui (2.3) cu subtermenul $A * B$ din membrul stâng al lui (2.4) şi calculând formele normale obţinem

$$(A' * A) * C \overset{2.3}{\Longrightarrow} e * C \overset{2.2}{\Longrightarrow} C$$

$$(A' * A) * C \overset{2.4}{\Longrightarrow} A' * (A * C)$$

prin urmare adăugăm regula

$$A' * (A * C) = C \tag{2.5}$$

Unificând membrul stâng al lui (2.2) cu subtermenul $A * C$ din membrul stâng al lui (2.5) şi calculând formele normale obţinem

$$e' * (e * A) \overset{2.5}{\Longrightarrow} A$$

$$e' * (e * A) \overset{2.2}{\Longrightarrow} e' * A$$

prin urmare adăugăm regula

$$e' * A = A \tag{2.6}$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (2.3) cu subtermenul $A * C$ din membrul stâng al lui (2.5) şi calculând formele normale obţinem

$$B'' * (B' * B) \overset{2.3}{\Longrightarrow} B'' * e$$

$$B'' * (B' * B) \overset{2.5}{\Longrightarrow} B$$

prin urmare adăugăm regula

$$B'' * e = B \tag{2.7}$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (2.7) cu subtermenul $A * B$ din membrul stâng al lui (2.4) şi calculând formele normale obţinem

$$(B'' * e) * A \overset{2.4}{\Longrightarrow} B'' * (e * A) \overset{2.2}{\Longrightarrow} B'' * A$$

$$(B'' * e) * A \overset{2.7}{\Longrightarrow} B * A$$

prin urmare adăugăm regula

$$B'' * A = B * A \tag{2.8}$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (2.7) cu membrul stâng al lui (2.8) şi calculând formele normale obţinem

$$B'' * e \overset{2.7}{\Longrightarrow} B$$

$$B'' * e \stackrel{2.8}{\Longrightarrow} B * e$$

prin urmare adăugăm regula

$$B * e = B \tag{2.9}$$

Unificând membrul stâng al lui (2.7) cu membrul stâng al lui (2.9) şi calculând formele normale obţinem

$$B'' * e \stackrel{2.7}{\Longrightarrow} B$$

$$B'' * e \stackrel{2.9}{\Longrightarrow} B''$$

prin urmare adăugăm regula

$$B'' = B \tag{2.10}$$

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 $l_1 \stackrel{\circ}{=} l_2$ că folosind celelalte reguli (fără $l_1 \stackrel{\circ}{=} l_2$) l_1 şi l_2 pot fi rescrise în aceeaşi formă normală, atunci vom elimina regula $l_1 \stackrel{\circ}{=} l_2$.

În acest moment regulile (2.7) şi (2.8) pot fi eliminate deoarece

$$B'' * e \stackrel{2.10}{\Longrightarrow} B * e \stackrel{2.9}{\Longrightarrow} B \quad \text{şi}$$

$$B'' * A \stackrel{2.10}{\Longrightarrow} B * A.$$

Lucrăm în continuare cu regulile (2.2, 2.3, 2.4, 2.5, 2.6, 2.9 şi 2.10).

Unificând membrul stâng al lui (2.6) cu membrul stâng al lui (2.9) şi calculând formele normale obţinem

$$e' * e \stackrel{2.6}{\Longrightarrow} e$$

$$e' * e \stackrel{2.9}{\Longrightarrow} e'$$

prin urmare adăugăm regula

$$e' = e \tag{2.11}$$

În acest moment regula (2.6) poate fi eliminată deoarece

$$e' * A \stackrel{2.11}{\Longrightarrow} e * A \stackrel{2.2}{\Longrightarrow} A$$

Lucrăm în continuare cu regulile (2.2, 2.3, 2.4, 2.5, 2.9, 2.10 şi 2.11).

Unificând membrul stâng al lui (2.10) cu subtermenul A' din membrul stâng al lui (2.3) şi calculând formele normale obţinem

$$B'' * B' \stackrel{2.3}{\Longrightarrow} e$$

$$B'' * B' \stackrel{2.10}{\Longrightarrow} B * B'$$

prin urmare adăugăm regula

$$B * B' = e \tag{2.12}$$

Unificând membrul stâng al lui (2.12) cu subtermenul $A * B$ din membrul stâng al lui (2.4) şi calculând formele normale obţinem

$$(A * A') * C \stackrel{2.4}{\Longrightarrow} A * (A' * C)$$

$$(A * A') * C \stackrel{2.12}{\Longrightarrow} e * C \stackrel{2.2}{\Longrightarrow} C$$

prin urmare adăugăm regula

$$A * (A' * C) = C \tag{2.13}$$

Unificând membrul stâng al lui (2.4) cu membrul stâng al lui (2.12) şi calculând formele normale obţinem

$$(A * B) * (A * B)' \stackrel{2.4}{\Longrightarrow} A * (B * (A * B)')$$

$$(A * B) * (A * B)' \stackrel{2.12}{\Longrightarrow} e$$

prin urmare adăugăm regula

$$A * (B * (A * B)') = e \tag{2.14}$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (2.14) cu subtermenul $A * C$ din membrul stâng al lui (2.5) şi calculând formele normale obţinem

$$A' * (A * (B * (A * B)')) \stackrel{2.14}{\Longrightarrow} A' * e \stackrel{2.9}{\Longrightarrow} A'$$

$$A' * (A * (B * (A * B)')) \stackrel{2.5}{\Longrightarrow} B * (A * B)'$$

prin urmare adăugăm regula

$$B * (A * B)' = A' \tag{2.15}$$

Mai târziu vom observa că această regulă este superfluă.

În acest moment regula (2.14) poate fi eliminată deoarece

$$A * (B * (A * B)') \stackrel{2.15}{\Longrightarrow} A * A' \stackrel{2.12}{\Longrightarrow} e$$

Lucrăm în continuare cu regulile (2.2, 2.3, 2.4, 2.5, 2.9, 2.10, 2.11, 2.12, 2.13 şi 2.15).

Unificând membrul stâng al lui (2.15) cu subtermenul $A * C$ din membrul stâng al lui (2.5) şi calculând formele normale obţinem

$$B' * (B * (A * B)') \stackrel{2.15}{\Longrightarrow} B' * A'$$

$$B' * (B * (A * B)') \stackrel{2.5}{\Longrightarrow} (A * B)'$$

prin urmare adăugăm regula

$$(A * B)' = B' * A' \tag{2.16}$$

În acest moment regula (2.15) poate fi eliminată deoarece

$$B * (A * B)' \stackrel{2.16}{\Longrightarrow} B * (B' * A') \stackrel{2.13}{\Longrightarrow} A'$$

Regulile rămase (2.2, 2.3, 2.4, 2.5, 2.9, 2.10, 2.11, 2.12, 2.13 şi 2.16) coincid cu programul prezentat în secţiunea 2.11.2.

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 2.11.2 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.

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 2.11.2 *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

$$u \equiv^{T_\Sigma(X)}_{\Gamma} v \text{ pentru orice } (\forall X)u \stackrel{\circ}{=} v \in \Gamma_1$$

Observaţia 2.11.3 $\Gamma \subseteq \equiv_{\Gamma}$

Demonstraţie: Fie $(\forall X)l \stackrel{\circ}{=}_s r \in \Gamma$. Fie $h : T_\Sigma(X) \longrightarrow \mathcal{A} \models \Gamma$. Deoarece $\mathcal{A} \models (\forall X)l \stackrel{\circ}{=}_s r$ deducem $h_s(l) = h_s(r)$. Deci $l \equiv^{T_\Sigma(X)}_{\Gamma} r$.

Propoziţie 2.11.4 *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 $(\forall X)u \stackrel{\circ}{=} v \in \Gamma_1$ 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 $h(u) = h(v)$. Deci $\mathcal{A} \models \Gamma_1$.

2) Presupunem în Σ-algebra $\mathcal{A}$ că $a \equiv^{\mathcal{A}}_{\Gamma_1} d$ şi probăm că $a \equiv^{\mathcal{A}}_{\Gamma} d$. 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 2.11.5 *Dacă* (w, v) *este o pereche critică pentru* Γ, *atunci* $w \equiv_{\Gamma} v$.

Demonstraţie: Presupunem $w = u(r_1)$ şi $v = u(c[r_2])$ unde $(l_1 = c[d], r_1)$ şi (l_2, r_2) sunt reguli cu mulţimi de variabile disjuncte din Γ şi $u = \mathbf{CGU}(d, l_2)$. Deoarece $u(l_1) \Rightarrow_{\Gamma} w$ şi $u(l_1) \Rightarrow_{\Gamma} v$ din corectitudinea relaţiei $\stackrel{*}{\Rightarrow}_{\Gamma}$ deducem $w \equiv_{\Gamma} v$. □

Propoziţie 2.11.6 *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 2.11.4 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 $l' \overset{\circ}{=} r'$ şi $\Gamma \cup \{l' \overset{\circ}{=} r'\}$. Din observaţia 2.11.5 deducem $P \subseteq \equiv_{\Gamma\cup\{l'\overset{\circ}{=}r'\}}$.

Din $l \overset{*}{\Rightarrow}_\Gamma l'$ şi $r \overset{*}{\Rightarrow}_\Gamma r'$ deducem $l \equiv_\Gamma l'$ şi $r \equiv_\Gamma r'$. Folosind $l \equiv_{\Gamma\cup(\forall X)l\overset{\circ}{=}r} r$ şi $l' \equiv_{\Gamma\cup(\forall X)l'\overset{\circ}{=}r'} r'$ deducem $l \equiv_{\Gamma\cup(\forall X)l'\overset{\circ}{=}r'} r$ şi $l' \equiv_{\Gamma\cup(\forall X)l\overset{\circ}{=}r} r'$, prin urmare

$$l \equiv_{E\cup P\cup\Gamma\cup(\forall X)l'\overset{\circ}{=}r'} r \text{ şi } l' \equiv_{E\cup\Gamma\cup(\forall X)l\overset{\circ}{=}r} r'.$$

Din $l' \equiv_{\Gamma\cup(\forall X)l\overset{\circ}{=}r} r'$ deducem $\Gamma \cup (\forall X)l' \overset{\circ}{=} r' \ \subseteq \equiv_{\Gamma\cup(\forall X)l\overset{\circ}{=}r}$, aşadar $\equiv_{\Gamma\cup(\forall X)l'\overset{\circ}{=}r'} \subseteq \equiv_{\Gamma\cup(\forall X)l\overset{\circ}{=}r}$, deci

$$P \subseteq \equiv_{E\cup\Gamma\cup(\forall X)l\overset{\circ}{=}r} .$$

Din cele de mai sus deducem

$$E \cup P \cup \Gamma \cup (\forall X)l' \overset{\circ}{=} r' \subseteq \equiv_{E\cup\Gamma\cup(\forall X)l\overset{\circ}{=}r} \text{ şi}$$

$$E \cup \Gamma \cup (\forall X)l \overset{\circ}{=} r \subseteq \equiv_{E\cup P\cup\Gamma\cup(\forall X)l'\overset{\circ}{=}r'},$$

deci

$$\equiv_{E\cup\Gamma\cup(\forall X)l\overset{\circ}{=}r} = \equiv_{E\cup P\cup\Gamma\cup(\forall X)l'\overset{\circ}{=}r'} .\Box$$

Propoziţie 2.11.7 *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 2.10.6 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 $a \overset{\circ}{=} b$ 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 $fn(a) \overset{\circ}{=} fn(b)$ sau $fn(a) \overset{\circ}{=} fn(b)$ ceea ce face ca a şi b să se întâlnească prin rescriere.

Pentru a obţine concluzia este suficient să mai aplicăm teorema 2.10.6 a perechilor critice pentru cazul ecuaţional. $\Box$

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 bine fondată.

Criteriul B: există o relaţie bine fondată $\vdash$ pe o mulţime M şi o funcţie $\rho: \ A \longrightarrow M$ cu proprietatea pentru orice $n \geq 1$ şi $\sigma \in \Sigma_{s_1 s_2 \ldots s_n, s}$,

$\rho(a) \vdash \rho(d)$ şi $\rho(A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n)) \neq \rho(A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n))$ implică $\rho(A_\sigma(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n)) \ \vdash \ \rho(A_\sigma(a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_n))$,

Aplicarea criteriului la alegerea regulilor face posibilă aplicarea teoremei 2.11.1 programului final. Se observă deci că programul final se termină.

Chapter 3

PROGRAMARE LOGICĂ ECUAŢIONALĂ

3.1 IN PRIMUL RAND SEMANTICA

Reamintim câteva concepte introduse în capitolul precedent pe care le vom folosi în continuare.

Fixăm o signatură multisortată (S, Σ).

O *ecuaţie condiţionată* este

$$(\forall X) l \stackrel{\circ}{=}_s r \textbf{ if } H$$

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 $Sen(T_\Sigma(X)) = T_\Sigma(X) \times T_\Sigma(X)$.

Algebra $\mathcal{D}$ *satisface* ecuaţia condiţionată $(\forall X) l \stackrel{\circ}{=}_s r$ **if** H, fapt notat prin

$$\mathcal{D} \models_\Sigma (\forall X) l \stackrel{\circ}{=}_s r \textbf{ if } H$$

dacă pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ pentru care $h_t(u) = h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$, are loc egalitatea $h_s(l) = h_s(r)$.

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

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.

Vom lucra într-o Σ-algebră $\mathcal{A}$.

Congruenţa semantică $\equiv_\Gamma$ a lui $\mathcal{A}$, notată şi $\equiv_\Gamma^{\mathcal{A}}$ atunci când este pericol de confuzie, a fost definită prin

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

Fie $G \subseteq Sen(T_\Sigma(X))$ o mulţime de egalităţi formale din $T_\Sigma(X)$. Reamintim că

$$\Gamma \models (\forall X) G$$

dacă şi numai dacă orice Γ–algebră satisface $(\forall X)G$, adică $h_s(u) = h_s(v)$ pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{M} \models \Gamma$ şi orice $u \stackrel{\circ}{=}_s t \in G$.

Observaţia 3.1.1 $\Gamma \models (\forall X)G$ *dacă şi numai dacă* $G \subseteq \equiv_\Gamma^{T_\Sigma(X)}$.

Demonstraţie: $\Gamma \models (\forall X)G$ este succesiv echivalent cu
- pentru orice Σ–algebră $\mathcal{M}$ dacă $\mathcal{M} \models \Gamma$, atunci $\mathcal{M} \models (\forall X)G$
- pentru orice Σ–algebră $\mathcal{M}$ care satisface Γ şi pentru orice morfism $h : T_\Sigma(X) \rightarrow \mathcal{M}$ are loc egalitatea $h_s(u) = h_s(v)$ pentru orice $u \stackrel{\circ}{=}_s v$ din G,
- $u \equiv_\Gamma v$ oricare ar fi $u \stackrel{\circ}{=}_s v \in G$,
- $G \subseteq \equiv_\Gamma^{T_\Sigma(X)}$. $\Box$

Propoziţie 3.1.1 *Fie $h : \mathcal{A} \to \mathcal{B}$ un Σ-morfism. Dacă $a \equiv_\Gamma^\mathcal{A} c$, atunci $h(a) \equiv_\Gamma^\mathcal{B} h(c)$.*

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 $h(a) \equiv_\Gamma^\mathcal{B} h(b)$. $\Box$

Această propoziţie arată de fapt corectitudinea regulii logicii ecuaţionale globale: dacă $h : \mathcal{A} \longrightarrow \mathcal{B}$ este un morfism de Σ-algebre, atunci $(\forall\mathcal{A})a \stackrel{\circ}{=}_s a'$ implică $(\forall\mathcal{B})h_s(a) \stackrel{\circ}{=}_s h_s(a')$.

Corolar 3.1.1 *Dacă $\Gamma \models (\forall X)G$ şi $h : T_\Sigma(X) \to T_\Sigma(Y)$ este un Σ-morfism, atunci $\Gamma \models (\forall Y)h(G)$.*

Demonstraţie: Din ipoteză, conform observaţiei, deducem că $G \subseteq \equiv_\Gamma^{T_\Sigma(X)}$. Aplicând Propoziţia 3.1.1, deducem $h(G) \subseteq \equiv_\Gamma^{T_\Sigma(Y)}$, deci conform observaţiei $\Gamma \models (\forall Y)h(G)$. $\Box$

Fie $\mathcal{A}_\Gamma$ factorizarea lui $\mathcal{A}$ prin congruenţa sa semantică $\equiv_\Gamma$ şi $\eta_\mathcal{A} :\ \mathcal{A} \longrightarrow \mathcal{A}_\Gamma$ morfismul de factorizare. Reamintim principalele proprietăţi ale acestei algebre

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

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

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

3.2 TEOREMELE LUI HERBRAND

3.2.1 Introducere

Morfismele de algebre se extind natural pentru perechi de elemente sau egalităţii formale şi pentru mulţimi de perechi de elemente sau mulţimi de egalităţii formale. Dacă $h : \mathcal{A} \longrightarrow \mathcal{B}$ este un Σ–morfism, atunci pentru orice $a, b \in \mathcal{A}$ prin definiţie $h((a, b)) = (h(a), h(b))$ sau $h_s(a \stackrel{\circ}{=}_s b) = (h_s(a) \stackrel{\circ}{=}_s h_s(b))$. De asemenea în locul notaţiei $h_s(a) = h_s(b)$ vom mai utiliza $h_s((a, b)) \in \Delta_B$.

Dacă G este o mulţime de egalităţi formale din $\mathcal{A}$ prin definiţie

$$h(G) = \{h_s(a) \stackrel{\circ}{=}_s h_s(b) : a \stackrel{\circ}{=}_s b \in G\}.$$

In plus $h(G) \subseteq \Delta_B$ înseamnă $h_s(a) = h_s(b)$ pentru orice $a \stackrel{\circ}{=}_s b \in G$.

Programarea logică ecuaţională se ocupă de rezolvarea unui sistem finit de ecuaţii format din egalităţi formale în care variabilele sunt privite ca necunoscute. De aici provine şi denumirea de ecuaţii folosită încă din capitolele precedente. De fapt rezolvarea unui astfel de sistem a fost făcută în algebra liberă din care sunt termenii ecuaţiilor aşa cum a fost prezentat în subcapitolul 2.6 privind unificarea.

Se caută soluţii în anumite Σ–algebre $\mathcal{D}$. Adică pentru o astfel de algebră $\mathcal{D}$ se caută valori date variabilelor în $\mathcal{D}$ pentru care egalităţile formale devin egalităţi adevărate în $\mathcal{D}$. Urmează formalizarea celor de mai sus.

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).

Definiţia 3.2.1 *Fie $\mathcal{A}$ o Σ-algebră şi G 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 $h_s(l) = h_s(r)$ pentru orice $l \stackrel{\circ}{=}_s r$.*

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 $\Gamma \models (\exists X)G$.

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

3.2.2 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 3.2.2 *Fie G o mulţime de egalităţi formale între elemente din $T_\Sigma(X)$. 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 $\Gamma \models (\forall\emptyset)\psi(G)$.*

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

(2) $\Rightarrow$ (3) Conform ipotezei, există morfismul $h : T_\Sigma(X) \to T_{\Sigma,\Gamma}$ astfel încât $h_s(l) = h_s(r)$ pentru orice $l \overset{\circ}{=}_s r \in G$. Pentru că $T_\Sigma(X)$ este algebră liberă şi deci proiectivă conform propoziţiei 1.7.2, deoarece morfismul η_{T_Σ} este surjectiv pe componente există morfismul $\psi : T_\Sigma(X) \to T_\Sigma$ astfel încât $\psi;\eta_{T_\Sigma} = h$.

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

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

Prin urmare $\eta_{T_\Sigma}(\psi(l)) = \eta_{T_\Sigma}(\psi(r))$ pentru orice $l \overset{\circ}{=} r \in G$. Deoarece $\eta : T_\Sigma \longrightarrow T_{\Sigma,\Gamma}$ este morfismul de factorizare al algebrei T_Σ la congruenţa semantică $\equiv_\Gamma$ deducem că $\psi(l) \equiv_\Gamma \psi(r)$ pentru orice $l \overset{\circ}{=} r \in G$, adică $\psi(G) \subseteq \equiv_\Gamma^{T_\Sigma}$. Din observaţia 3.1.1 deducem $\Gamma \models (\forall\emptyset)\psi(G)$.

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

Notăm cu $\alpha_\mathcal{M} : T_\Sigma \to \mathcal{M}$ unicul morfism existent între cele două algebre. Arătăm că morfismul

$$\psi;\alpha_\mathcal{M} : T_\Sigma(X) \to \mathcal{M}$$

este soluţie, adică $\alpha_\mathcal{M}(\psi(l)) = \alpha_\mathcal{M}(\psi(r))$, pentru orice $l \overset{\circ}{=} r \in G$.

Deoarece $\mathcal{M} \models \Gamma$, din ipoteză rezultă că $\mathcal{M} \models (\forall\emptyset)\psi(G)$. Utilizând în definiţia satisfacerii morfismul $\alpha_\mathcal{M} : T_\Sigma \to \mathcal{M}$ deducem că $\alpha_\mathcal{M}(\psi(l)) = \alpha_\mathcal{M}(\psi(r))$ pentru orice $l \overset{\circ}{=} r \in G$, prin urmare $(\psi;\alpha_\mathcal{M})(l) = (\psi;\alpha_\mathcal{M})(r)$, pentru orice $l \overset{\circ}{=} r \in G$.

Deci $\mathcal{M} \models (\exists X)G$ şi demonstraţia se încheie. $\square$

Prima echivalenţă a teoremei arată că problema programării logice se reduce la rezolvarea ei în Γ-algebra iniţială $T_{\Sigma,\Gamma}$, 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 prin rescriere şi cu conceptul de soluţie.

A treia afirmaţie mai arată că pentru existenţa soluţiei în T_Σ este necesar ca suporturile din Σ-algebra iniţială T_Σ, corespunzătoare sorturilor variabilelor cuantificate existenţial, să 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 sortului lui y este vid în T_Σ. În continuare ne vor interesa şi astfel de soluţii, chiar dacă ecuaţia $x = f(y)$ are soluţie numai în algebrele în care suportul corespunzător sortului lui y este nevid.

3.3 REGULILE PROGRAMĂRII LOGICE

În continuare vom considera Γ o mulţime de ecuaţii condiţionate şi G o mulţime finită de ecuaţii, egalităţi formale, din $T_\Sigma(X)$. *Problema programării logice* este $\Gamma \models (\exists X)G$.

3.3.1 Soluţii

Revenim asupra conceptului de soluţie deoarece în cele ce urmează vom utiliza o variantă puţin modificată, fapt datorat contextului în care suntem: soluţiile se caută în Γ–algebre. Comentariile următoare se bazează pe diagrama comutativă din figura 3.2. Observăm că:

1. Dat morfismul f, morfismul s este chiar compunerea lui f cu $\eta_{T_\Sigma(Y)}$.

2. Dat morfismul s existenţa morfismului f care face diagrama comutativă rezultă din proiectivitatea Σ-algebrei libere $T_\Sigma(X)$.

$$\begin{tikzcd}
T_\Sigma(X) \arrow[rr, "\exists f"] \arrow[dr, "s"'] & & T_\Sigma(Y) \arrow[dl, "\eta_{T_\Sigma(Y)}"] \\
& T_\Sigma(Y)_\Gamma &
\end{tikzcd}$$

Figure 3.2: Folosind proiectivitatea Σ-algebrei $T_\Sigma(X)$

Propoziţie 3.3.1 *Cu notaţiile din figura 3.2 următoarele afirmaţii sunt echivalente*

1. $s_t(u) = s_t(v)$ *pentru orice* $u \overset{\circ}{=}_t v \in G$

2. $f_t(u) \equiv_\Gamma f_t(v)$ *pentru orice* $u \overset{\circ}{=}_t v \in G$

Demonstraţie: Egalitatea $s_t(u) = s_t(v)$ este, datorită comutativităţii diagramei, echivalentă cu $\eta_{T_\Sigma(Y)}(f_t(u)) = \eta_{T_\Sigma(Y)}(f_t(v))$, adică cu $f_t(u) \equiv_\Gamma f_t(v)$ deoarece două elemente sunt egalate de surjecţia naturală $\eta_{T_\Sigma(Y)}$ dacă şi numai dacă sunt echivalente semantic. $\Box$

Prima condiţie spune că s este soluţie pentru $(\exists X)G$ în $T_\Sigma(Y)_\Gamma$. Vom prefera să folosim pentru conceptul de soluţie a doua condiţie din propoziţia de mai sus, fapt ce conduce la următoarea definiţie în care facem o generalizare înlocuind algebra liberă $T_\Sigma(Y)$ cu o algebră.

Definiţie 3.3.1 Σ-*morfismul* $f : T_\Sigma(X) \longrightarrow \mathcal{A}$ *se numeşte* **soluţie** *pentru* $(\exists X)G$ *dacă* $f(G) \subseteq \equiv_\Gamma^{\mathcal{A}}$.

Observaţia 3.3.2 *Dacă* Δ *este o mulţime de egalităţi adevărate din* $T_\Sigma(X)$*, atunci identitatea lui* $T_\Sigma(X)$ *este soluţie pentru* $(\exists X)\Delta$.

Demonstraţie: $1_{T_\Sigma(X)}(\Delta) \subseteq \Delta_{T_\Sigma(X)} \subseteq \equiv_\Gamma^{T_\Sigma(X)}$.

Observaţia 3.3.3 *Un Σ-morfism $f : T_\Sigma(X) \longrightarrow \mathcal{A}$ este soluţie* ***soluţie*** *pentru $(\exists X)G$ dacă $\Gamma \models (\forall \mathcal{A})f(G)$, adică pentru orice morfism $h : \mathcal{A} \longrightarrow \mathcal{M} \models_\Sigma \Gamma$, $h(f(l)) = h(f(r))$ pentru orice $l \overset{\circ}{=} r \in G$.*

Demonstraţie: Este suficient să folosim definiţia congruenţei semantice. Vezi şi demonstraţia observaţiei 3.1.1

Propoziţie 3.3.1 *Compunerea unei soluţii cu orice Σ-morfism este soluţie.*

Demonstraţie: Fie $s : T_\Sigma(X) \longrightarrow T_\Sigma(Y)$ o soluţie pentru $(\exists X)G$ şi $h : T_\Sigma(Y) \rightarrow T_\Sigma(Z)$ un morfism. Vrem să arătăm că $s;h : T_\Sigma(X) \rightarrow T_\Sigma(Z)$ este soluţie pentru $(\exists X)G$, adică $\Gamma \models (\forall Z)(s;h)(G)$.

Deoarece $s : T_\Sigma(X) \longrightarrow T_\Sigma(Y)$ este soluţie pentru $(\exists X)G$, atunci $\Gamma \models (\forall Y)s(G)$. Corolarul 3.1.1 implică $\Gamma \models (\forall Z)h(s(G))$.

Deci $\Gamma \models (\forall Z)(s;h)(G)$ şi astfel $s;h$ este soluţie pentru $(\exists X)G$. $\Box$

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*.

3.3.2 Reguli de deducţie

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

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

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

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

$$h(C[a]) = h^\bullet(C)[h(a)].$$

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 de egalităţi adevărate, în particular la mulţimea vidă. În acest moment, 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*

$$G \cup \{l \overset{\circ}{=}_s r\} \to_{re} \theta(G),$$

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*

$$G \cup \{l \overset{\circ}{=}_s r\} \to_r \theta(G),$$

cu morfismul calculat θ.

Observăm că regula reflexiei permite eliminarea din sistem a egalităţilor adevărate, caz în care morfismul calculat este o identitate.

Regula pararescrierii: *Fie* $G \subseteq Sen(T_\Sigma(X))$, $(\forall Y)l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$ *şi morfismul* $\theta : T_\Sigma(Y) \to T_\Sigma(X)$. *Dacă* C *este un context extins cu variabila distinsă* $\bullet$ *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* $(\forall Y)l \overset{\circ}{=}_s r$ **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* $\bullet$ *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* $(\forall Y)l \overset{\circ}{=}_s r$ **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* $\bullet$ *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.

Nu recomandăm utilizarea variantei extinse a regulilor şi nici a regulii morfismului deoarece ele pot conduce la pierderea parţială a generalităţii soluţiei. Aceste regulile au un rol teoretic.

3.4 CORECTITUDINEA REGULILOR PROGRAMĂRII LOGICE

Definiţie 3.4.1 *Fie R o regulă de deducţie. Să presupunem că aplicând regula R obţinem $G \rightarrow_R G'$ cu morfismul calculat $\theta : T_\Sigma(X) \longrightarrow T_\Sigma(Y)$. Spunem că regula R este o* ***regulă corectă*** *dacă este îndeplinită următoarea condiţie: dacă s este o soluţie pentru $(\exists Y)G'$, atunci $\theta; s$ este soluţie pentru $(\exists X)G$.*

Observăm că aplicarea unei reguli corecte $G \rightarrow_R G'$ reduce problema găsirii unei soluţii pentru $(\exists X)G$ la găsirea unei soluţii pentru $(\exists Y)G'$.

Dacă se aplică numai reguli corecte ajungându-se, în final la o mulţime formată doar din egalităţi adevărate, în particular la mulţimea vidă de ecuaţii, 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 $\models (\forall X) l \overset{\circ}{=}_s l$, orice soluţie pentru G este soluţie şi pentru $G \cup \{l \overset{\circ}{=}_s l\}$. Prin urmare *eliminarea egalităţilor adevărate* este o regulă corectă.

Propoziţie 3.4.1 *Regula morfismului este corectă.*

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

Propoziţie 3.4.2 *Regula reflexiei extinse este corectă.*

Demonstraţie: Observăm că regula reflexiei extinse se poate obţine din regula morfismului şi eliminarea egalităţilor adevărate. Cum regula morfismului şi eliminarea egalităţilor adevărate sunt corecte, rezultă că şi regula reflexiei extinse este corectă. $\square$

Corolar 3.4.1 *Regula reflexiei este corectă.*

Reamintim că $S : T_\Sigma(X) \longrightarrow \mathcal{B}$ este soluţie pentru $(\exists X)G$ dacă şi numai dacă $S(G) \subseteq \equiv^{\mathcal{B}}_\Gamma$.

Propoziţie 3.4.1 *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 $(\forall Y) l \overset{\circ}{=}_s r$ **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 $(\exists X)G \cup \theta(H) \cup \{C[\theta_s(r)]\}$, prin urmare

$$S(G \cup \theta(H) \cup \{C[\theta_s(r)]\}) \subseteq \equiv^{\mathcal{B}}_\Gamma. \qquad (1)$$

Trebuie să arătăm că $S : T_\Sigma(X) \rightarrow \mathcal{B}$ este soluţie pentru $(\exists X)(G \cup C[\theta_s(l)])$, adică

$$S(G \cup \{C[\theta_s(l)]\}) \subseteq \equiv^{\mathcal{B}}_\Gamma .$$

Deoarece $S(G) \subseteq \equiv^{\mathcal{B}}_\Gamma$ rezultă direct din (1) mai avem de arătat că $S(C[\theta_s(l)]) \in \equiv^{\mathcal{B}}_\Gamma$.

Fie $h : \mathcal{B} \rightarrow \mathcal{M} \models \Gamma$. Din (1) folosind morfismul $S; h : T_\Sigma(X) \longrightarrow \mathcal{M} \models \Gamma$ deducem

$$(S; h)(G) \cup (\theta; S; h)(H) \cup \{(S; h)(C[\theta_s(r)])\} \subseteq \Delta_M.$$

Prin urmare:

$$(\theta; S; h)(H) \subseteq \Delta_M, \qquad (2)$$
$$(S; h)(C[\theta_s(r)]) \in \Delta_M. \qquad (3)$$

Deoarece $\mathcal{M} \models (\forall Y) l \stackrel{\circ}{=}_s r$ **if** H, folosind morfismul $\theta; S; h : T_\Sigma(Y) \to \mathcal{M}$ şi relaţia (2), deducem că

$$(\theta; S; h)_s(l) = (\theta; S; h)_s(r). \qquad (4)$$

Observăm că

$$h(S(C[\theta_s(l)])) = (S;h)(C[\theta_s(l)]) = (S;h)^\bullet(C)[(S;h)(\theta_s(l))] = (S;h)^\bullet(C)[(\theta_s; S; h)(l)] =$$
$$= (S;h)^\bullet(C)[(\theta_s; S; h)(r)] = (S;h)^\bullet(C)[(S;h)(\theta_s(r))] = (S;h)(C[\theta_s(r)]).$$

Din (3) deducem că $h(S(C[\theta_s(l)])) \in \Delta_M$.

Deci $S(C[\theta_s(l)]) \in \equiv_\Gamma^{\mathcal{B}}$. $\Box$

Propoziţie 3.4.3 *Regula paramodulaţiei extinse se poate obţine din regula morfismului şi regula pararescrierii.*

Demonstraţie: Fie $(\forall Y) l \stackrel{\circ}{=}_s r$ **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 $\bullet$ 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:

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

Acum putem aplica regula pararescrierii şi obţinem:

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

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

Propoziţie 3.4.4 *Regula paramodulaţiei extinse este corectă.*

Demonstraţie: Din Propoziţia 3.4.3 ştim că orice paramodulaţie extinsă se poate obţine din regula morfismului şi regula pararescrierii. Din Propoziţiile 3.4.1 şi 3.4.1 ştim că regulile morfismului şi pararescrierii sunt corecte, de unde rezulta că şi regula paramodulaţiei extinse este corectă. $\Box$

Corolar 3.4.2 *Regula paramodulaţiei este corectă.*

3.5 COMPLETITUDINEA PARAMODULATIEI

3.5.1 Legături între regulile de deducţie

În continuare vom prezenta alte legături dintre regulile de deducţie pentru programarea logică.

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

$$G \cup \{l \doteq_s r\} \to_m \theta(G) \cup \{\theta_s(l) \stackrel{\circ}{=} \theta_s(r)\},$$

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

Propoziţie 3.5.1 *Pararescrierea este un caz particular de paramodulaţie extinsă.*

Demonstraţie: Considerăm pararescrierea

$$G \cup \{C[h_s(l)]\} \rightarrow_{pr} G \cup h(H) \cup \{C[h_s(r)]\},$$

unde $(\forall Y)l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$ şi $h : T_\Sigma(Y) \to T_\Sigma(X)$ este un Σ-morfism (presupunem $X \cap Y = \emptyset$).

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 $Sen(T_\Sigma(X \cup \{\bullet\}))$ şi astfel avem $\theta^\bullet(C) = C$.

Luăm $a = h_s(l)$. 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:

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

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

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

Propoziţie 3.5.2 *Dacă pentru orice clauză* $(\forall Y)l \overset{\circ}{=}_s r$ **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 $u : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ un unificator pentru l şi a. Deoarece $u_s(l) = u_s(a) = u_s(h_s(l))$ rezultă că $u(y) = u(h(y))$ pentru orice variabilă y care apare în l. Deoarece 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 $T_\Sigma(X)$. Evident $u_{|X}(t) = u(t)$, pentru orice $t \in T_\Sigma(X)$.

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 $u = \theta; u_{|X}$. $\square$

Lemă 3.5.1 *Dacă* $(\forall Y)t \overset{\circ}{=}_s t \in \Gamma$, *atunci* $G \rightarrow_p (x \leftarrow t)(G)$, *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 $(\forall Y)t \doteq_s t \in \Gamma$, $a = x$, $\theta = \mathbf{CGU}\{a, l\} = \mathbf{CGU}\{x, t\} = x \leftarrow t$, obţinem:

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

Penultima egalitate este adevărată deoarece x nu apare în t. $\square$

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

Demonstraţie Primele două afirmaţii care urmează dovedesc că axiomele lemei substituţiei, mai sărace decât cele ale lemei 3.5.1, sunt suficiente pentru a demonstra concluzia lemei 3.5.1:

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

În cazul în care x apare în G şi $x \neq y$ se aplica Lema 3.5.1. În rest, evident.

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

Vom demonstra acest lucru prin inducţie după structura termenului t.

Primul pas al inducţiei este chiar (1).

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

Dacă x nu apare în G, atunci nu avem nimic de demonstrat. Presupunem că x apare în G şi folosind $(\forall x_1\ \forall x_2 \ldots \forall x_n)f(x_1, x_2, \ldots, x_n) \doteq f(x_1, x_2, \ldots, x_n) \in \Gamma$, unde variabilele $x_1, \ldots, x_n$ sunt noi, şi Lema 3.5.1 deducem

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

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

Mai observăm că

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

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

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

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

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

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

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

$$G' \stackrel{*}{\rightarrow}_p (z_1 \leftarrow h(x_1))(G) \stackrel{*}{\rightarrow}_p \ldots \stackrel{*}{\rightarrow}_p (z_n \leftarrow h(x_n))(\ldots(z_1 \leftarrow h(x_1))(G)\ldots) = h(G). \square$$

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 $h : T_\Sigma(\{x, y\}) \rightarrow T_\Sigma(\{x, y, z\})$, $h(x) = z$, $h(y) = x$ şi $G = \{x \stackrel{\circ}{=} y\}$, atunci:

$$h(G) = (h(x) \stackrel{\circ}{=} h(y)) = (z \stackrel{\circ}{=} x),$$

$$(x \leftarrow h(x))((y \leftarrow h(y))(G)) = (x \leftarrow h(x))(x \stackrel{\circ}{=} x) = (z \stackrel{\circ}{=} z).$$

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

Putem sintetiza legăturile găsite între regulile de deducţie pentru programarea logică prin Figura 3.3.

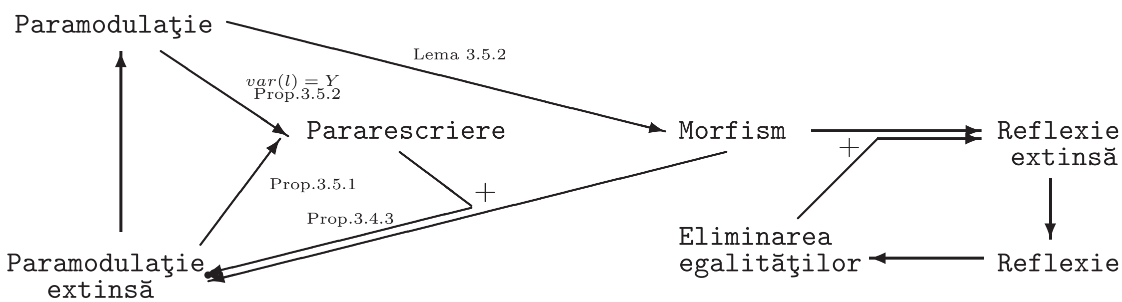


Figure 3.3: Legăturile dintre regulile de deducţie

3.5.2 Amintiri despre rescriere

Reamintim câteva fapte din secţiunea 2.3.4 privind rescrierile pe care le folosim în demonstraţia următoare. Lucrăm întro Σ–algebră $\mathcal{A}$.

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

Dată o relaţie Q pe A, relaţia

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

este cea mai mică relaţie inchisă la contexte care include Q.

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

Fixăm Γ.

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

$$Q_0 = \emptyset$$

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

Prin definiţie Q este reuniunea şirului crescător definit mai sus.

În cazul în care Q este definită ca mai sus, în loc de $\overset{*}{\longrightarrow}_Q$ vom prefera să scriem $\overset{*}{\Longrightarrow}_\Gamma$.

Relaţia $\overset{*}{\Longrightarrow}_\Gamma$ este denumită Γ–rescriere sau mai scurt rescriere.

Fie $\downarrow_\Gamma$ relaţia de întâlnire ataşată lui $\overset{*}{\Rightarrow}_\Gamma$. Prin definiţie $a \downarrow_\Gamma b$ dacă şi numai dacă există c astfel încât $a \overset{*}{\Rightarrow}_\Gamma c$ şi $b \overset{*}{\Rightarrow}_\Gamma c$. Reamintim că întâlnirea prin rescriere $\downarrow_\Gamma$ este corectă pentru $\equiv_\Gamma$, adică $\downarrow_\Gamma \subseteq \equiv_\Gamma$.

Teorema 3.5.1 *Dacă $\overset{*}{\Rightarrow}_\Gamma$ este confluentă, atunci $\downarrow_\Gamma$ este completă.*

3.5.3 Prolog

Reamintim că Δ înseamnă o mulţime de egalităţi adevărate.

Teoremă 3.5.1 *Dacă $a \downarrow_\Gamma d$, atunci $\{a \doteq_s d\} \overset{*}{\to}_{pr} \Delta$.*

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

Arătăm prin inducţie după n că $\{a \overset{\circ}{=}_s d\} \overset{*}{\to}_{pr} \Delta$.

1. Pentru $n = 0$ observăm că $\to_{Q_0}$ este vidă şi $\overset{*}{\to}_{Q_0}$ este relaţia de egalitate. Prin urmare $a = d$, concluzia fiind evidentă.

2. Presupunem, prin ipoteza de inducţie, că pentru orice x, y dacă $x \downarrow_{Q_n} y$, atunci $\{x \overset{\circ}{=}_s y\} \overset{*}{\to}_{pr} \Delta$. Presupunem că $a \downarrow_{Q_{n+1}} d$.

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

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

Deoarece $a \to_{Q_{n+1}} w$ există $(\forall Y) l \overset{\circ}{=}_{s'} r$ **if** $H \in \Gamma$, morfismul $h : T_\Sigma(Y) \to T_\Sigma(X)$ astfel încât $h(u) \downarrow_{Q_n} h(v)$, pentru orice $u \overset{\circ}{=} v \in H$, şi contextul c în $T_\Sigma(X \cup \{\bullet\})$ astfel încât $a = c[h_{s'}(l)]$ şi $w = c[h_{s'}(r)]$. Observăm că $\{c[h_{s'}(l)] \overset{\circ}{=}_s d\} \to_{pr} h(H) \cup \{c[h_{s'}(r)] \overset{\circ}{=}_s d\} = h(H) \cup \{w \overset{\circ}{=}_s d\}$.

Prin urmare, deoarece $\{w \overset{\circ}{=}_s d\} \overset{*}{\to}_{pr} \Delta$, deducem că $\{a \overset{\circ}{=}_s d\} \overset{*}{\to}_{pr} h(H) \cup \Delta$. Deoarece $h(u) \downarrow_{Q_n} h(v)$, pentru orice $u \overset{\circ}{=} v \in H$, din ipoteza de inducţie deducem $h(H) \overset{*}{\to}_{pr} \Delta$, deci $\{a \overset{\circ}{=}_s d\} \overset{*}{\to}_{pr} \Delta$. $\square$

Corolar 3.5.1 *Dacă G este o mulţime finită astfel încât $G \subseteq \downarrow_\Gamma$, atunci $G \overset{*}{\to}_{pr} \Delta$.*

3.5.4 Completitudinea

Observăm că identitatea lui $T_\Sigma(Y)$ este soluţie pentru $(\exists Y)\Delta$ deoarece $\Gamma \models (\forall Y)\Delta$. Prin urmare, dacă $G \overset{*}{\to}_p \Delta$ 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 3.5.2 (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 $\sigma : T_\Sigma(X) \to T_\Sigma(Y)$ o soluţie pentru $(\exists X)G$. 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 $\sigma(G) \overset{*}{\to}_{pr} \Delta$.

Deoarece pentru orice axiomă $(\forall Y) l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$, orice variabilă din Y apare în l, deducem, din Propoziţia 3.5.2, că orice pararescriere este un caz particular de paramodulaţie în care substituţia calculată este o identitate. În concluzie, putem scrie $\sigma(G) \overset{*}{\to}_p \Delta$ cu substituţia calculată identitatea.

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

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

3.6 COMPLETITUDINEA NARROWINGULUI

3.6.1 Introducere

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

Fie $(\forall Y) l \overset{\circ}{=}_s r$ **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

$$i_X : T_\Sigma(X) \longrightarrow T_\Sigma(X \cup Y) \text{ şi } i_Y : T_\Sigma(Y) \longrightarrow T_\Sigma(X \cup Y)$$

morfismele incluziune.

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

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

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

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

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

Propoziţie 3.6.1 *Dacă pentru orice clauză $(\forall Y)l \overset{\circ}{=}_s r$ **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 3.5.1 şi 3.5.2 şi din egalitatea $a = h_s(l)$ din faptul că l nu este o variabilă rezultă că nici a nu este variabilă.

3.6.2 Amintiri despre formele normale

Reamintim că $\overset{*}{\Rightarrow}_\Gamma$ este relaţia de rescriere în A. Faptele de mai jos au fost demonstrate în sectiunea 2.4.3.

Definiţia 3.6.2 Elementul $n \in A$ se numeşte **o formă normală** pentru dacă

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

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

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

Observaţia 3.6.3 *Dacă $a \overset{*}{\Rightarrow}_\Gamma d$, atunci $fn(a) = fn(d)$.*

Observaţia 3.6.4 *Axioma* **FN!** *implică $\overset{*}{\Rightarrow}_\Gamma$ este confluentă.*

Observaţia 3.6.5 *Funcţia $fn: \ A \longrightarrow N$ este surjectivă şi $a \downarrow_\Gamma d \Leftrightarrow fn(a) = fn(d)$.*

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

3.6.3 Lema de ridicare

Definiţia 3.6.6 O substituţie se numeşte normală dacă duce orice variabilă într-un element în formă normală.

Propoziţie 3.6.7 *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 $\rho : T_\Sigma(X \cup Y) \longrightarrow T_\Sigma(Z)$ o substituţie a cărei restricţie la $T_\Sigma(X)$ este normală şi $\rho(a) = \rho(l)$.

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

Demonstraţie: Existenţa lui $\mathbf{CGU}(a, l)$ rezultă din faptul că ρ este unificator pentru a şi l. În plus, conform observaţiei de la finalul secţiunii 2.6, putem să presupunem că $V \subseteq X \cup Y$ şi că $\psi(v) = v$ pentru orice $v \in V$.

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

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

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

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

Propoziţie 3.6.8 *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 formale din $T_\Sigma(X)$.*

Dacă $\theta : T_\Sigma(X) \longrightarrow T_\Sigma(Z)$ 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 $(\forall Y) l \overset{\circ}{=} r$ if $H \in \Gamma$ regula şi $\eta : T_\Sigma(Y) \longrightarrow T_\Sigma(Z)$ substituţia utilizate în pararescriere. 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:

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

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

Fie $\psi = \mathbf{CGU}(a, l) : T_\Sigma(X \cup Y) \longrightarrow T_\Sigma(V)$ şi $\theta' : T_\Sigma(V) \longrightarrow T_\Sigma(Z)$ unica substituţie cu proprietatea $\psi\theta' =< \theta, \eta >$. Deoarece restricţia θ a lui $< \theta, \eta >$ la $T_\Sigma(X)$ este normală conform ipotezei, şi $< \theta, \eta > (a) =< \theta, \eta > (l)$, aplicând propoziţia 3.6.7 rezultă normalitatea lui θ'.

Notând cu $\varphi : T_\Sigma(X) \longrightarrow T_\Sigma(V)$ 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:

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

Propoziţie 3.6.9 *Pentru orice $(\forall Y) l \overset{\circ}{=} r$ **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 $\theta : T_\Sigma(X) \longrightarrow T_\Sigma(Z)$ este normală şi

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

atunci

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

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

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

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 $(\varphi\sigma)\epsilon = \varphi\theta' = \theta$. $\square$

3.6.4 Epilog

Propoziţie 3.6.10 *Fie $G \subseteq T_\Sigma(X) \times T_\Sigma(X)$ finită. Dacă morfismul $h : T_\Sigma(X) \longrightarrow T_\Sigma(Y)$ este un unificator pentru G, atunci $G \xrightarrow{*}_r \emptyset$ cu substituţia calculată s' pentru care există morfismul f cu proprietatea $s'; f = h$.*

Demonstraţie: Reamintim definiţia reflexiei:
"Dacă $\theta : \mathcal{A} \longrightarrow \mathcal{B}$ este cel mai general unificator pentru l si r, atunci $G \cup \{l =_t r\} \longrightarrow_r \theta(G)$ cu morfismul calculat θ."

Vom demonstra propoziţa prin inducţie după numărul elementelor mulţimii G.

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

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

Deoarece $v(u(G')) = h(G')$ şi h este unificator pentru G rezultă că v este unificator pentru $u(G')$ care are mai puţire elemente decât G. 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 $h = u; v = (u; w); f$. $\square$

3.6.5 Completitudine

Ipoteze. Pentru orice $(\forall Y)l \stackrel{\circ}{=} r$ **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 $s : T_\Sigma(X) \longrightarrow T_\Sigma(Z)$ o soluţie pentru $(\exists X)G$. Cu ipoteza **FN!** pentru Γ soluţia s se poate normaliza obţinând soluţia normală $s' : T_\Sigma(X) \longrightarrow T_\Sigma(Z)$ 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 $u \stackrel{\circ}{=}_t v \in G$ 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 $s'(G) \subseteq \equiv_\Gamma$. Fie $u \stackrel{\circ}{=}_t v \in G$. Deoarece s este soluţie pentru $(\exists X)G$ deducem $s(u) \equiv_\Gamma s(v)$. Deoarece $\stackrel{*}{\Rightarrow}_\Gamma \subseteq \equiv_\Gamma$ din $s(u) \stackrel{*}{\Rightarrow}_\Gamma s'(u)$ şi $s(v) \stackrel{*}{\Rightarrow}_\Gamma s'(v)$ deducem $s(u) \equiv_\Gamma s'(u)$ şi $s(v) \equiv_\Gamma s'(v)$. Deoarece $\equiv_\Gamma$ este relaţie de echivalenţă obţinem $s'(u) \equiv_\Gamma s'(v)$. Deci s' este soluţie.

Propoziţie 3.6.11 *Orice soluţie normală se obţine prin particularizarea unei soluţii obţinute cu narrowing şi reflexie.*

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

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

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

$$G' \stackrel{*}{\longrightarrow}_r \emptyset \text{ cu morfismul calculat } \theta$$

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

3.7 REZOLUŢIE À LA PROLOG

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

3.7.1 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 finită de atomi.

Ţelul, scopul, este o mulţime finită 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 $\theta(H \cup T)$.

3.7.2 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 $\pi(v) \overset{\circ}{=}_b t$ unde t este o constantă de sort b reprezentând adevărul.

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

Propoziţie 3.7.1 *În varianta ecuaţională, rezoluţia se poate realiza prin narrowing şi eliminarea egalităţilor reale.*

Demonstraţie: Fie G o mulţime de atomi şi $(\forall Y)\,\pi(v)$ 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 $G^e = G'^e \cup \{\pi(s) \overset{\circ}{=}_b t\}$ şi $(\forall Y)\,\pi(v) \overset{\circ}{=}_b t$ if H^e.
Alegem $a = \pi(s)$, $l = \pi(v)$ şi observăm că $\theta = CGU(s,v) = CGU(\pi(s), \pi(v))$.
Cum a nu este variabilă rezultă că putem aplica narrowing-ul:

$$\begin{array}{rcl} G'^e \cup \{\pi(s) \overset{\circ}{=}_b t\} & \longrightarrow_n & \theta(G'^e \cup H^e \cup \{t \overset{\circ}{=}_b t\}) \;= \\ & & \theta((G' \cup H)^e \cup \{t \overset{\circ}{=}_b t\}) \;= \\ & & [\theta(G' \cup H)]^e \cup \{t \overset{\circ}{=}_b t\} \end{array}$$

În urma eliminării egalităţii adevărate $t \overset{\circ}{=}_b t$ obţinem varianta ecuaţională a rezultatului rezoluţiei $\theta(G' \cup H)$.

Corolar 3.7.1 *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 3.7.2 *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 $(\forall Y)\,\pi(v) \overset{\circ}{=}_b t$ if H^e o clauză din Γ^e astfel încât să se poată aplica narrowing-ul lui $P(s) \overset{\circ}{=}_b t$. 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

$$G^e \longrightarrow_n \theta(G'^e \cup \{t \overset{\circ}{=}_b t\} \cup H^e) = \theta(G' \cup H)^e \cup \{t \overset{\circ}{=}_b t\}.$$

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 $G_1 = F^e \cup \{t \overset{\circ}{=}_b t\}$. $\square$

Corolar 3.7.2 *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.

3.8 EXEMPLE

3.8.1 Primul

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
```

3.8.2 Al doilea

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 $w = s\ s\ y'$.

Chapter 4

INSTITUŢII

4.1 CATEGORII

Vom folosi cuvântul **clasă** pentru a desemna colecţiile arbitrare de elemente. Orice mulţime este o clasă. Reciproca este falsă, de exemplu clasa mulţimilor nu este o mulţime.

Categoria este unul dintre cele mai abstracte concepte create de algebră. Exemplele de categorii abundă în cele mai multe dintre ramurile matematicii. Definiţia categoriei este mai lungă şi o vom prezenta ca pe o poveste în mai multe episoade.

Definiţia 4.1.1 O **categorie** $\mathcal{C}$ este formată din **obiecte** şi **morfisme**. Ambele formează câte o clasă. Clasa obiectelor categoriei $\mathcal{C}$ se notează cu $Ob(\mathcal{C})$ sau $|\mathcal{C}|$.

Fiecare morfism f determină unic două obiecte:
1) Primul, sa-l notăm A, este numit domeniul sau sursa morfismului
2) Al doilea, sa-l notăm B, este numit codomeniul sau cosursa morfismului.
Pentru fiecare morfism f de sursă A şi cosursă B folosim una dintre notaţiile

$$f : A \longrightarrow B \quad \text{sau} \quad A \xrightarrow{f} B.$$

Pentru orice pereche de obiecte A şi B totalitatea morfismelor de sursă A şi cosursă B formează o mulţime notată $\mathcal{C}(A, B)$.

Morfismele se pot compune în anumite condiţii asemănătoare funcţiilor. Două morfisme f şi g se pot compune numai dacă codomeniul primului coincide cu domeniul celui de al doilea. Domeniul compunerii $f; g$ coincide cu domeniul primului morfism f şi codomeniul compunerii coincide cu codomeniul celui de al doilea morfism g aşa cum se vede mai jos

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[rr, bend right, "f;g"'] & B \arrow[r, "g"] & C \end{tikzcd} .$$

Compunerea este asociativă, adică oricare ar fi morfismele $f : A \longrightarrow B$, $g : B \longrightarrow C$ şi $h : C \longrightarrow D$ are loc egalitatea

$$(f; g); h = f; (g; h).$$

Pentru fiecare obiect A există un morfism distins $1_A : A \longrightarrow A$ care este numit morfismul **identitate** al lui A deoarece are efect neutru la compunere, adică

$1_A; f = f$ pentru orice morfism f de sursă A şi $g; 1_A = g$ penrtu orice morfism g de cosursă A. □

În orice categorie morfismele identitate sunt unice.

Vom continua cu mai multe exemple.

Sunt mai multe categorii în care obiectele sunt toate mulţimile.
1. Categoria relaţiilor notată **Rel** în care morfismele sunt relaţiile. Compunerea relaţiilor $P \in \mathbf{Rel}(A,B)$ şi $Q \in \mathbf{Rel}(B,C)$, adică $P \subseteq A \times B$ şi $Q \subseteq B \times C$ este prin definiţie

$$P;Q = \{(a,c) \in A \times C : (\exists b)(a,b) \in P \text{ şi } (b,c) \in Q\} \in \mathbf{Rel}(A,C).$$

2. Categoria funcţiilor partiale este notată **PFn**. Prin definiţie $F \in \mathbf{PFn}(A,B)$ dacă şi numai dacă $F \subseteq A \times B$ şi

$$(\forall a \in A)(\forall b \in B)(\forall c \in B)\{(a,b) \in F \text{ şi } (a,c) \in F \text{ implică } b = c\}.$$

Observaţia 4.1.2 *Compunerea ca relaţii a două funcţii parţiale este tot o funcţie parţială.*

Demonstraţie: Presupunem $F \in \mathbf{PFn}(A,B)$ şi $G \in \mathbf{PFn}(B,C)$ şi probăm că $F;G \in \mathbf{PFn}(A,C)$.

Fie $(a,c) \in F;G$ şi $(a,c') \in F;G$. Din definiţia compunerii există $b,b' \in B$ astfel încât $(a,b) \in F$, $(b,c) \in G$, $(a,b') \in F$ şi $(b',c') \in G$. Deoarece F este funcţie parţială deducem $b = b'$. Prin urmare $(b,c) \in G$ şi $(b,c') \in G$ deci $c = c'$ deoarece şi G este funcţie parţială. În concluzie $F;G$ este funcţie parţială. $\square$

3. Categoria mulţimilor notată **Set** are drept morfisme funcţiile, a căror compunere ca funcţii coincide cu a lor compunere ca relaţii.

Să continuăm cu algebra.
1. Categoria grupurilor are grupurile ca obiecte şi morfismele de grupuri drept morfisme.
2. Categoria grupurile abeliene sau comutative are grupurile abeliene ca obiecte şi morfismele de grupuri drept morfisme.
3. Categoria inelelor are inelele ca obiecte şi morfismele de inele drept morfisme.
4. Categoria Σ–algebrelor notată Alg_Σ.
5. Categoria Γ–algebrelor notată Alg_Γ.

Din topologie amintim categoria spaţiilor topologice în care obiectele sunt spaţie topologice iar morfismele sunt funcţiile continue.

Încheiem cu un exemplu în care morfismele nu mai sunt funcţii: categoria ataşată unei mulţimi preordonate $(P, \leq)$. Obiectele acetei categorii sunt chiar elementele mulţimii P. Date două elemente x, y din P mulţimea morfismelor de la x la y are cel mult un element. Există un morfism de la x la y dacă şi numai dacă $x \leq y$. Tranzitivitatea relaţiei $\leq$ ne asigură existenţa şi unicitatea compunerii morfismelor. Reflexivitatea relaţiei $\leq$ ne asigură existenţa morfismelor identitate.

4.1.1 Morfisme distinse

Definiţia 4.1.3 Un morfism $f : A \longrightarrow B$ se numeşte **izomorfism** dacă există un morfism $g : B \longrightarrow A$ cu proprietăţile $f;g = 1_A$ şi $g;f = 1_B$.

Observaţia 4.1.4 *Morfismul g din definiţia anterioară este unic determinat.*

Demonstraţie: Dacă $h : B \longrightarrow A$ are proprietăţile $f;h = 1_A$ şi $h;f = 1_B$, atunci

$$g = g;1_A = g;(f;h) = (g;f);h = 1_B;h = h.$$

Datorită unicităţii sale morfismul g de mai sus este numit **inversul lui** f şi este notat f^{-1}. $\square$

Propoziţie 4.1.5 *Dacă* $f : A \longrightarrow B$ *şi* $g : B \longrightarrow C$ *sunt izomorfisme, atunci* $f; g : A \longrightarrow C$ *este izomorfism. În plus*

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

Demonstraţie: Deoarece $f : A \longrightarrow B$ este izomorfism există morfismul $f^{-1} : B \longrightarrow A$ cu proprietăţile $f; f^{-1} = 1_A$ şi $f^{-1}; f = 1_B$. Deoarece $g : B \longrightarrow C$ este izomorfism există morfismul $g^{-1} : C \longrightarrow B$ cu proprietăţile $g; g^{-1} = 1_B$ şi $g^{-1}; g = 1_C$.

Egalităţile

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

şi

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

dovedesc că $f; g$ este izomorfism al cărui invers este $g^{-1}; f^{-1}$ ceeace probează egalitatea din enunţ. □

Definiţia 4.1.6 Un morfism $f : A \longrightarrow B$ se numeşte **monomorfism** sau **monic** dacă oricare ar fi morfismele $g : C \longrightarrow A$ şi $h : C \longrightarrow A$ egalitatea $g; f = h; f$ implică $g = h$.

Observaţia 4.1.7 *Orice izomorfism este monomorfism.*

Demonstraţie: Să presupunem că $f : A \longrightarrow B$ este izomorfism, prin urmare există morfismul $f^{-1} : B \longrightarrow A$ cu proprietăţile $f; f^{-1} = 1_A$ şi $f^{-1}; f = 1_B$.

Dacă morfismele $g : C \longrightarrow A$ şi $h : C \longrightarrow A$ verifică egalitatea $g; f = h; f$ prin compunere la dreapta cu f^{-1} deducem $g; f; f^{-1} = h; f; f^{-1}$ prin urmare $g; 1_A = h; 1_A$ deci $g = h$.

Propoziţie 4.1.8 *Compunerea monomorfismelor este un monomorfism.*

Demonstraţie: Să presupunem că morfismele $f : A \longrightarrow B$ şi $g : B \longrightarrow C$ sunt monomorfisme.

Dacă morfismele $u : D \longrightarrow A$ şi $v : D \longrightarrow A$ verifică egalitatea $u; (f; g) = v; (f; g)$ adică $(u; f); g = (v; f); g$, deoarece g este monomorfism rezultă că $u; f = v; f$, deci $u = v$ deoarece şi f este monomorfism.

Definiţia 4.1.9 Un morfism $f : A \longrightarrow B$ se numeşte **epimorfism** sau **epic** dacă oricare ar fi morfismele $g : B \longrightarrow C$ şi $h : B \longrightarrow C$ egalitatea $f; g = f; h$ implică $g = h$.

Observaţia 4.1.10 *Orice izomorfism este epimorfism.*

Propoziţie 4.1.11 *Compunerea epimorfismelor este un epimorfism.*

4.1.2 Dualitate

Fiecare categorie $\mathcal{C}$ are o categorie duală notată $\mathcal{C}^{op}$, definită în continuare.

Definiţia 4.1.12 Duala $\mathcal{C}^{op}$ are aceleaşi obiecte ca si $\mathcal{C}$. Morfismele işi permută sursa şi cosursa, adică prin definiţie oricare ar fi obiectele A şi B

$$\mathcal{C}^{op}(A, B) = \mathcal{C}(B, A).$$

Oricare ar fi morfismele $f \in \mathcal{C}^{op}(A, B)$ şi $g \in \mathcal{C}^{op}(B, C)$ compunerea lor este notată $f \circ g$ şi definită prin

$$f \circ g = g; f. \ \square$$

Corectitudinea acestei definiţii rezultă din urmatoarele:

$f : A \longrightarrow B$ şi $g : B \longrightarrow C$ sunt morfismele din $\mathcal{C}^{op}$;

în $\mathcal{C}$ morfismele devin $f : B \longrightarrow A$ şi $g : C \longrightarrow B$ şi se pot compune obţinând morfismul $g; f : C \longrightarrow A$;

care în $\mathcal{C}^{op}$ devine un morfism de la A la C.

Se poate verifica cu uşurinţă că $\mathcal{C}^{op}$ satisface condiţiile de a fi o categorie.

Observaţia 4.1.13 *Duala categoriei $\mathcal{C}^{op}$ este $\mathcal{C}$.*

Fiecare concept din teoria categoriilor are un dual. Pentru a obţine conceptul dual procedăm astfel:

Scriem definiţia conceptului în $\mathcal{C}^{op}$ şi apoi revenim în $\mathcal{C}$, obţinând conceptul dual.

Dualul dualului oricărui concept este conceptul iniţial.

Observaţia 4.1.14 *Noţiunile de monomorfism şi epimorfism sunt duale una alteia.*

Demonstraţie: Fie $f \in \mathcal{C}^{op}(A, B)$ un monomorfism, adică oricare ar fi morfismele $g \in \mathcal{C}^{op}(C, A)$ şi $h \in \mathcal{C}^{op}(C, A)$ egalitatea $g \circ f = h \circ f$ implică $g = h$.

Să ne mutăm în $\mathcal{C}$. Morfismul $f \in \mathcal{C}(B, A)$ este dualul unui monomorfism dacă oricare ar fi morfismele $g \in \mathcal{C}(A, C)$ şi $h \in \mathcal{C}(A, C)$ egalitatea $f; g = f; h$ implică $g = h$.

Deci dualul unui momomorfism este un epimorfism. $\Box$

Există şi concepte autoduale, adică dualul este chiar conceptul initial. Exemple de concepte autoduale sunt morfismele identitate şi izomorfismele.

Fiecare propoziţie P din teoria categoriilor are o propoziţie duală P^o, care se obţine din P înlocuind fiecare concept cu dualul sau.

Principiul dualităţi. Dacă o propoziţie privind teoria categoriilor este adevărată în orice categorie, atunci şi duala ei este adevărată în orice categorie.

Motivaţie: Presupunem că propoziţia P este adevărată în orice categorie şi arătăm că duala P^o este adevărată în orice categorie $\mathcal{C}$.

Deoarece P este adevărată în orice categorie, rezultă că P este adevărată în $\mathcal{C}^{op}$, prin urmare P^o este adevărată în categoria $\mathcal{C}$. $\Box$

Principiul dualităţii este util deoarece ne permite să facem mai puţine demonstraţii. De exemplu: propoziţiile "compunerea de monomorfisme este un monomorfism" şi "compunerea de epimorfisme este un epimorfism" sunt duale, prin urmare este suficent să demonstrăm numai una dintre ele deoarece a doua rezultă din prima folosind principiul dualităţii.

4.1.3 Subcategori

O categorie $\mathcal{D}$ se numeşte **subcategorie** a categoriei $\mathcal{C}$ dacă

1. Orice obiect din $\mathcal{D}$ este obiect al lui $\mathcal{C}$,
2. Oricere ar fi obiectele A şi B din $\mathcal{D}$, morfismele din $\mathcal{D}(A, B)$ sunt din $\mathcal{C}(A, B)$, adică $\mathcal{D}(A, B) \subseteq \mathcal{C}(A, B)$,
3. morfismele din $\mathcal{D}$ se compun ca în $\mathcal{C}$ şi
4. pentru orice obiect A din $\mathcal{D}$ morfismul 1_A din $\mathcal{C}(A, A)$ se găseşte în $\mathcal{D}(A, A)$.

PFn este subcategorie a lui **Rel**. **Set** este subcategorie a lui **PFn**. In aceste exemple obiectele subcategoriei coincid cu cele ale categoriei.

Subcategoria $\mathcal{D}$ a categoriei $\mathcal{C}$ se numeşte **plină** dacă $\mathcal{D}(A, B) = \mathcal{C}(A, B)$ oricare ar fi obiectele A şi B din $\mathcal{D}$. O subcategorie plină este caracterizată de obiectele ei, deoarece morifsmele sunt toate morfismele din categoria $\mathcal{C}$ cu sursa şi cosursa din $\mathcal{D}$. De exemplu subcategoria grupurilor comutative este subcategorie plină a categoriei grupurilor.

4.2 FUNCTORI

Categoriile, ca toate structurile algebrice, au si ele morfismele lor. Ele se numesc functori.

Definiţia 4.2.1 Fie două categorii $\mathcal{C}$ şi $\mathcal{D}$. Un functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ de la categoria $\mathcal{C}$ la categoria $\mathcal{D}$

1. asociază fiecărui obiect C din categoria $\mathcal{C}$ un alt obiect $F(C)$ din categoria $\mathcal{D}$
2. asociază fiecărui morfism $f : A \longrightarrow B$ din categoria $\mathcal{C}$ un alt morfism $F(f) : F(A) \longrightarrow F(B)$ din categoria $\mathcal{D}$ astfel încât
   (a) $F(f; g) = F(f); F(g)$ oricare ar fi morfismele $f : A \longrightarrow B$ şi $g : B \longrightarrow C$ din $\mathcal{C}$,
   (b) $F(1_A) = 1_{F(A)}$ pentru orice obiec A din $\mathcal{C}$. $\Box$

Fiecare categorie $\mathcal{C}$ posedă un functor identitate $1_\mathcal{C} : \mathcal{C} \longrightarrow \mathcal{C}$ definit prin

1. $1_\mathcal{C}(A) = A$ pentru orice obiect A din $\mathcal{C}$ şi,
2. $1_\mathcal{C}(f) = f$ pentru orice morfism f din $\mathcal{C}$.

Ca toate morfismele functorii se pot compune dacă cosursa primului coincide cu sursa celui de al doilea.

Definiţia 4.2.2 Compunerea $F; G : \mathcal{A} \longrightarrow \mathcal{C}$ a functorilor $F : \mathcal{A} \longrightarrow \mathcal{B}$ şi $G : \mathcal{B} \longrightarrow \mathcal{C}$ este definită prin

1. $(F; G)(A) = G(F(A))$ pentru orice obiect A din $\mathcal{C}$,
2. $(F; G)(f) = G(F(f))$ pentru orice morfism f din $\mathcal{C}$. $\Box$

Probăm că $F; G$ este functor.

1. $(F; G)(f; g) = G(F(f; g)) = G(F(f); F(g)) = G(F(f)); G(F(g)) = (F; G)(f); (F; G)(g)$ unde $f : A \longrightarrow B$ şi $g : B \longrightarrow C$ sunt morfisme din $\mathcal{C}$,
2. $(F; G)(1_A) = G(F(1_A)) = G(1_{F(A)}) = 1_{G(F(A))} = 1_{(F;G)(A)}$ pentru orice obiect A din $\mathcal{C}$.

Compunerea functorilor este asociativă. Fie $F : \mathcal{A} \longrightarrow \mathcal{B}$, $G : \mathcal{B} \longrightarrow \mathcal{C}$ şi $H : \mathcal{C} \longrightarrow \mathcal{D}$. Pentru orice obiect A din $\mathcal{A}$

$$((F; G); H)(A) = H((F; G)(A)) = H(G(F(A))) = (G; H)(F(A)) = (F; (G; H))(A).$$

Pentru orice morfism f din $\mathcal{A}$

$$((F; G); H)(f) = H((F; G)(f)) = H(G(F(f))) = (G; H)(F(f)) = (F; (G; H))(f).$$

În concluzie $(F; G); H) = F; (G; H)$.

Dacă $F : \mathcal{A} \longrightarrow \mathcal{B}$ este un functor, atunci $1_\mathcal{A}; F = F$ şi $F; 1_\mathcal{B} = F$. Prin urmare categoriile formează o categorie "mare" deoarece totalitatea functorilor dintre două categorii fixate s-ar putea sa nu fie o mulţime. Vom nota această categorie mare cu Cat.

Propoziţie 4.2.3 *Dacă $F : \mathcal{C} \longrightarrow \mathcal{D}$ este un functor şi $f : A \longrightarrow B$ un izomorfism, atunci $F(f)$ este un izomorfism.*

Demonstraţie: Deoarece $f : A \longrightarrow B$ un izomorfism există morfismul $f^{-1} : B \longrightarrow A$ cu proprietăţile $f; f^{-1} = 1_A$ şi $f^{-1}; f = 1_B$. Aplicând functorul functorul F obţinem

$$F(f; f^{-1}) = F(1_A) \text{ şi } F(f^{-1}; f) = F(1_B),$$

prin urmare

$$F(f); F(f^{-1}) = 1_{F(A)} \text{ şi } F(f^{-1}); F(f) = 1_{F(B)},$$

deci $F(f)$ este izomorfism având inversul $F(f^{-1})$ adică

$$F(f)^{-1} = F(f^{-1}). \ \Box$$

Fie $\mathcal{S}$ şi $\mathcal{C}$ două categorii şi $M : \mathcal{S} \longrightarrow \mathcal{C}^{op}$ un functor. Dacă $f : A \longrightarrow B$ şi $g : B \longrightarrow C$ sunt două morfisme din $\mathcal{S}$, atunci

$$M(f; g) = M(g); M(f).$$

Întradevăr $M(f; g) = M(f) \circ M(g) = M(g); M(f)$. Un astfel de functor se mai numeşte contravariant.

4.3 CONEXIUNI GALOIS

O **conexiune Galois**

$$(P, \leq) \underset{*}{\overset{*}{\rightleftarrows}} (Q, \leq)$$

este formată din două mulţimi parţial ordonate $(P, \leq)$ şi $(Q, \leq)$ şi două funcţii * (deşi sunt notate la fel deosebirea se va face din context) cu următoarele proprietăţi:

1. $(\forall p, p_1 \in P)\, p \leq p_1 \Longrightarrow p_1^* \leq p^*$,
2. $(\forall q, q_1 \in Q)\, q \leq q_1 \Longrightarrow q_1^* \leq q^*$,
3. $(\forall p \in P)\, p \leq p^{**}$,
4. $(\forall q \in Q)\, q \leq q^{**}$.

Primele două cerinţe ne spun că funcţiile * sunt descrescătoare.
Probăm unele consecinnţe ale axiomelor de mai sus

1. $*\,*\,* = *$.
Din (4) rezultă $p^* \leq p^{***}$. Din (3), $p \leq p^{**}$, şi cu (1) obţinem $p^{***} \leq p^*$. Deci $p^{***} = p^*$.
2. $\bullet = **$ este operator de închidere.
Din (3) avem $p \leq p^\bullet$. Cum $p^* = p^{***}$ rezultă $p^{**} = p^{****}$ deci $p^\bullet = p^{\bullet\bullet}$. Dacă $p \leq q$, aplicând (1) şi (2) avem $q^* \leq p^*$ şi $p^{**} \leq q^{**}$ deci $p^\bullet \leq q^\bullet$.

Exerciţiu Inlocuiţi una dintre relaţiile de ordine cu duala ei şi arătaţi că noţiunea de conexiune Galois este echivalentă cu o pereche de aplicaţii $F : Q \longrightarrow P$ şi $G : P \longrightarrow Q$ cu proprietatea

$$(\forall p \in P)(\forall q \in Q) F(q) \leq p \Leftrightarrow q \leq G(p).$$

Conceptul de conexiune Galois mai este echivalent şi cu cel de functori adjuncţi între categoriile ataşate unor mulţimi partial ordonate.

4.3.1 Conexiunea Galois ataşată unei relaţii

Fie A şi B două mulţimi şi $R \subseteq A \times B$ o relaţie. Relaţiei R îi ataşăm conexiunea Galois

$$(\mathcal{P}(A), \subseteq) \underset{*}{\overset{*}{\rightleftarrows}} (\mathcal{P}(B), \subseteq)$$

unde $(\mathcal{P}(X), \subseteq)$ reprezintă mulţimea părţilor mulţimii X ($X \in \{A, B\}$) ordonată (parţial) cu incluziunea iar funcţiile $*$ sunt definite în modul următor:

$C^* = \{b \in B \mid \forall c \in C, cRb\}$ pentru orice $C \subseteq A$,

$D^* = \{a \in A \mid \forall d \in D, aRd\}$ pentru orice $D \subseteq B$.

Se verifică uşor cele patru proprietăţi din definiţia conexiunii Galois. Probăm numai prima şi a treia condiţie

Fie $C \subseteq C_1 \subseteq A$. Probăm $C_1^* \subseteq C^*$.

Fie $b \in C_1^*$. Prin definiţie bRc pentru orice $c \in C_1$. Deoarece $C \subseteq C_1$ deducem că bRc pentru orice $c \in C$. Deci $b \in C^*$.

Fie $C \subseteq A$. Probăm că $C \subseteq C^{**}$.

Fie $c \in C$. Din definiţia lui C^* deducem că pentru orice $b \in C^*$ are loc cRb. Deoarece c este în relaţia R cu orice element din C^* deducem că $c \in C^{**}$.

Propoziţie 4.3.1 *Conexiunea Galois ataşată unei relaţii satisface*

1. $(\bigcup_{i\in I} C_i)^* = \bigcap_{i\in I} C_i^*$ *unde* $C_i \subseteq A$ *pentru orice* $i \in I$,

2. $(\bigcup_{i\in I} D_i)^* = \bigcap_{i\in I} D_i^*$ *unde* $D_i \subseteq B$ *pentru orice* $i \in I$.

Demonstraţie: Probăm numai prima egalitate. Pentru orice $b \in B$ remarcăm echivalenţa următoarelor observaţii

$b \in (\bigcup_{i\in I} C_i)^*$,

bRc pentru orice $c \in \bigcup_{i\in I} C_i$,

bRc pentru orice $i \in I$ şi $c \in C_i$,

bRc pentru orice $c \in C_i$ şi $i \in I$,

$b \in C_i^*$ pentru orice $i \in I$ şi

$b \in \bigcap_{i\in I} C_i^*$.

4.4 INSTITUŢII

Se numeşte *instituţie* (notată cu $\mathcal{I}$) următoarea construcţie

$$\begin{tikzcd} & Set \\ Sign \arrow[ur, "Sen"] \arrow[dr, "Mod"'] & \\ & Cat^{op} \end{tikzcd}$$

unde $Sign$ este o categorie, Set este categoria mulţimilor, Cat^{op} este duala categoriei categoriilor iar Sen şi Mod sunt doi functori. Obiectele categoriei $Sign$ se numesc *signaturi*. $Sign$ este numită categoria signaturilor. De asemenea, se mai dă câte o **relaţie de satisfacere** pentru fiecare signatură (obiect din categoria signaturilor),

$$\models_\Sigma \subseteq |Mod(\Sigma)| \times Sen(\Sigma) \text{ pentru orice signatură } \Sigma \in |Sign|$$

care verifică următoarea **condiţie de satisfacere**:

$$\forall\varphi : \Sigma \longrightarrow \Sigma', \forall M' \in |Mod(\Sigma')|, \forall e \in Sen(\Sigma),$$

$$M' \models_{\Sigma'} Sen(\varphi)(e) \Longleftrightarrow Mod(\varphi)(M') \models_{\Sigma} e. \ \square$$

Se numeşte Σ-model orice obiect $M \in |Mod(\Sigma)|$. Se numeşte Σ-propoziţie orice element $e \in Sen(\Sigma)$.

Pentru orice signatură Σ, relaţiei $\models_\Sigma$ îi putem asocia o conexiune Galois ca mai înainte

$$(\mathcal{P}(|Mod(\Sigma)|), \subseteq) \underset{*}{\overset{*}{\rightleftarrows}} (\mathcal{P}(Sen(\Sigma)), \subseteq).$$

Definiţia 4.4.1 Spunem că Σ-modelul M satisface mulţimea de Σ-propoziţii $E \subseteq Sen(\Sigma)$, fapt notat prin $M \models_\Sigma E$ dacă $M \models_\Sigma e$ pentru orice $e \in E$.

Observaţia 4.4.2 *$M \models_\Sigma E$ dacă şi numai dacă $M \in E^*$.*

Observăm că E^* este mulţimea modelelor lui E.

Observaţia 4.4.3 *Dacă $M \models_\Sigma E$ atunci $M \models_\Sigma E^\bullet$.*

Demonstraţie: Într-adevăr, dacă $M \models_\Sigma E$ atunci $M \in E^* = E^{***} = E^{\bullet *}$ adică $M \models_\Sigma E^\bullet$. $\square$

Are loc şi **condiţia de satisfacere extinsă:**
Pentru orice morfism de signaturi $\varphi : \Sigma \longrightarrow \Sigma'$, pentru orice $M' \in |Mod(\Sigma')|$ şi pentru orice $E \subseteq Sen(\Sigma)$

$$M' \models_{\Sigma'} Sen(\varphi)(E) \Longleftrightarrow Mod(\varphi)(M') \models_\Sigma E.$$

Membrul stâng este echivalent cu
$M' \models_{\Sigma'} Sen(\varphi)(e)$ pentru orice $e \in E$ care conform condiţiei de satisfacere este echivalent cu $Mod(\varphi)(M') \models_\Sigma e$ pentru orice $e \in E$, adică cu membrul drept.

4.4.1 Consecinţă semantică

Definiţia 4.4.4 Spunem că mulţimea de Σ-propoziţii E implică Σ-propoziţia $e \in Sen(\Sigma)$, sau că e este o consecinţă semantică a lui E şi notăm $E \models_\Sigma e$ dacă

$$(\forall M \in |Mod(\Sigma)|) M \models_\Sigma E \Longrightarrow M \models_\Sigma e. \square$$

Observaţia 4.4.5 *$E \models_\Sigma e$ dacă şi numai dacă $e \in E^\bullet$.*

Demonstraţie: Din definiţie $E \models_\Sigma e$ este echivalent cu $(\forall M) M \in E^* \Longrightarrow M \models_\Sigma e$, adică cu $e \in E^{**}$, deci cu $e \in E^\bullet$.

Mulţimea $E^\bullet$ se numeşte **mulţimea consecinţelor semantice** ale mulţimii E.

Definiţia 4.4.6 Oricare ar fi mulţimile de Σ-propoziţii E şi E_1 spunem că E_1 este o consecinţă semantică a lui E şi notăm $E \models_\Sigma E_1$ dacă

$$(\forall M \in |Mod(\Sigma)|) M \models_\Sigma E \Longrightarrow M \models_\Sigma E_1. \square$$

Observaţia 4.4.7 *$E \models_\Sigma E_1$ dacă şi numai dacă $E^* \subseteq E_1^*$, dacă şi numai dacă $E_1 \subseteq E^\bullet$*

Lemă 4.4.1 *Pentru orice morfism de signaturi* $\varphi : \Sigma \longrightarrow \Sigma'$ *şi pentru orice* $E \subseteq Sen(\Sigma)$ *avem*

$$Sen(\varphi)(E)^* = Mod(\varphi)^{-1}(E^*).$$

Demonstraţie: Din condiţia de satisfacere extinsă, pentru orice $M' \in |Mod(\Sigma')|$,

$$M' \models_{\Sigma'} Sen(\varphi)(E) \Longleftrightarrow Mod(\varphi)(M') \models_{\Sigma} E$$

de unde $M' \in Sen(\varphi)(E)^* \Longleftrightarrow Mod(\varphi)(M') \in E^*$ deci

$$M' \in Sen(\varphi)(E)^* \Longleftrightarrow M' \in Mod(\varphi)^{-1}(E^*)$$

ceea ce demonstrează egalitatea din enunţ. $\square$

Observaţia 4.4.8 (Lema închiderii) *Pentru orice* $\varphi : \Sigma \longrightarrow \Sigma'$ *morfism de signaturi şi orice* $E \subseteq Sen(\Sigma)$ *avem*

$$Sen(\varphi)(E^\bullet) \subseteq Sen(\varphi)(E)^\bullet.$$

Demonstraţie: Plecând de la

$$Sen(\varphi)(E^\bullet) \subseteq Sen(\varphi)(E^\bullet)^\bullet = Sen(\varphi)(E^\bullet)^{**}$$

şi folosind observaţia anterioară

$$Sen(\varphi)(E^\bullet)^{**} = (Mod(\varphi)^{-1}(E^{\bullet *}))^* = (Mod(\varphi)^{-1}(E^*))^* = (Sen(\varphi)(E))^{**} = (Sen(\varphi)(E))^\bullet$$

obţinem concluzia. $\square$

Observaţia 4.4.9 *Fie* $\varphi : \Sigma \longrightarrow \Sigma'$ *un morfism de signaturi* $e \in Sen(\Sigma)$ *şi* $E \subseteq Sen(\Sigma)$.

1. $E \models_{\Sigma} e$ *implică* $Sen(\varphi)(E) \models_{\Sigma'} Sen(\varphi)(e)$,
2. *Dacă* $Mod(\varphi) : Mod(\Sigma') \longrightarrow Mod(\Sigma)$ *este surjectiv pe obiecte, atunci* $E \models_{\Sigma} e$ *dacă şi numai dacă* $Sen(\varphi)(E) \models_{\Sigma'} Sen(\varphi)(e)$.

Demonstraţie:

1. Semnificaţia lemei închiderii este aceea că pentru orice $e \in E^\bullet, Sen(\varphi)(e) \in Sen(\varphi)(E)^\bullet$ adică dacă $E \models_{\Sigma} e$ atunci $Sen(\varphi)(E) \models_{\Sigma'} Sen(\varphi)(e)$ ceea ce, în cuvinte, înseamnă că dacă aplicăm unei deductibilităţi semantice un morfism de signaturi obţinem tot o deductibilitate semantică.

2. Pentru demonstrarea incluziunii contrare folosim ipoteza $Sen(\varphi)(E) \models_{\Sigma'} Sen(\varphi)(e)$.

Fie $M \models_{\Sigma} E$. Deoarece $Mod(\varphi) : Mod(\Sigma') \longrightarrow Mod(\Sigma)$ este surjectiv pe obiecte există $M' \in |Mod(\Sigma')|$ astfel încât $M = Mod(\varphi)(M')$, prin urmare $Mod(\varphi)(M') \models_{\Sigma} E$. Din condiţia de satisfacere deducem $M' \models_{\Sigma'} Sen(\varphi)(E)$, prin urmare din ipoteză $M' \models_{\Sigma'} Sen(\varphi)(e)$. Folosind din nou condiţia de satisfacere $Mod(\varphi)(M') \models_{\Sigma} e$, deci $M \models_{\Sigma} e$. $\square$

Propoziţie 4.4.10 *Dacă* $\varphi : \Sigma \longrightarrow \Sigma'$ *este morfism de signaturi iar* $Mod(\varphi) : Mod(\Sigma') \longrightarrow Mod(\Sigma)$ *este surjectiv pe obiecte, atunci pentru orice* $E \subseteq Sen(\Sigma)$ *avem* $Mod(\varphi)(Sen(\varphi)(E)^*) = E^*$.

Demonstraţie: Facem observaţia că pentru o funcţie surjectivă $f : A \longrightarrow B$, pentru orice $C \subseteq B$ are loc egalitatea $f(f^{-1}(C)) = C$. Cum din lema 4.4.1 rezultă egalitatea

$$Sen(\varphi)(E)^* = Mod(\varphi)^{-1}(E^*)$$

obţinem, ţinând seamă de observaţia anterioară, $Mod(\varphi)(Sen(\varphi)(E)^*) = Mod(\varphi)(Mod(\varphi)^{-1}(E^*)) = E^*$.

4.4.2 Exemple

Un exemplu din algebră

Vom descrie succint un exemplu simplu şi neartificial de instituţie. In acest exemplu X este o mulţime fixată de variabile.

Drept categorie a signaturilor luăm categoria inelelor comutative $CRing$. Functorul

$$Sen : CRing \longrightarrow Set$$

asociază fiecărui inel R, inelul de polinoame cu coeficienţi în R şi variabile din X, adică $Sen(R) = R[X]$. In plus orice morfism de inele $g : R \to R'$ se prelungeşte în mod natural la un morfism de inele $Sen(g) : R[X] \longrightarrow R'[X]$ prin $Sen(g)(x) = x$ pentru orice $x \in X$.

Functorul

$$Mod : CRing \longrightarrow Cat^{op}$$

asociază fiecărui inel comutativ R categoria R-algebrelor notată $Mod(R)$. O R-algebră este o pereche (a, I) unde $a : R \to I$ este un morfism de inele.

Pentru orice morfism de inele $\varphi : R' \to R$ functorul

$$Mod(\varphi) : Mod(R) \to Mod(R')$$

este definit prin:

$Mod(\varphi)(a, I) = (\varphi; a, I)$ pentru orice R-algebră (a, I) şi

$Mod(\varphi)(m) = m$ pentru orice morfism m de R-algebre.

Relaţia de satisfacere aşa cum este definită mai jos reflectă ideea că întro R–algebră polinomul are rădăcini.

Pentru orice R-algebră (a, I) şi orice funcţie $f : X \to I$ notăm cu $< a, f >: R[X] \longrightarrow I$ unicul morfism de inele a cărui restricţie la R este a şi a cărui restricţie la X este f. Pentru orice R-algebră (a, I) şi orice polinom $p \in R[X]$, relaţia de satisfacere este definită prin:

$$(a, I) \models_R p \Longleftrightarrow (\exists s : X \to I) < a, s > (p) = 0.$$

Pentru orice morfism de signaturi, adică de inele comutative $\varphi : R' \to R$, pentru orice R-algebră (a, I) şi pentru orice funcţie $s : X \to I$ observăm că

$$Sen(\varphi); < a, s >=< \varphi; a, s > .$$

Verificăm condiţia de satisfacere. Fie $\varphi : R' \to R$ un morfism de inele comutative, (a, I) o R-algebră şi $p \in R'[X]$. Observăm că afirmaţia

$$(a, I) \models_R Sen(\varphi)(p)$$

este succesiv echivalentă cu

$$\begin{gathered}(\exists s : X \to I) < a, s > (Sen(\varphi)(p)) = 0,\\ (\exists s : X \to I) < \varphi; a, s > (p) = 0,\\ (\varphi; a, I) \models_{R'} p,\\ Mod(\varphi)(a, I) \models_{R'} p\end{gathered}$$

deci condiţia de satisfacere este verificată.

Instituţia calculului propoziţional

Fie X o mulţime de variabile propoziţionale şi Σ o signatură algebrică cu un singur sort, cu simbolurile de funcţii

$$\neg : \mathit{bool} \to \mathit{bool} \text{ şi } \wedge : \mathit{bool}\,\mathit{bool} \to \mathit{bool}.$$

Considerăm $\mathcal{T}_\Sigma(X)$, Σ-algebra expresiilor booleene cu variabile din X. Menţionăm că textul următor este independent de signatura aleasă pentru a defini expresiile calculului propoziţional.

Pentru instituţia calculului propoziţional categoria signaturilor este dată de **Set**, categoria mulţimilor. O mulţime este o signatură deoarece în definiţia următoare se consideră a fi mulţimea variabilelor propoziţionale folosite în construcţia expresiilor booleene.

Functorul Sen

Functorul $Sen : \mathbf{Set} \to \mathbf{Set}$ se defineşte pe obiecte prin $Sen(X) = T_\Sigma(X)$. Astfel se asociază fiecărei mulţimi de variabile propoziţionale X, mulţimea suport a Σ-algebrei $\mathcal{T}_\Sigma(X)$ a expresiilor booleene cu variabile din X formate cu ajutorul conectorilor logici daţi de signatura Σ.

Fie $f : X \to Y$ o funcţie şi $i_X : X \to T_\Sigma(X)$, $i_Y : Y \to T_\Sigma(Y)$ incluziuni. Atunci conform proprietăţii de universalitate a Σ-algebrei liber generate există un unic Σ-morfism $Sen(f) : \mathcal{T}_\Sigma(X) \to \mathcal{T}_\Sigma(Y)$ cu proprietatea $i_X; Sen(f) = f; i_Y$, fapt reflectat de comutativitatea diagramei de mai jos.

$$\begin{tikzcd}
X \arrow[r, "i_X"] \arrow[d, "f"'] & T_\Sigma(X) \arrow[d, dotted, "\exists! Sen(f)"] \\
Y \arrow[r, "i_Y"] & T_\Sigma(Y)
\end{tikzcd}$$

Arătăm că Sen astfel definit este functor.

Mai intâi legea compunerii. Fie $f : X \to Y$ şi $g : Y \to Z$ două funcţii. Atunci conform diagramei de mai jos şi a modului în care a fost definit Sen rezultă că

$$i_X; Sen(f); Sen(g) = f; i_Y; Sen(g) = f; g; i_Z$$

Din proprietatea de universalitate a Σ-algebrei $\mathcal{T}_\Sigma(X)$, partea cerând unicitatea, rezultă că $Sen(f; g) = Sen(f); Sen(g)$.

$$\begin{tikzcd}
X \arrow[r, "i_X"] \arrow[d, "f"'] & T_\Sigma(X) \arrow[d, dotted, "\exists! Sen(f)"] \arrow[dd, dotted, bend left=60, "\exists! Sen(f;g)"] \\
Y \arrow[r, "i_Y"] \arrow[d, "g"'] & T_\Sigma(Y) \arrow[d, dotted, "\exists! Sen(g)"] \\
Z \arrow[r, "i_Z"] & T_\Sigma(Z)
\end{tikzcd}$$

Pentru o mulţime de variabile X şi funcţia identitate $1_X : X \to X$ observăm că $Sen(1_X) = 1_{T_\Sigma(X)} = 1_{Sen(X)}$

Functorul Mod

Vom nota cu $\mathbf{B}_2$ algebra booleană cu două elemente care în cele ce urmează este privită ca o Σ-algebră.

Pentru o mulţime de variabile X definim $Mod(X)$ ca fiind categoria discretă (în care toate morfismele sunt identităţi) ale cărei obiecte sunt Σ-morfismele $h \in Alg_\Sigma(\mathcal{T}_\Sigma(X), \mathbf{B}_2)$.

Menţionăm că un functor între două categorii provenind din mulţimi ca mai sus este pur şi simplu o funcţie. Pentru a defini Mod pentru funcţii considerăm $f : X \to Y$ o funcţie. Atunci $Mod(f) : Mod(Y) \to Mod(X)$ sau

$$Mod(f) : Alg_\Sigma(\mathcal{T}_\Sigma(Y), \mathbf{B}_2) \to Alg_\Sigma(\mathcal{T}_\Sigma(X), \mathbf{B}_2).$$

Definim $Mod(f)$ prin $Mod(f)(h) = Sen(f); h$ pentru orice morfism $h : \mathcal{T}_\Sigma(Y) \to \mathbf{B}_2$ iar pentru morfisme $Mod(f)$ duce identităţile în identităţi. Din construcţie rezultă imediat că pentru orice funcţie $f : X \to Y$, $Mod(f)$ este functor.

Arătăm că Mod este functor.

Pentru compunere considerăm $X \xrightarrow{f} Y \xrightarrow{g} Z$ două funcţii şi imaginile lor prin Mod

$$Alg_\Sigma(\mathcal{T}_\Sigma(Z), \mathbf{B}_2) \xrightarrow{Mod(g)} Alg_\Sigma(\mathcal{T}_\Sigma(Y), \mathbf{B}_2) \xrightarrow{Mod(f)} Alg_\Sigma(\mathcal{T}_\Sigma(X), \mathbf{B}_2)$$

Verificarea legii compunerii rezultă din şirul de egalităţi

$$\begin{aligned} Mod(f;g)(h) &= Sen(f;g); h \\ &= Sen(f); Sen(g); h \\ &= Sen(f); Mod(g)(h) \\ &= Mod(f)(Mod(g)(h)) \\ &= (Mod(g); Mod(f))(h) \end{aligned}$$

Deci $Mod(f;g) = Mod(g); Mod(f)$.

Pentru identităţi avem $Mod(1_X)(h) = Sen(1_X); h = 1_{\mathcal{T}_\Sigma(X)}; h = h$ pentru orice $h \in Alg_\Sigma(\mathcal{T}_\Sigma(X), \mathbf{B}_2)$, deci $Mod(1_X) = 1_{Alg_\Sigma(\mathcal{T}_\Sigma(X), \mathbf{B}_2)}$.

Relaţia de satisfacere

Fie X o mulţime de variabile, $e \in Sen(X)$ şi $h \in Mod(X) = Alg_\Sigma(\mathcal{T}_\Sigma(X), \mathbf{B}_2)$. Intuitiv, morfismul $h : \mathcal{T}_\Sigma(X) \longrightarrow \mathbf{B}_2$ care dă valori de adevăr variabilelor din X, satisface expresia e a calculului propoziţional dacă rezultatul evaluării expresiei e pentru valorile de adevăr ale variabilelor din X este 1, adică adevărul. Definim relaţia de satisfacere $\models_X$ prin

$$h \models_X e \Leftrightarrow h(e) = 1$$

Verificăm condiţia de satisfacere. Fie $f : X \to Y$ o funcţie, $e \in Sen(X)$, $h' \in Mod(Y)$. Atunci

$$\begin{aligned} Mod(f)(h') \models_X e &\Leftrightarrow Mod(f)(h')(e) = 1 \\ &\Leftrightarrow (Sen(f); h')(e) = 1 \\ &\Leftrightarrow h'(Sen(f)(e)) = 1 \\ &\Leftrightarrow h' \models_Y Sen(f)(e) \end{aligned}$$

4.5 PREZENTĂRI

Observaţia 4.5.1 *Fie $f : A \longrightarrow B$ o funcţie.*

1. *$f(f^{-1}(D)) \subseteq D$ pentru orice $D \subseteq B$.*

2. *Dacă $C \subseteq A$ şi $D \subseteq B$ atunci, $f(C) \subseteq D$ dacă şi numai dacă $C \subseteq f^{-1}(D)$.*

Propoziţie 4.5.2 *Fie $\varphi : \Sigma \longrightarrow \Sigma_1$ un morfism de signaturi, $E \subseteq Sen(\Sigma)$ şi $E_1 \subseteq Sen(\Sigma_1)$. Următoarele afirmaţii sunt echivalente:*

1. *$Sen(\varphi)(E) \subseteq E_1^\bullet$,*
2. *$E_1 \models_\Sigma Sen(\varphi)(E)$.*
3. *pentru orice $M_1 \in |Mod(\Sigma_1)|, M_1 \models_{\Sigma_1} E_1 \Longrightarrow Mod(\varphi)(M_1) \models_\Sigma E$,*

Demonstraţie: Observaţia 4.4.7 dovedeşte echivalenţa primelor două afirmaţii, dar şi echivalenţa lor cu

$$E_1^* \subseteq Sen(\varphi)(E)^*.$$

Folosind lema 4.4.1 ultima incluziune este echivalentă cu

$$E_1^* \subseteq Mod(\varphi)^{-1}(E^*).$$

Prin urmare conform proprietăţii 2 din observaţia 4.5.1 este echivalentă cu

$$Mod(\varphi)(E_1^*) \subseteq E^*$$

adică pentru orice $M_1 \in |Mod(\Sigma_1)|, M_1 \in E_1^*$ implică $Mod(\varphi)(M_1) \in E^*$ ceea ce, evident, este a treia afirmaţie. □

Definiţia 4.5.3 Se numeşte **prezentare** o pereche (Σ, E) unde $\Sigma \in |Sign|$, şi $E \subseteq Sen(\Sigma)$.

Un morfism de signaturi $\varphi : \Sigma \longrightarrow \Sigma'$ se numeşte **morfism de prezentări** $\varphi : (\Sigma, E) \longrightarrow (\Sigma', E')$ dacă satisface condiţiile echivalente ale propoziţiei precedente.

Observaţia 4.5.4 *Dacă $\varphi : (\Sigma, E) \longrightarrow (\Sigma', E')$ este morfism de prezentări, atunci $\varphi : (\Sigma, E^\bullet) \longrightarrow (\Sigma', E')$ este tot morfism de prezentări.*

Demonstraţie: Din ipoteză rezultă că $Sen(\varphi)(E) \subseteq E'^\bullet$ de unde $Sen(\varphi)(E)^\bullet \subseteq E'^\bullet$ şi, folosind lema închiderii, obţinem $Sen(\varphi)(E^\bullet) \subseteq Sen(\varphi)(E)^\bullet \subseteq E'^\bullet$ ceea ce demonstrează concluzia. □

Propoziţie 4.5.5 *Compunerea ca morfisme de signaturi a două morfisme de prezentări este tot un morfism de prezentări.*

Demonstraţie: *Fie $(\Sigma, E) \xrightarrow{\varphi} (\Sigma', E') \xrightarrow{\psi} (\Sigma'', E'')$ două morfisme de prezentări. Observăm că*

$$Sen(\varphi\psi)(E) = (Sen(\varphi)Sen(\psi))(E) = Sen(\psi)(Sen(\varphi)(E)) \subseteq Sen(\psi)(E'^\bullet) \subseteq E''^\bullet$$

ultima incluziune datorându-se observaţiei anterioare. □

4.5.1 Instituţia prezentărilor

Din propoziţia anterioară obţinem că mulţimea prezentărilor unei instituţii $\mathcal{I}$ împreună cu morfismele de prezentări formează o categorie, pe care o vom nota cu $Pres(\mathcal{I})$.

Definim instituţia prezentărilor $\mathcal{I}^p$ unei instituţii $\mathcal{I}$. Categoria prezentărilor lui $\mathcal{I}$ devine categoria signaturilor lui $\mathcal{I}^p$

$$Sign^p = Pres(\mathcal{I}).$$

Notăm cu $V : Pres(\mathcal{I}) \longrightarrow Sign$ functorul uituc, definit prin $V(\Sigma, E) = \Sigma$ pentru orice prezentare (Σ, E) şi $V(\phi) = \phi$ pentru orice morfism de prezentări ϕ. Prin definiţie

$$Sen^p = V; Sen.$$

Folosind propoziţia 4.5.2 putem defini corespondentul lui Mod pentru categoria prezentărilor şi anume

$$Mod^p : Pres(\mathcal{I}) \longrightarrow Cat^{op}.$$

Prin definiţie $Mod^p(\Sigma, E)$ reprezintă subcategoria plină dată de toate modelele ce satisfac E. Pentru orice morfism de prezentări $\varphi : (\Sigma, E) \longrightarrow (\Sigma', E')$ considerăm diagrama

$$
\begin{tikzcd}
Mod(\Sigma') \arrow[r, "Mod(\varphi)"] & Mod(\Sigma) \\
Mod^p(\Sigma', E') \arrow[u, hook] \arrow[r, "Mod^p(\varphi)"] & Mod^p(\Sigma, E) \arrow[u, hook]
\end{tikzcd}
$$

unde prin definiţie $Mod^p(\varphi)$ este restricţia şi co-restricţia lui $Mod(\varphi)$.

Este suficient să mai definim relaţia de satisfacere şi să demonstrăm condiţia de satisfacere. Prin definiţie

$$M \models_{(\Sigma,E)} e \Longleftrightarrow M \models_{\Sigma} e$$

pentru orice (Σ, E)-model M şi orice $e \in Sen(\Sigma)$.

Demonstăm condiţia de satisfacere. Fie $\varphi : (\Sigma, E) \longrightarrow (\Sigma', E')$ un morfism de prezentări, M' un (Σ', E') model şi $e \in Sen(\Sigma)$.

$$Mod^p(\varphi)(M') \models_{(\Sigma,E)} e$$

adică $Mod(\varphi)(M') \models_{\Sigma} e$ este echivalentă conform condiţiei de satisfacere din $\mathcal{I}$ cu $M' \models_{\Sigma} Sen(\varphi)(e)$, deci echivalentă cu

$$M' \models_{(\Sigma,E)} Sen^p(\varphi)(e).$$

4.6 TEORII

Se numeşte **teorie** o prezentare (Σ, E) cu $E = E^{\bullet}$. Deoarece $\bullet$ este operator de închidere rezultă că teoriile formează pentru fiecare signatură câte o familie Moore în mulţimea parţial ordonată prin incluziune a mulţimii părţilor lui $Sen(\Sigma)$. Propoziţia următoare este un caz particular al propoziţiei 1.3.8.

Propoziţie 4.6.1 *Orice intersecţie de teorii este o teorie.*

Propoziţie 4.6.2 *Dacă $\varphi : \Sigma \longrightarrow \Sigma'$ este morfism de signaturi iar (Σ', E') este o teorie atunci şi $(\Sigma, Sen(\varphi)^{-1}(E'))$ este teorie.*

Demonstraţie: *Din lema închiderii*

$$Sen(\varphi)(Sen(\varphi)^{-1}(E')^{\bullet}) \subseteq Sen(\varphi)(Sen(\varphi)^{-1}(E'))^{\bullet}.$$

Folosind prima proprietate din observaţia 4.5.1 şi faptul că închiderea este crescătoare

$$Sen(\varphi)(Sen(\varphi)^{-1}(E'))^{\bullet} \subseteq E'^{\bullet} = E'.$$

Prin tranzitivitatea incluziunii

$$Sen(\varphi)(Sen(\varphi)^{-1}(E')^{\bullet}) \subseteq E',$$

deci $Sen(\varphi)^{-1}(E')^{\bullet} \subseteq Sen(\varphi)^{-1}(E')$, conform proprietăţii a doua din observaţia 4.5.1.

4.6.1 Instituţia teoriilor

Un morfism de teorii este un morfism de prezentări între două teorii. Se obnţine astfel o nouă categorie, categoria teoriilor, notată $Th(\mathcal{I})$. Prin definiţie, $Th(\mathcal{I})$ este o subcategorie plină a lui $Pres(\mathcal{I})$.

Putem considera restricţia lui Mod^p la categoria teoriilor pe care o vom nota cu Mod^t.

$$
\begin{tikzcd}
Th(\mathcal{I}) \arrow[r, hook] \arrow[rr, bend right, "Mod^t"'] & Pres(\mathcal{I}) \arrow[r, "Mod^p"] & Cat^{op}
\end{tikzcd}
$$

4.7 INSTITUŢIA LOGICII ECUAŢIONALE MULTISORTATE

Instituţia logicii ecuaţionale multisortate este următoarea construcţie

$$\begin{tikzcd} & Set \\ Sig \arrow[ur, "Sen"] \arrow[dr, "Alg"'] & \\ & Cat^{op} \end{tikzcd}$$

unde categoria Sig a signaturilor algebrice, functorii Alg şi Sen şi relaţia de satisfacere $\models_\Sigma$ vor fi definite în secţiunile următoare.

4.7.1 Categoria signaturilor algebrice Sig

Definiţia 4.7.1 Fie S o mulţime şi $\Sigma = \{\Sigma_{w,s}\}_{w\in S^*, s\in S}$ o familie de mulţimi. Perechea

$(S, \{\Sigma_{w,s}\}_{w\in S^*, s\in S})$ se numeşte **signatură algebrică**.

În cele ce urmează vom folosi şi notaţiile $(S, \Sigma_{w,s})$ sau (S, Σ).

Vom nota cu Sig categoria signaturilor algebrice construită astfel :

Obiectele categoriei Sig sunt signaturile algebrice.

Dacă $(S, \Sigma_{w,s})$ şi $(S', \Sigma'_{w',s'})$ sunt două signaturi algebrice, **morfismele** categoriei Sig între obiectele $(S, \Sigma_{w,s})$ şi $(S', \Sigma'_{w',s'})$ sunt de forma

$$\varphi = (f, \{g_{w,s}\}_{w\in S^*, s\in S}) : (S, \Sigma_{w,s}) \longrightarrow (S', \Sigma'_{w',s'}),$$

unde $f : S \longrightarrow S'$ şi $g_{w,s} : \Sigma_{w,s} \longrightarrow \Sigma'_{f^*(w), f(s)}$ pentru orice $w \in S^*, s \in S$ sunt funcţii. (Am notat cu $f^* : S^* \longrightarrow S'^*$ funcţia definită prin $f^*(s_1 s_2 \ldots s_n) = f(s_1)f(s_2)\ldots f(s_n)$, pentru orice $n \in I\!N$, şi $s_i \in S$ pentru orice $1 \le i \le n$).

În cele ce urmează vom utiliza şi notaţia $\varphi = (f, g)$.

Operaţia de compunere a morfismelor se defineşte astfel :

Fie $\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$ şi $\varphi' = (f', g') : (S', \Sigma') \longrightarrow (S'', \Sigma'')$ două morfisme. Atunci $\varphi; \varphi' : (S, \Sigma) \longrightarrow (S'', \Sigma'')$ este morfismul

$$\varphi; \varphi' = (f; f', \{g_{w,s}; g'_{f^*(w), f(s)}\}_{w\in S^*, s\in S}).$$

Propoziţie 4.7.2 *Sig este o categorie.*

Demonstraţie: Pentru orice signatură algebrică (S, Σ) morfismul identic este $1_{(S,\Sigma)} = (1_S, \{1_{\Sigma_{w,s}}\}_{w\in S^*, s\in S})$.
În plus, operaţia de compunere a morfismelor este asociativă. Într-adevăr, fie
$\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$, $\quad \varphi' = (f', g') : (S', \Sigma') \longrightarrow (S'', \Sigma'')$ şi
$\varphi'' = (f'', g'') : (S'', \Sigma'') \longrightarrow (S''', \Sigma''')$ trei morfisme de signaturi algebrice. Atunci

$$\begin{aligned} (\varphi; \varphi'); \varphi'' &= ((f; f'); f'', (g_{w,s}; g'_{f^*(w), f(s)}); g''_{(f;f')^*(w), (f;f')(s)}) \\ &= (f; (f'; f''), g_{w,s}; (g'_{f^*(w), f(s)}; g''_{f'^*(f^*(w)), f'(f(s))})) \\ &= (f, g); (f'; f'', g'_{w',s'}; g''_{f'^*(w'), f'(s')}) \\ &= \varphi; (\varphi'; \varphi'') \end{aligned}$$

Deci Sig este categorie.

4.7.2 Functorul Alg

Definim functorul

$$Alg : Sig \longrightarrow Cat^{op}.$$

Pe obiecte:

$$Alg(S,\Sigma) = Alg_{\Sigma},$$

pentru orice $(S,\Sigma) \in\ |Sig|$, unde Alg_{Σ} este categoria Σ-algebrelor.

Un obiect $\mathcal{A}$ din $|Alg_{\Sigma}|$ este de forma $\mathcal{A} = (A, \{A^{w,s}\}_{w\in S^*, s\in S})$, unde

$$A : S \longrightarrow Set \text{ şi } A^{w,s} : \Sigma_{w,s} \longrightarrow (A_w \to A_s) \text{ pentru orice } w \in S^*, s \in S \text{ sunt funcţii.}$$

Vom nota $A_s = A(s)$, pentru orice $s \in S$. Reamintim că $A_{s_1 s_2 \ldots s_n} = A_{s_1} \times A_{s_2} \times \cdots \times A_{s_n}$. Dacă $\sigma \in \Sigma_{w,s}$ în loc de $A^{w,s}(\sigma)$ mai scriem $A^{w,s}_{\sigma}$ sau chiar A_{σ} dacă nu este pericol de confuzie.

Pe morfisme: pentru orice $(f,g) \in Sig((S,\Sigma),(S',\Sigma'))$, functorul

$$Alg(f,g) : Alg_{\Sigma'} \longrightarrow Alg_{\Sigma}$$

este definit astfel:

a) Pe obiecte: Fie $(A, \{A^{w,s}\}_{w\in S'^*, s\in S'})$ o Σ'–algebră. Atunci

$$Alg(f,g)(A, \{A^{w',s'}\}_{w'\in S'^*, s'\in S'}) = (f;A, \{g_{w,s}; A^{f^*(w),f(s)}\}_{w\in S^*, s\in S}) \in |Alg_{\Sigma}|$$

Din $f : S \longrightarrow S'$ şi $A : S' \longrightarrow Set$ rezultă că $f;A : S \longrightarrow Set$. Conform notaţiei precedente, $A_{f(s)} = (f;A)_s$ pentru orice $s \in S$. Prin urmare elementele de sort s din $Alg(f,g)(A, \{A^{w',s'}\}_{w'\in S'^*, s'\in S'})$ sunt elementele de sort $f(s)$ din $(A, \{A^{w',s'}\}_{w'\in S'^*, s'\in S'})$.

Observaţie: Dacă $w = s_1 \ldots s_n$, atunci
$(f;A)_w = (f;A)_{s_1} \times \cdots \times (f;A)_{s_n} = A_{f(s_1)} \times \cdots \times A_{f(s_n)} = A_{f(s_1)\ldots f(s_n)} = A_{f^*(w)}$.

Deoarece $g_{w,s} : \Sigma_{w,s} \longrightarrow \Sigma'_{f^*(w),f(s)}$ şi $A^{f^*(w),f(s)} : \Sigma'_{f^*(w),f(s)} \longrightarrow (A_{f^*(w)} \longrightarrow A_{f(s)})$ rezultă că $g_{w,s}; A^{f^*(w),f(s)} : \Sigma_{w,s} \longrightarrow ((f;A)_w \longrightarrow (f;A)_s)$, pentru orice $w \in S^*$ şi orice $s \in S$.

b) Pe morfisme : Fie $h : (A, A^{w',s'}) \longrightarrow (B, B^{w',s'})$ un morfism de Σ'–algebre. Definim $Alg(f,g)(h) : (f;A, \{g_{w,s}; A^{f^*(w),f(s)}\}) \longrightarrow (f;B, \{g_{w,s}; B^{f^*(w),f(s)}\})$ prin

$$Alg(f,g)(h) = f;h,$$

unde $(f;h)(s) = h_{f(s)} : A_{f(s)} \longrightarrow B_{f(s)}$ pentru orice $s \in S$.

Arătăm că definiţia este corectă, adică familia de funcţii $f;h$ este un morfism de Σ–algebre. Aceasta revine la a arăta că pentru orice $w \in S^*$, orice $s \in S$ şi orice $\sigma \in \Sigma_{w,s}$ diagrama următoare este comutativă

$$\begin{tikzcd}
(f;A)_w \arrow[r, "(g_{w,s};A^{f^*(w),f(s)})(\sigma)"] \arrow[d, "(f;h)_w"'] & (f;A)_s \arrow[d, "(f;h)_s"] \\
(f;B)_w \arrow[r, "(g_{w,s};B^{f^*(w),f(s)})(\sigma)"] & (f;B)_s
\end{tikzcd}$$

Comutativitatea diagramei precedente este echivalentă cu comutativitatea diagramei

$$\begin{tikzcd}
A_{f^*(w)} \arrow[r, "A^{f^*(w),f(s)}(g_{w,s}(\sigma))"'] \arrow[d, "h_{f^*(w)}"'] & A_{f(s)} \arrow[d, "h_{f(s)}"] \\
B_{f^*(w)} \arrow[r, "B^{f^*(w),f(s)}(g_{w,s}(\sigma))"] & B_{f(s)}
\end{tikzcd}$$

iar aceasta rezultă din faptul că h este morfism de Σ'–algebre.

Propoziţie 4.7.3 *Alg(f,g) este functor.*

Demonstraţie: Dacă $1_{\mathcal{A}}$ este morfismul identic al unei Σ-algebre, atunci $Alg(f,g)(1_{\mathcal{A}}) = 1_{Alg(f,g)(\mathcal{A})}$.

Fie $h : \mathcal{A} \longrightarrow \mathcal{B}$ şi $h' : \mathcal{B} \longrightarrow \mathcal{C}$ două morfisme de Σ'–algebre. Prin urmare pentru orice $s \in S$

$$\begin{aligned}
(Alg(f,g)(h); Alg(f,g)(h'))_s &= Alg(f,g)(h)_s; Alg(f,g)(h')_s \\
&= (f;h)_s; (f;h')_s = h_{f(s)}; h'_{f(s)} \\
&= (h;h')_{f(s)} = (f;(h;h'))_s = (Alg(f,g)(h;h'))_s,
\end{aligned}$$

deci $Alg(f,g)(h); Alg(f,g)(h') = Alg(f,g)(h;h')$. Aşadar $Alg(f,g)$ este functor.

Propoziţie 4.7.4 *Alg este functor.*

Demonstraţie: Pentru orice signatură algebrică (S,Σ) avem evident $Alg(1_{(S,\Sigma)}) = 1_{Alg_\Sigma}$.

Fie $\varphi = (f,g) : (S,\Sigma) \longrightarrow (S',\Sigma')$ şi $\varphi' = (f',g') : (S',\Sigma') \longrightarrow (S'',\Sigma'')$ două morfisme de signaturi algebrice. Vom arăta că functorii $Alg(\varphi'); Alg(\varphi)$ şi $Alg(\varphi;\varphi')$ coincid, demonstrând că acţionează identic pe obiecte şi pe morfisme.

Fie $\mathcal{A} = (A, \{A^{w,s}\}_{w\in S''^*, s\in S''})$ o Σ''–algebră. Atunci

$$\begin{aligned}
(Alg(\varphi'); Alg(\varphi))(\mathcal{A}) &= Alg(\varphi)(Alg(\varphi')(\mathcal{A})) = Alg(\varphi)(f'; A, \{g'_{w,s}; A^{f'^*(w),f'(s)}\}_{w\in S'^*, s\in S'}) = \\
&= (f;(f';A), \{g_{w,s}; (g'_{f^*(w),f(s)}; A^{f'^*(f^*(w)),f'(f(s))})\}_{w\in S^*, s\in S}) \\
&= ((f;f');A, \{(g_{w,s}; g'_{f^*(w),f(s)}); A^{(f;f')^*(w)),(f;f')(s)}\}_{w\in S^*, s\in S}) \\
&= Alg(f;f', \{g_{w,s}; g'_{f^*(w),f(s)}\})(\mathcal{A}) = Alg(\varphi;\varphi')(\mathcal{A}).
\end{aligned}$$

Pentru orice morfism de Σ''–algebre $h : \mathcal{A} \longrightarrow \mathcal{B}$

$$(Alg(\varphi'); Alg(\varphi))(h) = Alg(f,g)(Alg(f',g')(h)) = Alg(f,g)(f';h) = f;(f';h) = (f;f');h = Alg(\varphi;\varphi')(h).$$

Deci $Alg(\varphi); Alg(\varphi') = Alg(\varphi;\varphi')$,oricare ar fi morfismele de signaturi algebrice φ şi φ' . Aşadar Alg este functor.

4.7.3 Functorul *Sen*

Un comentariu este necesar pentru a motiva prezentarea de mai jos.

Propoziţiile logicii ecuaţionale sunt şi propoziţii ale calculului cu predicare cu egalitate. Reciproca nu este adevarată. În multe variante ale calculului cu predicate apar variabile libere şi variabile legate, fapt ce îngreuneaza, cel putin din punct de vedere tehnic, atât expunerea cât şi inţelegerea. Vom folosi din nou artificiul de transformare a variabilelor în operaţii constante. După cum am mai facut-o în capitolele precedente, această transformare a variabilelor în constante se realizează prin schimbarea signaturii. Acest fapt explică utilizarea unei algebre iniţiale în locul unei algebrei libere pentru modelarea "expresiilor" din calculul cu predicate.

Pentru orice morfism de signaturi algebrice $\varphi : (S, \Sigma) \longrightarrow (S', \Sigma')$, notăm cu $\varphi^{\#} : T_\Sigma \longrightarrow Alg(\varphi)(T_{\Sigma'})$ unicul morfism de Σ–algebre. Reamintim că T_Σ este Σ–algebră iniţială.

Definim functorul $Sen : Sig \longrightarrow Set$ astfel :

1. Pe obiecte : Pentru orice signatură algebrică Σ

$$Sen(\Sigma) = T_\Sigma \times T_\Sigma = \{l \overset{\circ}{=}_s r | s \in S, \;\; l, r \in (T_\Sigma)_s\}_{s \in S}.$$

2. Pe morfisme : Pentru orice morfism de signaturi algebrice $\varphi : (S, \Sigma) \longrightarrow (S', \Sigma')$ funcţia $Sen(\varphi) : Sen(\Sigma) \longrightarrow Sen(\Sigma')$ este definită prin

$$Sen(\varphi)(l \overset{\circ}{=}_s r) = \varphi^{\#}_s(l) \overset{\circ}{=}_{f(s)} \varphi^{\#}_s(r).$$

Propoziţie 4.7.5 *Sen este functor.*

Demonstraţie:

Fie $\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$ şi $\psi = (f', g') : (S', \Sigma') \longrightarrow (S'', \Sigma'')$ două morfisme de signaturi algebrice. Deoarece $Alg(\varphi)(Alg(\psi)(T_{\Sigma''})) = (Alg(\psi); Alg(\varphi))(T_{\Sigma''}) = Alg(\varphi; \psi)(T_{\Sigma''})$ şi $\psi^{\#} : T_{\Sigma'} \to Alg(\psi)(T_{\Sigma''})$, deducem

$$Alg(\varphi)(\psi^{\#}) : Alg(\varphi)(T_{\Sigma'}) \longrightarrow Alg(\varphi; \psi)(T_{\Sigma''}),$$

prin urmare există două morfisme

$$T_\Sigma \;\underset{(\varphi;\psi)^{\#}}{\overset{\varphi^{\#}; Alg(\varphi)(\psi^{\#})}{\rightrightarrows}}\; Alg(\varphi; \psi)(T_{\Sigma''})$$

care deoarece T_Σ este Σ–algebră iniţială sunt egale. Aşadar avem

$$\begin{aligned}
(Sen(\varphi); Sen(\psi))(l \overset{\circ}{=}_s r) &= Sen(\psi)(\varphi^{\#}_s(l) \overset{\circ}{=}_{f(s)} \varphi^{\#}_s(r)) \\
&= (\psi^{\#}_{f(s)}(\varphi^{\#}_s(l)) \overset{\circ}{=}_{f'(f(s))} \psi^{\#}_{f(s)}(\varphi^{\#}_s(r)) \\
&= (\varphi^{\#}_s; Alg(\varphi)(\psi^{\#})_s)(l) \overset{\circ}{=}_{(f;f')(s)} (\varphi^{\#}_s; Alg(\varphi)(\psi^{\#})_s)(r) \\
&= (\varphi; \psi)^{\#}_s(l) \overset{\circ}{=}_{(f;f')(s)} (\varphi; \psi)^{\#}_s(r) \\
&= Sen(\varphi; \psi)(l \overset{\circ}{=}_s r),
\end{aligned}$$

deci Sen este functor.

4.7.4 Incluziuni de signaturi

Definiţia 4.7.6 O **incluziune de signaturi** este un morfism de signaturi

$$\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$$

care verifică condiţiile suplimentare

1. S este o submulţime a lui S' şi f este funcţia incluziune a lui S în S'

2. Pentru orice $w \in S^*$ şi $s \in S$, $\Sigma_{w,s}$ este o submulţime a lui $\Sigma'_{f^*(w),f(s)}$ şi funcţia $g_{w,s}$ este incluziunea dintre cele două mulţimi.

Propoziţie 4.7.7 *Dacă φ este o incluziune de signaturi, atunci $Alg(\varphi)$ este un functor uituc.*

Cuantificarea $.s$ are rolul de a indica sortul.

Fie (S, Σ) o signatură algebrică. Vom nota cu L_Σ o anumită Σ-algebră iniţială. Suportul ei este definit drept cea mai mică familie $\{T_s\}_{s \in S}$ de mulţimi cu proprietatea

$$\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}, x_i \in T_{s_i} \Longrightarrow \sigma(x_1, x_2, \dots, x_n).s \in T_s.$$

Folosind construcţia de mai sus observăm că pentru orice incluziune de signaturi $i : (S, \Sigma) \hookrightarrow (S', \Sigma')$ obţinem $(L_\Sigma)_s \subset (L_{\Sigma'})_s$ pentru orice $s \in S$. Mai mult L_Σ este subalgebră a lui $Alg(i)(L_{\Sigma'})$, morfismul de incluziune al subalgebrei L_Σ în algebra $Alg(i)(L_{\Sigma'})$ fiind chiar $i^\#$.

Propoziţie 4.7.8 *Dacă i este o incluziune de signaturi, atunci $Sen(i)$ este o incluziune de mulţimi.*

4.7.5 Relaţia şi condiţia de satisfacere

Vom defini în continuare relaţia de satisfacere.

Definiţia 4.7.9 Fie $\mathcal{A}$ o Σ–algebră. Spunem că $\mathcal{A}$ satisface Σ–ecuaţia $l \overset{\circ}{=}_s r$ dacă şi numai dacă $u_s(l) = u_s(r)$ pentru unicul Σ–morfism $u : T_\Sigma \longrightarrow \mathcal{A}$.

Notaţii 4.7.10 *Dacă $\mathcal{A}$ satisface $l \overset{\circ}{=}_s r$, vom nota $\mathcal{A} \models_\Sigma l \overset{\circ}{=}_s r$.*

Cu această definiţie are loc

Propoziţie 4.7.11 *Fie $\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$ un morfism de signaturi algebrice, $l \overset{\circ}{=}_s r \in Sen(\Sigma)$ şi $\mathcal{B}$ o Σ'–algebră. Atunci este îndeplinită condiţia de satisfacere*

$$Alg(\varphi)(\mathcal{B}) \models_\Sigma l \overset{\circ}{=}_s r \quad \Longleftrightarrow \quad \mathcal{B} \models_{\Sigma'} Sen(\varphi)(l \overset{\circ}{=}_s r).$$

Demonstraţie: Notând cu $t : T_{\Sigma'} \longrightarrow \mathcal{B}$ unicul Σ'-morfism deducem că

$$\varphi^\#; Alg(\varphi)(t) : T_\Sigma \longrightarrow Alg(\varphi)(\mathcal{B})$$

este unicul Σ-morfism. Din definiţia relaţiei de satisfacere deducem următoarele echivalenţe:

$$\begin{gathered} Alg(\varphi)(\mathcal{B}) \models_\Sigma l \overset{\circ}{=}_s r \Leftrightarrow \\ (\varphi^\#; Alg(\varphi)(t))_s(l) = (\varphi^\#; Alg(\varphi)(t))_s(r) \Leftrightarrow \\ (\varphi^\#_s; Alg(\varphi)(t)_s)(l) = (\varphi^\#_s; Alg(\varphi)(t)_s)(r) \Leftrightarrow \\ t_{f(s)}(\varphi^\#_s(l)) = t_{f(s)}(\varphi^\#_s(r)) \Leftrightarrow \\ \mathcal{B} \models_{\Sigma'} \varphi^\#_s(l) \overset{\circ}{=}_{f(s)} \varphi^\#_s(r) \Leftrightarrow \\ \mathcal{B} \models_{\Sigma'} Sen(\varphi)(l \overset{\circ}{=}_s r). \end{gathered}$$

Cu aceasta condiţia de satisfacere este demonstrată.

Chapter 5

MODULARIZARE

5.1 LIMITE şi COLIMITE ÎNTR-O CATEGORIE

Limitele şi colimitele sunt noţiuni duale. Din punct de vedere tehnic nu contează pe care o tratăm. Vom prefera colimitele deoarece au mai multe aplicaţii în informatică.

Definiţia 5.1.1 Fie $F : I \longrightarrow \mathcal{C}$ un functor. Un **cocon** sau con inductiv $F(i) \xrightarrow{\gamma_i} A$ pentru functorul F este format dintr-un obiect A din categoria $\mathcal{C}$ şi o familie de morfisme $\gamma_i : F(i) \longrightarrow A$ indexată de obiectele i ale categoriei I cu proprietatea $F(\alpha); \gamma_j = \gamma_i$ pentru orice morfism $\alpha : i \longrightarrow j$ din I.

În loc de $F(i) \xrightarrow{\gamma_i} A$ mai scriem şi (γ_i, A). $\square$

Numele coconului provine din cel al conceptului dual numit **con**

Compunerea unui cocon cu un morfism este tot un cocon.

Observaţia 5.1.2 *Dacă* $F(i) \xrightarrow{\gamma_i} A$ *este un cocon pentru* $F : I \longrightarrow \mathcal{C}$ *şi dacă* $f : A \longrightarrow B$ *este un morfism, atunci* $F(i) \xrightarrow{\gamma_i; f} B$ *este un cocon pentru* F*.*

Demonstraţie: Pentru orice morfism $\alpha : i \longrightarrow j$ din I, din $F(\alpha); \gamma_j = \gamma_i$ deducem $F(\alpha); (\gamma_j; f) = \gamma_i; f$ deci $F(i) \xrightarrow{\gamma_i; f} B$ este un cocon pentru F. $\square$

Orice functor duce un cocon întrun cocon.

Observaţia 5.1.3 *Dacă* $F(i) \xrightarrow{\gamma_i} A$ *este un cocon pentru* $F : I \longrightarrow \mathcal{C}$ *şi dacă* $G : \mathcal{C} \longrightarrow \mathcal{D}$ *este un functor, atunci* $G(F(i)) \xrightarrow{G(\gamma_i)} G(A)$ *este un cocon pentru* $F; G$*.*

Demonstraţie: Pentru orice morfism $\alpha : i \longrightarrow j$ din I, din $F(\alpha); \gamma_j = \gamma_i$ deducem

$$(F : G)(\alpha); G(\gamma_j) = G(F(\alpha)); G(\gamma_j) = G(F(\alpha); \gamma_j) = G(\gamma_i),$$

deci $G(F(i)) \xrightarrow{G(\gamma_i)} G(A)$ este cocon pentru $F; G$. $\square$

Definiţia 5.1.4 Un cocon $F(i) \xrightarrow{\gamma_i} A$ pentru functorul $F : I \longrightarrow \mathcal{C}$ se numeşte **colimită** a lui F dacă pentru orice alt cocon $F(i) \xrightarrow{\beta_i} B$ există un unic morfism $f : A \longrightarrow B$ pentru care $\alpha_i; f = \beta_i$ pentru orice obiect i din I. Vom utiliza notaţia

$$colim\ \ F = F(i) \xrightarrow{\gamma_i} A. \ \Box$$

Definiţia 5.1.5 O categorie se numeşte mică dacă obiectele sale formează o mulţme.

Un caz particular de colimite sunt **sumele directe,** numite şi coproduse. Ele se obţin pentru o categorie mică I în care toate morfismele sunt identitaţi, adică o mulţime. Un functor de la I la categoria $\mathcal{C}$ este de fapt o familie $\{F(i)\}_{i\in I}$ de obiecte din $\mathcal{C}$ indexată de mulţimea I. Observăm că un cocon pentru F este pur şi simplu o familie de morfisme $\alpha_i : F(i) \longrightarrow A$, deoarece condiţia impusă mai sus pentru morfismele din I este automat îndeplinită pentru morfismele identitate.

Definiţia 5.1.6 Familia de morfisme $\alpha_i : F(i) \longrightarrow A$ este o sumă directă a familiei de obiecte $\{F(i)\}_{i\in I}$ dacă pentru orice altă familie de morfisme $\beta_i : F(i) \longrightarrow B$ indexată de obiectele i din I există un unic morfism $f : A \longrightarrow B$ cu proprietatea $\alpha_i; f = \beta_i$ pentru orice obiect i din I.

Menţionăm că în categoria ataşată unei mulţimi parţial ordonate sumele directe, dacă există, coincid cu supremurile.

5.1.1 Colimite de mulţimi

Propoziţie 5.1.7 *În cazul particular al teoriei multimilor, suma directă a unei familii $\{F_i\}_{i\in I}$ de mulţimi este reuniunea lor disjunctă $A = \bigcup_{i\in I} F_i \times \{i\}$ împreună cu familia de funcţii $\{\alpha_i : F_i \longrightarrow A\}_{i\in I}$ definită pentru orice obiect i din I şi orice $a \in F_i$ prin $\alpha_i(a) = (a, i)$.*

Demonstraţie: Fie $\{\beta_i : F_i \longrightarrow B\}_{i\in I}$ o familie de funcţii. Definim funcţia $\beta : A \longrightarrow B$ prin $\beta(a, i) = \beta_i(a)$ pentru orice $i \in I$ şi $a \in F_i$. Observăm că pentru orice $i \in I$ şi $a \in F_i$

$$(\alpha_i; \beta)(a) = \beta(\alpha_i(a)) = \beta(a, i) = \beta_i(a).$$

Deci pentru orice $i \in I$ rezultă că $\alpha_i; \beta = \beta_i$.

Pentru a proba unicitatea presupunem că funcţia $\gamma : A \longrightarrow B$ verifică egalitătile $\alpha_i; \gamma = \beta_i$ pentru orice $i \in I$. Observăm pentru orice $i \in I$ şi orice $a \in F_i$ că

$$\gamma(a, i) = \gamma(\alpha_i(a)) = (\alpha_i; \gamma)(a) = \beta_i(a) = \beta(a, i)$$

deci $\gamma = \beta$. $\Box$

Definiţia 5.1.8 **Categoria $\mathcal{C}$ are colimite** dacă orice functor $F : I \longrightarrow \mathcal{C}$ definit pe o categorie mică I are o colimită.

Propoziţie 5.1.9 *Categoria mulţimilor are colimite.*

Demonstraţie: Fie I o categorie mică şi $F : I \longrightarrow Set$ un functor. Vom construi o colimită pentru F.

Fie (α_i, A) suma directă a familiei de mulţimi $\{F(i)\}_{i\in|I|}$ construită ca mai sus. Definim în mulţimea A relaţia

$$R = \{((a,i),(F(\alpha)(a),j)) : i,j \in |I|, a \in F(i), \alpha \in I(i,j)\}.$$

Fie $\sim$ relaţia de echivalenţă generată de R, adică cea mai mică relaţie de echivalenţă din A care include R. Fie $A/\sim$ câtul lui A prin echivalenţa $\sim$ şi $s : A \longrightarrow A/\sim$ surjecţia canonică care duce fiecare element din A în clasa sa de echivalenţă. Vom proba că

$$F(i) \xrightarrow{\alpha_i;s} A/\sim$$

este o colimită a functorului F.

Probăm că $F(i) \xrightarrow{\alpha_i;s} A/\sim$ este cocon. Fie $\alpha \in I(i,j)$ un morfism din I. Pentru orice $a \in F(i)$ observăm că

$$(F(\alpha);(\alpha_j;s))(a) = (\alpha_j;s)(F(\alpha)(a)) = s(\alpha_j(F(\alpha)(a))) = s(F(\alpha)(a),j) = s(a,i) = (\alpha_i;s)(a)$$

deci $F(\alpha);(\alpha_j;s) = \alpha_i;s$.

Fie $F(i) \xrightarrow{\beta_i} B$ un alt cocon pentru functorul F. Deoarece $F(i) \xrightarrow{\alpha_i} A$ este sumă directă există un unic morfism $f : A \longrightarrow B$ cu proprietatea $\alpha_i; f = \beta_i$ pentru orice obiect i din I. Probăm că relaţia R este inclusă în echivalenţa nucleară a lui f. Fie $\alpha \in I(a,b)$ un morfism din I şi $a \in F(i)$. Observăm că

$$f(F(\alpha)(a),j) = f(\alpha_j(F(\alpha)(a))) = (\alpha_j;f)(F(\alpha)(a)) = \beta_j(F(\alpha)(a)) = (F(\alpha);\beta_j)(a) =$$

$$\text{(deoarece } \{\beta_i\} \text{ este cocon)} \qquad = \beta_i(a) = (\alpha_i;f)(a) = f(\alpha_i(a)) = f(a,i).$$

Rezultă că $\sim$ este inclusă în echivalenţa nucleară a lui f, prin urmare există o unică funcţie $g : A/\sim \longrightarrow B$ cu proprietatea $s;g = f$. Deci

$$(\alpha_i;s);g = \alpha_i;(s;g) = \alpha_i;f = \beta_i.$$

Dacă $h : A/\sim \longrightarrow B$ este o altă funcţie cu proprietatea $(\alpha_i;s);h = \beta_i$ pentru orice obiect i din I. Deoarece $\alpha_i;(s;h) = \beta_i$ pentru orice obiect i din I deducem $s;h = f$, prin urmare din $s;h = s;g$ deducem datorită surjectivităţii lui s că $h = g$. $\square$

Observaţia 5.1.10 *Familia de funcţii* $F(i) \xrightarrow{\alpha_i;s} A/\sim$ *are proprietatea*

pentru orice $b \in A/\sim$ *există* $i \in |I|$ *şi* $a \in F(i)$ *cu proprietatea* $b = (\alpha_i;s)(a)$

fapt pentru care se numeşte epimorfă.

Demonstraţie: Fie $b \in A/\sim$. Deoarece funcţia $s : A \longrightarrow A/\sim$ este surjectivă există $c \in A$ cu propietatea $b = s(c)$. Din definiţia lui A există $i \in |I|$ şi $a \in F(i)$ astfel încât $c = (a,i)$. Prin urmare
$(\alpha_i;s)(a) = s(\alpha_i(a)) = s(a,i) = s(c) = b$.

5.1.2 Despre unicitatea colimitelor

Pentru un functor dat $F : I \longrightarrow \mathcal{C}$ colimitele lui sunt unice abstracţie făcând de un izomorfism. Acest fapt este ilustrat de urmatoarele două propoziţii.

Propoziţie 5.1.11 *Dacă* (α_i, A) *este o colimită a functorului* $F : I \longrightarrow \mathcal{C}$ *şi* $j : A \longrightarrow B$ *un izomorfism, atunci* $(\alpha_i;j, B)$ *este o colimită a lui* F.

Demonstraţie: Observăm că $(\alpha_i; j, B)$ este cocon deoarece se obţine din alt cocon prin compunere cu un morfism.

Fie (β_i, C) un cocon pentru F. Deoarece (α_i, A) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ există un unic morfism $f : A \longrightarrow C$ cu proprietatea $\alpha_i; f = \beta_i$ pentru orice obiect i din I.

Observăm că $(\alpha_i; j); (j^{-1}; f) = \beta_i$ pentru orice obiect i din I. În plus dacă $g : B \longrightarrow C$ este un alt morfism cu proprietatea $(\alpha_i; j); g = \beta_i$ pentru orice obiect i din I, deducem $\alpha_i; (j; g) = \beta_i$ pentru orice obiect i din I. Prin urmare $f = j; g$, deci $g = j^{-1}; f$.

Propoziţie 5.1.12 *Două colimite ale aceluiaşi functor sunt izomorfe.*

Demonstraţie: Fie (α_i, A) şi (β_i, B) două colimite ale functorului $F : I \longrightarrow \mathcal{C}$.

Deoarece (α_i, A) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ şi (β_i, B) este cocon pentru F există un unic morfism $f : A \longrightarrow B$ cu proprietatea $\alpha_i; f = \beta_i$ pentru orice obiect i din I.

Deoarece (β_i, B) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ şi (α_i, A) este cocon pentru F există un unic morfism $g : B \longrightarrow A$ cu proprietatea $\beta_i; g = \alpha_i$ pentru orice obiect i din I.

Deoarece $\alpha_i; (f; g) = (\alpha_i; f); g = \beta_i; g = \alpha_i = \alpha_i; 1_A$ pentru orice obiect i din I. Ţinând cont că (α_i, A) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ deducem din unicitate că $f; g = 1_A$.

Deoarece $\beta_i; (g; f) = (\beta_i; g); f = \alpha_i; f = \beta_i = \beta_i; 1_B$ pentru orice obiect i din I. Ţinând cont că (β_i, B) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ deducem din unicitate că $g; f = 1_B$.

Egalităţile $f; g = 1_A$ şi $g; f = 1_B$ arată că f şi g sunt izomorfisme inverse unul altuia.

5.1.3 Crearea colimitelor

Definiţia 5.1.13 Fie $F : I \longrightarrow \mathcal{C}$ un functor şi (γ_i, C) o colimită a sa. Spunem că functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ **conservă** colimita (γ_i, C) dacă $(G(\gamma_i), G(C))$ este o colimită a functorului $F; G$. Spunem că functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ conservă colimitele dacă el conservă orice colimită a oricărui functor $F : I \longrightarrow \mathcal{C}$.

Observaţia 5.1.14 *Dacă functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ conservă o colimită a functorului $F : I \longrightarrow \mathcal{C}$, atunci functorul G conservă orice colimită a functorului F.*

Demonstraţie: Considerăm o colimită a functorului F

$$Colim\ F = (\gamma_i, C).$$

pe care functorul G o duce într-o colimită a functorului $F; G$

$$Colim\ F; G = (G(\gamma_i), G(C)).$$

Fie (γ'_i, C') o altă colimită a functorului F.

Cele două colimite ale lui F fiind izomorfe rezultă existenţa izomorfismului $f : C \longrightarrow C'$ cu proprietatea $\gamma_i; f = \gamma'_i$ pentru orice obiect i din I.

Aplicând G deducem $G(\gamma_i); G(f) = G(\gamma'_i)$. Deoarece $G(f)$ este izomorfism şi $Colim\ F; G = (G(\gamma_i), G(C))$ rezultă că $(G(\gamma'_i), G(C'))$ este o colimită a functorului $F; G$.

Definiţia 5.1.15 Functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ **crează slab(crează) colimitele** dacă pentru orice functor $F : I \longrightarrow \mathcal{C}$ şi pentru orice colimită $G(F(i)) \xrightarrow{\delta_i} D$ functorului $F; G$ există(şi e unică) o colimită $F(i) \xrightarrow{\gamma_i} C$ a functorului F cu proprietăţile $G(C) = D$ şi $G(\gamma_i) = \delta_i$ pentru orice obiect i din I.

Propoziţie 5.1.16 *Dacă categoria $\mathcal{D}$ are colimite şi functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ crează slab colimitele, atunci categoria $\mathcal{C}$ are colimite şi functorul G comută cu colimitele.*

Demonstraţie: Fie $F : I \longrightarrow \mathcal{C}$ un functor şi o colimită a functorului $F;G$

$$Colim\ F;G = G(F(i)) \xrightarrow{\delta_i} D.$$

Deoarece functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ crează slab colimitele există o colimită $F(i) \xrightarrow{\gamma_i} C$ a functorului F cu proprietăţile $G(C) = D$ şi $G(\gamma_i) = \delta_i$ pentru orice obiect i din I. Am probat existenţa unei colimite a lui F cu care G comută. Este acum suficient să aplicăm observaţia precedentă.

5.2 COLIMITE DE SIGNATURI ALGEBRICE

Functorul uituc $U : Sig \longrightarrow Set$ este definit astfel:

- pe obiecte: pentru orice signatură algebrică $(S, \Sigma) \in |Sig|$

$$U(S, \Sigma) = S;$$

- pe morfisme: pentru orice morfism de signaturi algebrice $(f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$

$$U(f, g) = f.$$

Teorema 5.2.1 *Functorul uituc $U : Sig \longrightarrow Set$ crează slab(dar nu crează) colimitele.*

Demonstraţie: Fie I o categorie mică şi $F : I \longrightarrow Sig$ un functor. Folosim următoarele notaţii:

$$F(i) = (S_i, \Sigma^i) \quad \text{pentru orice obiect } i \in |I|,$$

$$F(\alpha) = (f_\alpha, g^\alpha) : (S_i, \Sigma^i) \longrightarrow (S_j, \Sigma^j) \quad \text{pentru orice morfism } \alpha \in I(i, j).$$

Dacă $\alpha : i \longrightarrow j$ şi $\beta : j \longrightarrow k$ sunt morfisme din I din $F(\alpha;\beta) = F(\alpha); F(\beta)$, adică

$$(f_{\alpha;\beta}\ ,\ g^{\alpha;\beta}_{w^i,s^i}) = (f_\alpha; f_\beta\ ,\ g^\alpha_{w^i,s^i}; g^\beta_{f^*_\alpha(w^i), f_\alpha(s^i)})$$

deducem

$$f_{\alpha;\beta} = f_\alpha; f_\beta \quad \text{şi} \quad g^{\alpha;\beta}_{w^i,s^i} = g^\alpha_{w^i,s^i}; g^\beta_{f^*_\alpha(w^i), f_\alpha(s^i)} \tag{5.1}$$

pentru orice $w^i \in S^*_i$ şi $s^i \in S_i$.

Fie

$$colimF;U = (f_i, S) = S_i \xrightarrow{f_i} S \tag{5.2}$$

Observăm că pentru orice morfism $\alpha : i \longrightarrow j$ din I

$$f_i = f_\alpha; f_j. \tag{5.3}$$

Pentru orice $w \in S^*$ **şi** $s \in S$ vom constui mulţimea $\Sigma_{w,s}$. Definim categoria mică $\mathcal{C}_{w,s}$ astfel: pe obiecte:

$$|\mathcal{C}_{w,s}| = \{\ (i, w^i, s^i)\ |\ \ i \in |I|, w^i \in S^*_i, s^i \in S_i,\ \text{cu}\ f^*_i(w^i) = w, f_i(s^i) = s\ \},$$

pe morfisme:

$$\mathcal{C}_{w,s}((i,w^i,s^i),(j,w^j,s^j)) = \{\alpha \in I(i,j) \mid f^*_\alpha(w^i) = w^j, f_\alpha(s^i) = s^j\}.$$

Compunerea morfismelor din $\mathcal{C}_{w,s}$ se face ca în categoria I. Pentru morfismele $\alpha : (i,w^i,s^i) \longrightarrow (j,w^j,s^j)$ şi $\beta : (j,w^j,s^j) \longrightarrow (k,w^k,s^k)$ observăm că

$$(f_{\alpha;\beta})(s^i) = f_\beta(f_\alpha(s^i)) = f_\beta(s^j)) = s^k \text{ şi că } (f^*_{\alpha;\beta})(w^i) = f^*_\beta(f^*_\alpha(w^i)) = f^*_\beta(w^j)) = w^k,$$

ceea ce arată că

$$\alpha \in \mathcal{C}_{w,s}((i,w^i,s^i),(j,w^j,s^j)) \text{ şi } \beta \in \mathcal{C}_{w,s}((j,w^j,s^j),(k,w^k,s^k)) \text{ implică } \alpha;\beta \in \mathcal{C}_{w,s}((i,w^i,s^i),(k,w^k,s^k)).$$

Definim functorul

$$G_{w,s} : \mathcal{C}_{w,s} \longrightarrow Set$$

pe obiecte:

$$G_{w,s}(i,w^i,s^i) = \Sigma^i_{w^i,s^i};$$

şi pe morfisme: pentru orice $\alpha \in \mathcal{C}_{w,s}((i,w^i,s^i),(j,w^j,s^j))$

$$G_{w,s}(\alpha) = g^\alpha_{w^i,s^i} : \Sigma^i_{w^i,s^i} \longrightarrow \Sigma^j_{w^j,s^j}.$$

Pentru morfismele $\alpha : (i,w^i,s^i) \longrightarrow (j,w^j,s^j)$ şi $\beta : (j,w^j,s^j) \longrightarrow (k,w^k,s^k)$ observăm că

$$G_{w,s}(\alpha);G_{w,s}(\beta) = g^\alpha_{w^i,s^i};g^\beta_{w^j,s^j} = g^\alpha_{w^i,s^i};g^\beta_{f^*_\alpha(w^i),f_\alpha(s^i)} = g^{\alpha;\beta}_{w^i,s^i} = G_{w,s}(\alpha;\beta)$$

Fie

$$colim\ G_{w,s} = (g^i_{w^i,s^i},\Sigma_{w,s}) = \Sigma^i_{w^i,s^i} \overset{g^i_{w^i,s^i}}{\longrightarrow} \Sigma_{w,s} \tag{5.4}$$

deoarece $f^*_i(w^i) = w$ şi $f^*_i(s^i) = s$ pentru orice $(i,w^i,s^i) \in |\mathcal{C}_{w,s}|$.

Pentru orice $i \in |I|$, $w^i \in S^*_i$ şi $s^i \in S_i$, pentru $w = f^*_i(w^i)$ şi $s = f^*_i(s^i)$ constatăm existenţa morfismului

$$g^i_{w^i,s^i} : \Sigma^i_{w^i,s^i} \longrightarrow \Sigma_{f^*_i(w^i),f_i(s^i)}.$$

Vom dovedi

$$colim\ F = ((f_i,g^i),(S,\Sigma)) = (S_i,\Sigma^i) \overset{(f_i,g^i_{w^i,s^i})}{\longrightarrow} (S,\Sigma).$$

A. Pentru început arătăm că $((f_i,g^i),(S,\Sigma))$ este un con inductiv de bază F.
Pentru $\alpha \in I(i,j)$ probăm că $(f_\alpha,g^\alpha)(f_j,g^j) = (f_i,g^i)$.

Deoarece egalitatea $f_\alpha;f_j = f_i$ este evidentă conform (5.3) mai trebuie probat pentru $w^i \in S^*_i$ şi $s^i \in S_i$ că

$$g^\alpha_{w^i,s^i}g^j_{f^*_\alpha(w^i),f_\alpha(s^i)} = g^i_{w^i,s^i}$$

Definind $w = f^*_i(w^i)$ şi $s = f_i(s^i)$ observăm că $(i,w^i,s^i) \in |\mathcal{C}_{w,s}|$. Deoarece $w = f^*_j(f^*_\alpha(w^i))$ şi $s = f_j(f_\alpha(s^i))$ observăm că $(j,f^*_\alpha(w^i),f_\alpha(s^i)) \in |\mathcal{C}_{w,s}|$ şi că $\alpha : (i,w^i,s^i) \longrightarrow (j,f^*_\alpha(w^i),f_\alpha(s^i))$ este morfism în $\mathcal{C}_{w,s}$. Proprietatea de con inductiv a lui $(g^i_{w^i,s^i},\Sigma_{w,s})$ aplicată lui α implică comutativitatea următoarei diagrame:

$$\begin{tikzcd}
G_{w,s}(i,w^i,s^i) = \Sigma^i_{w^i,s^i} \arrow[rr, "G_{w,s}(\alpha) = g^{\alpha}_{w^i,s^i}"] \arrow[dr, "g^i_{w^i,s^i}"'] & & \Sigma^j_{f^*_\alpha(w^i), f_\alpha(s^i)} = G_{w,s}(j, f^*_\alpha(w^i), f_\alpha(s^i)) \arrow[dl, "g^j_{f^*_\alpha(w^i), f_\alpha(s^i)}"] \\
& \Sigma_{w,s} &
\end{tikzcd}$$

Comutativitatea diagramei este chiar egalitatea de mai sus.

B. Fie $((u_i, v^i), (S', \Sigma')$ alt con inductiv de bază F.

Deoarece (u_i, S') este con inductiv de bază $F;U$ există un unic morfism $f : \ S \longrightarrow S'$ cu proprietatea $f_i; f = u_i$ pentru orice $i \in |I|$.

$$\begin{tikzcd}
S_i \arrow[rr, "f_i"] \arrow[dr, "u_i"'] & & S \arrow[dl, "f"] \\
& S' &
\end{tikzcd}$$

Pentru orice $i \in |I|$, orice $w^i \in S_i^*$ şi orice $s^i \in S$ considerăm morfismul

$$v^i_{w^i,s^i} : \Sigma^i_{w^i,s^i} \longrightarrow \Sigma'_{u^*_i(w^i), u_i(s^i)}$$

Dacă $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$ utilizând $f^*(w) = f^*(f^*_i(w^i)) = u^*_i(w^i)$ şi $f(s) = u_i(s^i)$ observăm că

$$v^i_{w^i,s^i} : G_{w,s}(i, w^i, s^i) \longrightarrow \Sigma'_{f^*(w), f(s)}.$$

Pentru orice $w \in S^*$ şi $s \in S$ probăm că

$$(v^i_{w^i,s^i}, \Sigma'_{f^*(w), f(s)})$$

este cocon pentru $G_{w,s}$. Intr-adevăr pentru orice morfism α de la (i, w^i, s^i) la (j, w^j, s^j) din $\mathcal{C}_{w,s}$ din proprietatea de cocon a lui $((u_i, v^i), (S', \Sigma'))$ aplicată morfismului $\alpha : i \longrightarrow j$ obţinem comutativitatea diagramei

$$\begin{tikzcd}
(S_i, \Sigma^i) \arrow[rr, "(f_\alpha, g^\alpha)"] \arrow[dr, "(u_i, v^i)"'] & & (S_j, \Sigma^j) \arrow[dl, "(u_j, v^j)"] \\
& (S', \Sigma') &
\end{tikzcd}$$

prin urmare folosind compenentele de pe a doua poziţie deducem $g^{\alpha}_{w^i,s^i} v^j_{w^j,s^j} = v^i_{w^i,s^i}$, deci următoarea diagramă este comutativă

$$\begin{tikzcd}
\Sigma^i_{w^i,s^i} \arrow[rr, "G_{w,s}(\alpha) = g^{\alpha}_{w^i,s^i}"] \arrow[dr, "v^i_{w^i,s^i}"'] & & \Sigma^j_{w^j,s^j} \arrow[dl, "v^j_{w^j,s^j}"] \\
& \Sigma'_{w,s} &
\end{tikzcd}$$

Prin urmare pentru orice $w \in S^*$ şi $s \in S$ există un unic morfism $g_{w,s}: \ \Sigma_{w,s} \longrightarrow \Sigma'_{f(w),f(s)}$ cu proprietatea $g^i_{w^i,s^i}; g_{w,s} = v^i_{w^i,s^i}$ pentru orice $(i, w^i, s^i) \in |\mathcal{C}w, s|$.

$$\begin{tikzcd}
G_{w,s}(i, w^i, s^i) = \Sigma^i_{w^i,s^i} \arrow[rr, "g^i_{w^i,s^i}"] \arrow[dr, "v^i_{w^i,s^i}"'] & & \Sigma_{w,s} \arrow[dl, "g_{w,s}"] \\
& \Sigma'_{f^*(w),f(s)} &
\end{tikzcd}$$

Deoarece $g^i_{w^i,s^i}; g_{f_i^*(w^i),f_i(s^i)} = v^i_{w^i,s^i}$ deducem că $(f_i, g^i); (f, g) = (u_i, v^i)$.

$$
\begin{tikzcd}
(S_i, \Sigma^i) \arrow[rr, "(f_i, g^i)"] \arrow[dr, "(u_i, v^i)"'] & & (S, \Sigma) \arrow[dl, "(f, g)"] \\
& (S', \Sigma') &
\end{tikzcd}
$$

Unicitatea se arată uşor. □

Folosind propoziţia 5.1.16 şi existenţa colimitelor din **Set** deducem

Corolar 5.2.2 *Categoria signaturilor algebrice are colimite şi functorul uituc comută cu colimitele.*

5.3 FUNCTORUL Alg CONSERVĂ COLIMITELE

Reamintim definiţia functorului

$$Alg : Sig \longrightarrow Cat^{op}$$

după cum urmează:

1) pe obiecte: pentru orice signatură algebrică (S, Σ)

$$Alg(S, \Sigma) = Alg_\Sigma$$

este categoria Σ-algebrelor.

Un obiect $\mathcal{A}$ din Alg_Σ este de forma: $\mathcal{A} = (A, \{A_{w,s}\}_{w\in S^*, s\in S})$ unde

$$A : S \longrightarrow Set \quad \text{şi} \quad A_{w,s} : \Sigma_{w,s} \longrightarrow (A_w \to A_s) \text{ pentru orice } w \in S^*, s \in S.$$

2) pe morfisme: pentru orice morfism de signaturi $(f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$

$$Alg(f, g) : Alg_{\Sigma'} \longrightarrow Alg_\Sigma,$$

este functorul definit astfel:

α) pe obiecte: pentru orice Σ'-algebra $(A, \{A_{w,s}\}_{w\in S'^*, s\in S'})$

$$Alg(f, g)(A, \{A_{w,s}\}_{w\in S'^*, s\in S'}) = (f; A, \{g_{w,s}; A_{f^*(w), f(s)}\}_{w\in S^*, s\in S}) \in |Alg_\Sigma|. \tag{5.5}$$

Din $f : S \longrightarrow S'$ şi $A : S' \longrightarrow Set$ rezultă că

$$f; A : S \longrightarrow Set;$$

Pentru orice $w \in S^*$ ąi $s \in S$ din $g_{w,s} : \Sigma_{w,s} \longrightarrow \Sigma'_{f^*(w), f(s)}$ şi $A_{f^*(w), f(s)} : \Sigma'_{f^*(w), f(s)} \longrightarrow (A_{f^*(w)} \longrightarrow A_{f(s)})$ deducem

$$g_{w,s}; A_{f^*(w), f(s)} : \Sigma_{w,s} \longrightarrow ((fA)_w \longrightarrow (fA)_s).$$

β) pe morfisme: pentru orice morfism h de Σ-algebre

$$Alg(f, g)(h) = f; h. \tag{5.6}$$

5.3.1 Alg conservă colimitele

Fie I o categorie mică şi $F : I \longrightarrow Sig$ un functor. Folosim aceleaşi notaţii şi construim ca în secţiunea anterioară

$$colim\ F = ((f_i, g^i), (S, \Sigma)) = (S_i, \Sigma_i) \stackrel{(f_i, g^i_{w^i, s^i})}{\longrightarrow} (S, \Sigma).$$

Vom arăta in continuare că functorul Alg comută cu colimitele, adică limita functorului $F; Alg$ este

$$Alg(f_i, g^i) : \ Alg_{\Sigma} \longrightarrow Alg_{\Sigma^i}.$$

Este un con fiind obţinut prin aplicarea functorului contravariant Alg unui cocon.

Fie $\mathcal{C}$ o categorie şi pentru orice $i \in |I|$, fie $G_i : \mathcal{C} \to Alg_{\Sigma^i}$ un functor. Pentru orice $c \in |\mathcal{C}|$, notăm Σ^i-algebra

$$G_i(c) = (G_i(c)^s, G_i(c)^o),$$

unde $G_i(c)^s$ este suport algebrei şi $G_i(c)^o$ sunt operaţiile algebrei.

Presupunem că $(\mathcal{C}, G_i)$ este con pentru $F; Alg$, adică pentru orice morfism $\alpha \in I(i, j)$, următoarea diagramă este comutativă:

$$\begin{tikzcd} & \mathcal{C} \arrow[dl, "G_i"'] \arrow[dr, "G_j"] & \\ Alg_{\Sigma^i} & & Alg_{\Sigma^j} \arrow[ll, "Alg(f_\alpha, g^\alpha)"] \end{tikzcd}$$

prin urmare pentru orice morfism $\alpha \in I(i, j)$

a) pentru orice obiect c din C

$$(G_i(c)^s, G_i(c)^o) = Alg(f_\alpha, g^\alpha)((G_j(c)^s, G_j(c)^o) = (f_\alpha; G_j(c)^s, \{g^\alpha_{w,s}; G_j(c)^o_{f^*_\alpha(w), f_\alpha(s)}\}_{w \in S^*_i, s \in S_i})$$

de unde pentru fiecare componentă obţinem

$$f_\alpha G_j(c)^s = G_i(c)^s \tag{5.7}$$

şi pentru orice $w^i \in S^*_i$ şi $s^i \in S_i$

$$G_i(c)^o_{w^i, s^i} = g^\alpha_{w^i, s^i}; G_j(c)^o_{f^*_\alpha(w^i), f_\alpha(s^i)}, \tag{5.8}$$

b) pentru orice morfism $u \in \mathcal{C}(c, c')$

$$Alg(f_\alpha, g^\alpha)(G_j(u)) = f_\alpha G_j(u) = G_i(u) \tag{5.9}$$

1. Existenţa

Vom construi un functor $G : \mathcal{C} \longrightarrow Alg_{\Sigma}$ astfel încât următoarea diagramă să fie comutativă pentru orice obiect i din I:

$$\begin{tikzcd} Alg_\Sigma \arrow[rr, "Alg(f_i, g^i)"] & & Alg_{\Sigma^i} \\ & \mathcal{C} \arrow[ul, "G"] \arrow[ur, "G_i"'] & \end{tikzcd}$$

Construcţia lui G pe obiecte

Fie c un obiect din categorie $\mathcal{C}$. Din relatia (5.7) rezultă că familia de funcţii $G_i(c)^s : S_i \longrightarrow Set$ formează un cocon pentru $F;U$ prin urmare există o unică funcţie $G(c)^s : S \longrightarrow Set$ astfel încât pentru orice obiect i din $|I|$

$$f_i; G(c)^s = G_i(c)^s \tag{5.10}$$

$$\begin{tikzcd} S_i \arrow[rr, "f_i"] \arrow[dr, "G_i(c)^s"'] & & S \arrow[dl, "G(c)^s"] \\ & Set & \end{tikzcd}$$

$G(c)^s$ astfel determinat va constitui suportul Σ-algebrei $G(c)$.

Din (5.10) deducem că pentru orice $s_i \in S_i$

$$G(c)^s_{f_i(s_i)} = G_i(c)^s_{s_i} \tag{5.11}$$

şi analog pentru orice cuvînt $w^i \in S_i^*$

$$G(c)^s_{f_i^*(w^i)} = G_i(c)^s_{w^i}. \tag{5.12}$$

Operaţiile Σ-algebrei $G(c)$ se construiesc astfel:
Fie $w \in S^*$ şi $s \in S$. Familia de funcţii $G_i(c)^o_{w^i,s^i} : \Sigma^i_{w^i,s^i} \longrightarrow (G_i(c)^s_{w^i} \longrightarrow G_i(c)^s_{s^i})$ indexată de $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$, formează, aşa cum ne arată relaţiile (5.8), un cocon pentru $G_{w,s}$. Este util să remarcăm aici că toate codomeniile funcţiilor din familia de mai sus coincid deoarece conform egalităţilor (5.11) şi (5.12)

$$G_i(c)^s_{w^i} \longrightarrow G_i(c)^s_{s^i} \quad \text{coincide cu} \quad G(c)^s_w \longrightarrow G(c)^s_s.$$

Prin urmare există o unică funcţie

$$G(c)^o_{w,s} : \Sigma_{w,s} \longrightarrow (G(c)^s_w \to G(c)^s_s)$$

astfel incât pentru orice $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$

$$g^i_{w^i,s^i} G(c)^o_{w,s} = G_i(c)^o_{w^i,s^i} \tag{5.13}$$

adică următoarea diagramă este comutativă:

$$\begin{tikzcd}
G_{w,s}(i,w^i,s^i)=\Sigma^i_{w^i,s^i} \arrow[rr, "g^i_{w^i,s^i}"] \arrow[dr, "G_i(c)^o_{w^i,s^i}"'] & & \Sigma_{w,s} \arrow[dl, "G(c)^o_{w,s}"] \\
& (G(c)^s_w \longrightarrow G(c)^s_s) &
\end{tikzcd}$$

Prin urmare pentru orice obiect $i \in |I|$ pentru orice $w^i \in S_i^*$ şi orice $s^i \in S_i$, pentru $w = f_i^*(w^i)$ şi $s = f_i(s^i)$ deoarece $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$ rezultă că

$$g^i_{w^i,s^i}; G(c)^o_{f_i^*(w^i),f_i(s^i)} = G_i(c)^o_{w^i,s^i}. \tag{5.14}$$

Pentru orice obiect $c \in |\mathcal{C}|$ prin definiţie

$$G(c) = (G(c)^s, G(c)^o)$$

cu $G(c)^s$ şi $G(c)^o$ definiţi mai sus.

Pentru orice $c \in |\mathcal{C}|$ observăm că

$$G_i(c) = (G_i(c)^s, G_i(c)^o) \overset{(5.10)}{=} (f_i; G(c)^s, G_i(c)^o) \overset{(5.14)}{=} (f_i; G(c)^s, \{g^i_{w^i,s^i}; G(c)^o_{f_i^*(w^i),f_i(s^i)}\}_{w^i,s^i}),$$

adică pentru orice $c \in |\mathcal{C}|$ şi orice $i \in |I|$

$$Alg(f_i, g^i)(G(c)) = G_i(c). \tag{5.15}$$

Construcţia lui G pe morfisme:

Fie $u : c \longrightarrow c'$ un morfism din $\mathcal{C}$. Observăm că pentru orice obiect i din I

$$G_i(u) : G_i(c) \longrightarrow G_i(c').$$

Familia funcţiilor indexată de obiectele lui I

$$G_i(u) : S_i \longrightarrow \bigcup_{i\in|I|}\bigcup_{s^i\in S_i}(G_i(c)^s_{s^i} \longrightarrow G_i(c')^s_{s^i})$$

formează în conformitate cu (5.9) un cocon pentru functorul $F; U$, prin urmare există şi este unică o funcţie

$$G(u) : S \rightarrow \bigcup (G_i(c)^s_{s^i} \rightarrow G_i(c)^s_{s^i})$$

astfel incât pentru orice $i \in |I|$

$$f_i; G(u) = G_i(u) \tag{5.16}$$

Vom arăta că $G(u)_s : \ G(c)^s_s \longrightarrow G(c')^s_s$ pentru orice $s \in S$.
Fie $s \in S$. Deoarece $\{f_i\}_{i\in I}$ este familie epimorfă conform observaţiei 5.1.10, există $i \in |I|$ şi $s_i \in S_i$, cu $f_i(s_i) = s$. Rezultă că

$$G(u)_s = G(u)_{f_i(s_i)} \overset{(5.16)}{=} G_i(u)_{s_i} : \ G_i(c)^s_{s_i} \longrightarrow G_i(c')^s_{s_i}.$$

Dar

$$G_i(c)^s_{s_i} \overset{(5.10)}{=} (f_i G(c)^s)_{s_i} = G(c)^s_{f_i(s_i)} = G(c)^s_s$$

şi analog

$$G_i(c')^s_{s_i} = G(c')^s_s$$

deci pentru orice $s \in S$

$$G(u)_s : G(c)^s_s \to G(c')^s_s.$$

G(u) este morfism de Σ-algebre

Arătăm că $\{G(u)_s\}_{s\in S}$ formează un morfism de la $G(c)$ in $G(c')$, pentru orice $u \in \mathcal{C}(c, c')$.

Fie $w \in S^*$, $s \in S$ şi $\sigma \in \Sigma_{w,s}$. Deoarece familia

$$\{g^i_{w^i,s^i}\}_{(i,w^i,s^i)\in|\mathcal{C}_{w,s}|}$$

este epimorfă, vezi şi 5.1.10, rezultă că există $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$ şi $\sigma^i \in \Sigma^i_{w^i,s^i}$ astfel incât

$$\sigma = g^i_{w^i,s^i}(\sigma^i) \tag{5.17}$$

Deoarece $G_i(u) : G_i(c) \longrightarrow G_i(c')$ este morfism de Σ^i-algebre, diagrama de mai jos este comutativă:

$$\begin{tikzcd}
G_i(c)^s_{w^i} \arrow[r, "G_i(c)^o_{w^i,s^i}(\sigma^i)"] \arrow[d, "G_i(u)_{w^i}"'] & G_i(c)^s_{s^i} \arrow[d, "G_i(u)_{s^i}"] \\
G_i(c')^s_{w^i} \arrow[r, "G_i(c')^o_{w^i,s^i}(\sigma^i)"] & G_i(c')^s_{s^i}
\end{tikzcd}$$

Dar $f_i(s^i) = s$ şi egalităţile (5.11, 5.16) implică $G_i(c)^s_{s^i} = G(c)^s_s$, $G_i(c')^s_{s^i} = G(c')^s_s$ şi $G_i(u)_{s^i} = G(u)_s$.

Analog $f_i^*(w^i) = w$ şi egalităţile (5.12, 5.16) implică $G_i(c)^s_{w^i} = G(c)^s_w$, $G_i(c')^s_{w^i} = G(c')^s_w$ şi $G_i(u)_{w^i} = G(u)_w$.

Din $f_i(s^i) = s$, $f_i^*(w^i) = w$ şi 5.13 deducem că

$$G_i(c)^o_{w^i,s^i}(\sigma^i) = G(c)^o_{w,s}(\sigma) \quad \text{şi} \quad G_i(c')^o_{w^i,s^i}(\sigma^i) = G(c')^o_{w,s}(\sigma)$$

prin urmare următoarea diagramă este comutativă:

$$\begin{tikzcd}
G(c)^s_w \arrow[r, "G(c)^o_{w,s}(\sigma)"] \arrow[d, "G(u)_w"'] & G(c)^s_s \arrow[d, "G(u)_s"] \\
G(c')^s_w \arrow[r, "G(c')^o_{w,s}(\sigma)"'] & G(c')^s_s
\end{tikzcd}$$

ceea ce arată că $G(u):\ G(c) \longrightarrow G(c')$ este morfism pentru $\sigma \in \Sigma_{w,s}$.
În concluzie $G(u):\ G(c) \longrightarrow G(c')$ este morfism de Σ-algebre.

G este functor

Fie $s \in S$. Deoarece $(f_i)_{i\in|I|}$ este epimorfă, 5.1.10, există $i \in |I|$ şi $s_i \in S_i$ astfel incât $f_i(s_i) = s$. Din
$G(1_c)_s = G(1_c)_{f_i(s_i)} = (f_i; G(1_c))_{s_i} \overset{5.16}{=} (G_i(1_c))_{s_i} \overset{G_i functor}{=} (1_{G_i(c)^s})_{s_i} \overset{5.10}{=} 1_{(f_i G(c)^s)_{s_i}} = 1_{G(c)^s_{f_i(s_i)}} = 1_{G(c)^s_s} = (1_{G(c)})_s$ deducem

$$G(1_c) = 1_{G(c)}.$$

Fie $u : c \longrightarrow c',\ v : c' \longrightarrow c''$. Fie $s \in S$. Observăm că există $i \in |I|$ şi există $s_i \in S_i$ astfel incât $f_i(s_i) = s$. Din

$$G(u)_s; G(v)_s = G(u)_{f_i(s_i)}; G(v)_{f_i(s_i)} \overset{5.16}{=} G_i(u)_{s_i}; G_i(v)_{s_i} = (G_i(u); G_i(v))_{s^i} = G_i(u; v)_{s^i} \overset{5.16}{=} (f_i; G(u; v))_{s^i} =$$

$$= G(u; v)_{f_i(s_i)} = G(u; v)_s.$$

deducem $G(u); G(v) = G(u; v)$. Deci G este functor.

Mai avem de probat că $G_i = G; Alg(f_i, g^i)$ pentru orice obiect $i \in |I|$. Întrucât egalitatea a fost deja probată pentru obiecte (5.15) o vom arăta numai pentru morfisme. Pentru $u:\ c \longrightarrow c'$ observăm că egalitatea (5.16) implică $Alg(f_i, g^i)(G(u)) = f_i G(u) = G_i(u)$.

2. Unicitatea

Vom arăta că G construit mai sus este unic.
Fie $H : \mathcal{C} \longrightarrow Alg_\Sigma$ un functor astfel încât pentru orice $i \in |I|$

$$H; Alg(f_i, g^i) = G_i.$$

Arătăm că $G = H$ pe obiecte şi pe morfisme. Pentru $c \in |\mathcal{C}|$ notăm $H(c) = (H(c)^s, H(c)^o)$.
a) pe obiecte:
pentru $i \in |I|$ şi $c \in |\mathcal{C}|$, observăm că

$$G_i(c) = Alg(f_i, g^i)(H(c)) = (f_i; H(c)^s, \{g^i_{w^i,s^i}; H(c)^o_{f_i^*(w^i), f_i(s^i)}\})$$

prin urmare $G_i(c)^s = f_i H(c)^s$ pentru orice $i \in |I|$, deci în conformitate cu unicitatea din (5.10) deducem $G(c)^s = H(c)^s$, adică algebrele $G(c)$ şi $H(c)$ au aceleaşi suporturi.

Pentru $i \in |I|$, $c \in |\mathcal{C}|$, $w^i \in S_i^*$ şi $s^i \in S_i$ din egalitatea de mai sus rezultă că

$$G_i(c)^o_{w^i,s^i} = g^i_{w^i,s^i} H(c)^o_{f_i^*(w^i), f_i(s^i)}.$$

Fie $w \in S^*$ şi $s \in S$. Pentru orice $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$ deducem

$$G_i(c)^o_{w^i,s^i} = g^i_{w^i,s^i} H(c)^o_{w,s}$$

deci, din unicitatea din (5.13) rezultă că

$$G(c)^o_{w,s} = H(c)^o_{w,s}$$

adică algebrele $G(c)$ şi $H(c)$ au aceleaşi operaţii.
Deci pentru orice obiect c din $\mathcal{C}$

$$G(c) = H(c).$$

b) pe morfisme:
Fie $u \in \mathcal{C}(c, c')$. Deoarece pentru orice $i \in |I|$

$$G_i(u) = Alg(f_i, g^i)(H(u)) = f_i; H(u),$$

din unicitatea din (5.16), deducem $G(u) = H(u)$.

Deci functorii G şi H coincid şi pe morfisme. În concluzie $G = H$.

5.4 COLIMITE ÎN INSTITUŢII

Fie $\mathcal{I}$ o instituţie, $Pres(\mathcal{I})$ categoria prezentărilor lui $\mathcal{I}$. Functorul uituc

$$V:\ Pres(\mathcal{I}) \longrightarrow Sign$$

este definit prin $V(\Sigma, E) = \Sigma$ pentru orice prezentare (Σ, E) şi $V(\phi) = \phi$ pentru orice morfism de prezentări ϕ.

Propoziţie 5.4.1 *Fie $\{(\Sigma_i, E_i)\}_{i\in I}$ o familie de prezentări şi $\phi_i:\ \Sigma_i \longrightarrow \Sigma$ o familie de morfisme de signaturi. Familia de morfisme de prezentări*

$$\phi_i:\ (\Sigma_i, E_i) \longrightarrow (\Sigma, \bigcup_{i\in I} Sen(\phi_i)(E_i))$$

are următoarea proprietate:

pentru orice prezentare (Σ', E) şi pentru orice morfism de signaturi $\psi:\ \Sigma \longrightarrow \Sigma'$ pentru care

$$\phi_i\psi:\ (\Sigma_i, E_i) \longrightarrow (\Sigma', E)$$

sunt morfisme de prezentări, $\psi:\ (\Sigma, \bigcup_{i\in I} Sen(\phi_i)(E_i)) \longrightarrow (\Sigma', E)$ este morfism de prezentări.

Demonstraţie: Deoarece

$$\phi_i\psi:\ (\Sigma_i, E_i) \longrightarrow (\Sigma', E)$$

sunt morfisme de prezentări, $Sen(\phi_i\psi)(E_i) \subseteq E^\bullet$ pentru orice $i \in I$. Deoarece

$$Sen(\psi)(\bigcup_{i\in I} Sen(\phi_i)(E_i)) = \bigcup_{i\in I} Sen(\psi)(Sen(\phi_i)(E_i)) = \bigcup_{i\in I} Sen(\phi_i\psi)(E_i) \subseteq E^\bullet,$$

deducem că $\psi:\ (\Sigma, \bigcup_{i\in I} Sen(\phi_i)(E_i)) \longrightarrow (\Sigma', E)$ este morfism de prezentări. $\Box$

Propoziţie 5.4.2 *Functorul uituc $V:\ Pres(\mathcal{I}) \longrightarrow Sign$ crează slab colimitele.*

Demonstraţie: Fie $F : I \longrightarrow Pres(\mathcal{I})$ un functor. Pentru orice obiect i din I notăm $F(i) = (\Sigma^i, E_i)$.

Fie $V(F(i)) \xrightarrow{\phi_i} \Sigma$ o colimită a functorului $F;V : I \longrightarrow Sign$. Probăm că

$$\phi_i : (\Sigma^i, E_i) \longrightarrow (\Sigma, \bigcup_{i\in I} Sen(\phi_i)(E_i))$$

este o colimită a functorului F. Evident ϕ_i sunt morfisme de prezentări.

Deoarece $V(F(i)) \xrightarrow{\phi_i} \Sigma$ este o colimită a functorului $F;V : I \longrightarrow Sign$ deducem că

$$F(i) \xrightarrow{\phi_i} (\Sigma, \bigcup_{i\in I} Sen(\phi_i)(E_i))$$

este un cocon pentru functorul F.

Fie $F(i) \xrightarrow{\rho_i} (\Sigma', E)$ un cocon pentru functorul F. Deoarece $V(F(i)) \xrightarrow{\rho_i} \Sigma'$ este un cocon pentru functorul $F;V$ şi $V(F(i)) \xrightarrow{\phi_i} \Sigma$ o colimită a functorului $F;V : I \longrightarrow Sign$ există un unic morfism de signaturi $\psi :\ \Sigma \longrightarrow \Sigma'$ cu proprietate $\phi_i;\psi = \rho_i$ pentru orice obiect i din I. Deoarece $\phi_i;\psi = \rho_i$ sunt morfisme de prezentări, din propoziţia precedentă rezultă că ψ este morfism de prezentări. $\square$

Corolar 5.4.3 *Într-o instituţie în care categoria signaturilor are colimite, categoria prezentărilor are colimite şi functorul V comută cu ele.*

Fie $Th(\mathcal{I})$ categoria teoriilor instituţiei $\mathcal{I}$. Functorul uituc

$$U :\ Th(\mathcal{I}) \longrightarrow Sign$$

este prin definiţie restricţia lui V la categoria teoriilor lui $\mathcal{I}$.

Propoziţie 5.4.4 *Fie $\{(\Sigma_i, E_i)\}_{i\in I}$ o familie de teorii şi $\phi_i :\ \Sigma_i \longrightarrow \Sigma$ o familie de morfisme de signaturi. Familia de morfisme de teorii*

$$\phi_i :\ (\Sigma_i, E_i) \longrightarrow (\Sigma, [\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet)$$

are următoarea proprietate:

pentru orice teorie (Σ', E) şi pentru orice morfism de signaturi $\psi :\ \Sigma \longrightarrow \Sigma'$ pentru care

$$\phi_i\psi :\ (\Sigma_i, E_i) \longrightarrow (\Sigma', E)$$

sunt morfisme de teorii, $\psi :\ (\Sigma, [\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet) \longrightarrow (\Sigma', E)$ este morfism de teorii.

Demonstraţie: Deoarece

$$\phi_i\psi :\ (\Sigma_i, E_i) \longrightarrow (\Sigma', E)$$

sunt morfisme de teori, $Sen(\phi_i\psi)(E_i) \subseteq E$ pentru orice $i \in I$. Prin urmare

$$[\bigcup_{i\in I} Sen(\phi_i\psi)(E_i)]^\bullet \subseteq E$$

deoarece E este închisă. Conform lemei închiderii

$$Sen(\psi)([\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet) \subseteq [Sen(\psi)(\bigcup_{i\in I} Sen(\phi_i)(E_i))]^\bullet = [\bigcup_{i\in I} Sen(\psi)(Sen(\phi_i)(E_i))]^\bullet.$$

Prin urmare

$$Sen(\psi)([\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet) \subseteq [\bigcup_{i\in I} Sen(\phi_i\psi)(E_i)]^\bullet \subseteq E.$$

Deducem că $\psi : \ (\Sigma, [\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet) \longrightarrow (\Sigma', E)$ este morfism de teorii. □

Propoziţie 5.4.5 *Functorul uituc $U : \ Th(\mathcal{I}) \longrightarrow Sign$ crează slab colimitele.*

Demonstraţie: Fie $F : I \longrightarrow Th(\mathcal{I})$ un functor. Pentru orice obiect i din I notăm $F(i) = (\Sigma^i, E_i)$.

Fie $U(F(i)) \xrightarrow{\phi_i} \Sigma$ o colimită a functorului $F;U : I \longrightarrow Sign$. Probăm că

$$\phi_i : (\Sigma^i, E_i) \longrightarrow (\Sigma, [\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet)$$

este o colimită a functorului F. Evident ϕ_i sunt morfisme de teori.

Deoarece $V(F(i)) \xrightarrow{\phi_i} \Sigma$ este o colimită a functorului $F;V : I \longrightarrow Sign$ deducem că

$$F(i) \xrightarrow{\phi_i} (\Sigma, [\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet)$$

este un cocon pentru functorul F.

Fie $F(i) \xrightarrow{\rho_i} (\Sigma', E)$ un cocon pentru functorul F. Deoarece $U(F(i)) \xrightarrow{\rho_i} \Sigma'$ este un cocon pentru functorul $F;U$ şi $U(F(i)) \xrightarrow{\phi_i} \Sigma$ o colimită a functorului $F;U : I \longrightarrow Sign$ există un unic morfism de signaturiri $\psi : \ \Sigma \longrightarrow \Sigma'$ cu proprietate $\phi_i;\psi = \rho_i$ pentru orice obiect i din I. Deoarece $\phi_i;\psi = \rho_i$ sunt morfisme de teorii, din propoziţia precedentă rezultă că ψ este morfism de teorii. □

Corolar 5.4.6 *Într-o instituţie în care categoria signaturilor are colimite, categoria teoriilor are colimite şi functorul U comută cu ele.*

Din teoremele de mai sus rezultă că într-o instituţie în care categoria signaturilor are colimite, atât categoria prezentărilor ei cât şi categoria teoriilor ei au colimite. Mai mult propoziţiile de mai sus dau şi o construcţie a acestora.

5.4.1 Functorul $\bullet$

Observaţia 5.4.7 *Dacă $\psi : (\Sigma, E) \longrightarrow (\Sigma_1, E_1)$ este morfism de prezentări,*

atunci $\psi : (\Sigma, E^\bullet) \longrightarrow (\Sigma_1, E_1^\bullet)$ este morfism de teorii.

Demonstraţie: Din lema închiderii deducem $Sen(\psi)(E^\bullet) \subseteq Sen(\psi)(E)^\bullet$. Deoarece ψ este morfism de prezentări $Sen(\psi)(E) \subseteq E_1^\bullet$ prin urmare $Sen(\psi)(E)^\bullet \subseteq E_1^\bullet$ deci $Sen(\psi)(E^\bullet) \subseteq E_1^\bullet$, adică
$\psi : (\Sigma, E^\bullet) \longrightarrow (\Sigma_1, E_1^\bullet)$ este morfism de teorii.

$$\begin{tikzcd}
Pres(\mathcal{I}) \arrow[rr, "\bullet"] \arrow[dr, "V"'] & & Th(\mathcal{I}) \arrow[dl, "U"] \\
& Sign &
\end{tikzcd}$$

Functorul $\bullet : \ Pres(\mathcal{I}) \longrightarrow Th(\mathcal{I})$ este definit prin

1. $(\Sigma, E)^\bullet = (\Sigma, E^\bullet)$ pentru orice prezentare (Σ, E).
2. $\psi^\bullet = \psi$ pentru orice morfism de prezentări ψ.

Observăm că $\bullet\, ; U = V$.

Observaţia 5.4.8 *Dacă $\phi_i : \Sigma_i \longrightarrow \Sigma$ este o familie de morfisme de signaturi şi $E_i \subseteq Sen(\Sigma_i)$ pentru orice indice $i \in I$, atunci*

$$[\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet = [\bigcup_{i\in I} Sen(\phi_i)(E_i^\bullet)]^\bullet$$

Demonstraţie: *Demonstraţia prin dublă incluziune incepe prin a observa că incluziunea*

$$[\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet \subseteq [\bigcup_{i\in I} Sen(\phi_i)(E_i^\bullet)]^\bullet$$

rezultă uşor plecând de la $E_i \subseteq E_i^\bullet$ şi observând că toate construcţiile folosite sunt crescătoare.

Reciproc, utilizând lema incluziunii

$$Sen(\phi_i)(E_i^\bullet) \subseteq [Sen(\phi_i)(E_i)]^\bullet$$

deducem succesiv

$$Sen(\phi_i)(E_i^\bullet) \subseteq [\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet,$$

$$\bigcup_{i\in I} Sen(\phi_i)(E_i^\bullet) \subseteq [\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet \text{ şi}$$

$$[\bigcup_{i\in I} Sen(\phi_i)(E_i^\bullet)]^\bullet \subseteq [\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet. \ \Box$$

Propoziţie 5.4.9 *Într-o instituţie în care categoria signaturilor are colimite functorul $\bullet$ conservă colimitele.*

Demonstraţie: *Fie $F : I \longrightarrow Pres(\mathcal{I})$ un functor. Pentru orice obiect i din I notăm $F(i) = (\Sigma^i, E_i)$.*

Pentru a demonstra că functorul $\bullet$ conserva colimitele functorului F vom utiliza observaţia 5.1.14 şi vom arăta că functorul $\bullet$ conservă o anumită colimită a functorului F. Mai precis vom arăta că functorul $\bullet$ duce colimita functorului F construită în conformitate cu propoziţia 5.4.2 în colimita functorului $F; \bullet$ construită conform propoziţiei 5.4.5.

Fie $V(F(i)) \xrightarrow{\phi_i} \Sigma$ *o colimită a functorului* $F;V : I \longrightarrow Sign$. *Conform construcţiei din propoziţia 5.4.2*

$$\phi_i : (\Sigma^i, E_i) \longrightarrow (\Sigma, \bigcup_{i\in I} Sen(\phi_i)(E_i)) \tag{5.18}$$

este o colimită a functorului F *şi deoarece* $(F;\bullet)(i) = (\Sigma^i, E_i^\bullet)$ *pentru orice obiect* i *din* I, *conform construcţiei din propoziţia 5.4.5*

$$\phi_i : (\Sigma^i, E_i) \longrightarrow (\Sigma, [\bigcup_{i\in I} Sen(\phi_i)(E_i^\bullet)]^\bullet) \tag{5.19}$$

este o colimită a functorului $F;\bullet$.

Egalitatea demonstrată mai sus

$$[\bigcup_{i\in I} Sen(\phi_i)(E_i)]^\bullet = [\bigcup_{i\in I} Sen(\phi_i)(E_i^\bullet)]^\bullet$$

arată că aplicând functorul $\bullet$ *colimitei 5.18 a functorului* F *se obţine colimita 5.19 a functorului* $F;\bullet$. $\square$

Practic în calculator nu putem lucra decât cu prezentări finite care reprezintă teorii. Se stie că în procesul de crearea de programe mari din îmbinarea unor programe mai mici(modularizare) colimitele joacă un rol deosebit. Pentru a fi convinşi că lucrurile sunt corecte este necesar să arătăm următoarele: colimita prezentărilor reprezintă chiar teoria colimită a teoriilor reprezentate de prezentarile care au intrat în calcul. Ultima propoziţie ne asigură de acest fapt.

5.5 CATEGORII INCLUSIVE

Conceptul de incluziune abstractă a fost introdus de Diaconescu, Goguen şi Stefaneas [6], unde definiţia categoriei inclusive cere mai multe condiţii. Definiţie de mai jos, sub numele de categorie slab inclusivă, a fost introdusă de Căzănescu şi Roşu [3]. Definiţia utilizată aici coincide cu cea din cartea lui Răzvan Diaconescu [4].

Definiţia 5.5.1 O categorie $\mathcal{I}$ este numită **categorie de incluziuni** dacă pentru orice pereche de obiecte A şi B

1) $\mathcal{I}(A,B)$ are cel mult un element şi

2) $\mathcal{I}(A,B) \neq \emptyset$ şi $\mathcal{I}(B,A) \neq \emptyset$ implică $A = B$. **antisimetrie**

Morfismele din $\mathcal{I}$ vor fi numite **incluziuni**. $\square$

Se observă că o categorie de incluziuni nu este altceva decât o clasă parţial ordonată organizată ca o categorie.

Pentru incluziuni $i \in \mathcal{I}(A,B)$ mai utilizăm şi notaţia $i : A \hookrightarrow B$.

Definiţia 5.5.2 O **subcategorie de incluziuni** a categoriei $\mathcal{C}$ este o categorie de incluziuni care este o subcategorie a lui $\mathcal{C}$ având aceleaşi obiecte ca şi $\mathcal{C}$.

Definiţia 5.5.3 Spunem că $(\mathcal{I}, \mathcal{E})$ este un **sistem de incluziuni** pentru categoria $\mathcal{C}$ dacă

1. $\mathcal{I}$ este o subcategorie de incluziuni a lui $\mathcal{C}$,
2. $\mathcal{E}$ este subcategorie a lui $\mathcal{C}$ care are aceleaşi obiecte ca şi $\mathcal{C}$ şi
3. Orice morfism f din $\mathcal{C}$ poate fi factorizat unic $f = e_f; i_f$ unde $e_f \in \mathcal{E}$ şi $i_f \in \mathcal{I}$.

O categorie împreună cu un sistem de incluziuni se numeşte **categorie inclusivă.** $\square$

Fie $f : A \longrightarrow B$ un morfism dintr-o categorie inclusivă. Imaginea lui f notată cu $f(A)$ este prin definiţie codomeniul morfismului $e_f : A \longrightarrow f(A)$ acelaşi cu domeniul morfismului $i_f : f(A) \hookrightarrow B$.

Continuăm cu mai multe exemple.

Set

Considerăm categoria mulţimilor **Set** cu sistemul de inclusiuni **(Incluziuni,Surjecţii)**. Prin incluziune de mulţimi se inţelege unica funcţie de la o submulţime la o supramulţime a sa care duce orice element în el însuşi. Subcategoria incluziunilor este o latice unde *supremum* e reuniunea şi $infimum$ este intersecţia.

Este binecunoscut că orice funcţie $f : A \longrightarrow B$ se poate scrie ca o compunere $f = e_f; i_f$ unde $i_f : f(A) \hookrightarrow B$ este incluziune şi surjecţia $e_f : A \longrightarrow f(A)$ este definită prin $e_f(a) = f(a)$ pentru orice $a \in A$.

Fie $f = e; i$ este o altă descompunere a lui f într-o surjecţie $e : A \longrightarrow C$ şi o incluziune $i : C \hookrightarrow B$. Evident $C \subseteq B$. Observăm că pentru orice $a \in A$, $f(a) = i(e(a)) = e(a) \in C$, prin urmare $f(A) \subseteq C$. Probăm incluziunea contrară. Fie $c \in C$. Deoarece e este surjectivă există $a \in A$ astfel îmcât $c = e(a) = i(e(a)) = f(a)$, deci $c \in f(A)$. Deoarece $f(A) = C$ deducem uşor că $e = e_f$ şi $i = i_f$.

Sig

Definiţia 5.5.4 Fie două signaturi algebrice multisortate $(S, \Sigma), (S', \Sigma') \in |\mathbf{Sig}|$. Prin definiţie

$(S, \Sigma) \subseteq (S', \Sigma')$ dacă şi numai dacă $S \subseteq S'$ şi $\Sigma_{w,s} \subseteq \Sigma'_{w,s}$ pentru orice $w \in S^*, s \in S$.

Cu relaţia astfel definită signaturile algebrice formează o clasă parţial ordonată.

Incluziunile lui **Sig** sunt definite prin morfismele care au toate componentele incluziuni de mulţimi. Notăm cu $\mathcal{I}$ categoria incluziunilor de signaturi. Ea este o subcategorie de incluziuni a lui **Sig**.

Evident că pentru orice pereche de signaturi $(S, \Sigma), (S', \Sigma')$ şi pentru orice pereche de incluziuni $(f_1, g_1), (f_2, g_2)$ din $\mathcal{I}((S, \Sigma), (S', \Sigma'))$ deoarece funcţiile $f_1, f_2 : S \hookrightarrow S'$, sunt incluziuni rezultă că $f_1 = f_2$. De asemenea, pentru orice $w \in S^*$ şi $s \in S$ funcţiile $g_{1w,s}, g_{2w,s} : \Sigma_{w,s} \hookrightarrow \Sigma'_{f^*(w),f(s)}$, unde $f = f_1$, sunt incluziuni. Din faptul că $g_{1_{w,s}}$ şi $g_{2_{w,s}}$ sunt incluziuni rezultă $g_{1_{w,s}} = g_{2_{w,s}}$. Deci $(f_1, g_1) = (f_2, g_2)$.

Mai observăm că $(S, \Sigma) \subseteq (S', \Sigma')$ şi $(S', \Sigma') \subseteq (S, \Sigma)$ implică $(S, \Sigma) = (S', \Sigma')$. Prin urmare, $\mathcal{I}$ este subcategorie de incluziuni a lui **Sig**.

Definiţia 5.5.5 Un morfism de signaturi $(f,g) : (S',\Sigma') \longrightarrow (S,\Sigma)$ se numeşte *surjecţie abstractă de signaturi* dacă $f : S' \longrightarrow S$ este o funcţie surjectivă şi pentru orice $w \in S^*$, $s \in S$ şi $\sigma \in \Sigma_{w,s}$ există $w' \in S'^*$, $s' \in S'$ şi $\sigma' \in \Sigma'_{w',s'}$ cu proprietăţile $f^*(w') = w$, $f(s') = s$ şi $g_{w',s'}(\sigma') = \sigma$.

Notăm cu $\mathcal{E}$ categoria surjecţiilor abstracte de signaturi.

Observaţia 5.5.6 *$\mathcal{E}$ este subcategorie a lui* **Sig** *care are aceleaşi obiecte ca şi* **Sig**.

Demonstraţie: Fie $(f',g') : (S'',\Sigma'') \longrightarrow (S',\Sigma')$ şi $(f,g) : (S',\Sigma') \longrightarrow (S,\Sigma)$ două surjecţii abstracte de signaturi. Vom arăta că şi compunerea lor este o surjecţie abstractă de signaturi.

Deoarece f' şi f sunt surjecţii rezultă că $f';f$ este surjecţie.

Fie $w \in S^*$, $s \in S$ şi $\sigma \in \Sigma_{w,s}$. Deoarece $(f,g) : (S',\Sigma') \longrightarrow (S,\Sigma)$ este surjecţie abstractă de signaturi există $w' \in S'^*$, $s' \in S'$ şi $\sigma' \in \Sigma'_{w',s'}$ cu proprietăţile $f^*(w') = w$, $f(s') = s$ şi $g_{w',s'}(\sigma') = \sigma$. Deoarece $(f',g') : (S'',\Sigma'') \longrightarrow (S',\Sigma')$ este surjecţie abstractă de signaturi există $w'' \in S''^*$, $s'' \in S''$ şi $\sigma'' \in \Sigma''_{w'',s''}$ cu proprietăţile $f'^*(w'') = w'$, $f'(s'') = s'$ şi $g'_{w'',s''}(\sigma'') = \sigma'$. Rezultă că
$(f';f)^*(w'') = f^*(f'^*(w'')) = f^*(w') = w$, $(f';f)(s'') = f(f'(s'')) = f(s') = s$ şi
$(g';g)_{w'',s''}(\sigma'') = (g'_{w'',s''};g_{f'^*(w''),f'(s'')})(\sigma'') = g_{w',s'}(g'_{w'',s''}(\sigma'')) = g_{w',s'}(\sigma') = \sigma$.

Deci $(f',g');(f,g)$ este surjecţie abstractă de signaturi.

Mai observăm că orice morfism identitate de signaturi este o surjecţie abstractă de signaturi.

Propoziţie 5.5.7 *În categoria signaturilor algebrice* **Sig** $<\mathcal{I},\mathcal{E}>$ *este sistem inclusiv.*

Demonstraţie: Demonstrăm că orice morfism de signaturi (f,g) poate fi scris unic ca o compunere între o surjecţie abstractă de signaturi (e_f,e_g) din $\mathcal{E}$ şi o incluziune (i_f,i_g) din $\mathcal{I}$. Căutăm o descompunere de tipul:

$$\begin{tikzcd}
(S,\Sigma) \arrow[rr, "{(f,g)}"] \arrow[dr, "{(e_f,e_g)}"'] & & (S',\Sigma') \\
& (f(S),\{\Sigma^0_{w,s}\}_{w\in f(S)^*,s\in f(S)}) \arrow[ur, "{(i_f,i_g)}"'] &
\end{tikzcd}$$

Demonstrăm existenţa descompunerii.

În categoria **Set**, pentru orice funcţie f există o unică descompunere cu e_f surjecţie şi i_f incluziune astfel încât $f = e_f;i_f$ de forma :

$$\begin{tikzcd}
S \arrow[rr, "f"] \arrow[dr, "e_f"'] & & S' \\
& f(S) \arrow[ur, "i_f"'] &
\end{tikzcd}$$

Ştim că $e_f(x) = f(x)$ pentru orice $x \in S$.

Pentru orice $w' \in f(S)^*$ şi $s' \in f(S)$ definim

$$\Sigma^0_{w',s'} = \bigcup\{g_{w,s}(\Sigma_{w,s}) | w \in (f^*)^{-1}(w') \text{ şi } s \in f^{-1}(s')\} \subseteq \Sigma'_{w',s'}.$$

Pentru orice $w \in S^*, s \in S$ definim

$$e_{g_{w,s}} : \Sigma_{w,s} \longrightarrow \Sigma^0_{e_f(w),e_f(s)}$$

astfel $e_{g_{w,s}}(\sigma) = g_{w,s}(\sigma) \in \Sigma^0_{f^*(w),f(s)}$, pentru fiecare $\sigma \in \Sigma_{w,s}$.

Observăm că $(e_f, e_g) \in \mathcal{E}$, $(i_f, i_g) \in \mathcal{I}$ şi $(f, g) = (e_f, e_g); (i_f, i_g)$.

Arătăm unicitatea acestei descompuneri.
Presupunem că există $(f', g') \in \mathcal{E}$ şi o incluziune (i'_f, i'_g) de signaturi în $\mathcal{I}$ astfel încât diagrama următoare să fie comutativă:

$$\begin{tikzcd}
(S,\Sigma) \arrow[rr, "{(f,g)}"] \arrow[dr, "{(f',g')}"'] & & (S',\Sigma') \\
& (S'',\Sigma'') \arrow[ur, "{(i'_f,i'_g)}"'] &
\end{tikzcd}$$

Dar, în categoria mulţimilor **Set**, e_f şi f' sunt ambele surjecţii. Cum o astfel de descompunere $f = f'; i'_f = e_f; i_f$ ese unică în **Set** înseamnă că $f' = e_f$, $i'_f = i_f$ şi $S'' = f(S)$.

Pentru orice $w \in S^*$, $s \in S$ şi $\sigma \in \Sigma_{w,s}$ observăm că

$$g'_{w,s}(\sigma) = (i'_g)_{f'^*(w), f'(s)}(g'_{w,s}(\sigma)) = (g'_{w,s}; (i'_g)_{f'^*(w), f'(s)})(\sigma) = g_{w,s}(\sigma) = (e_g)_{w,s}(\sigma).$$

Deoarece (f', g') este surjecţie abstractă pentru orice $w' \in f^*(S)$ şi $s' \in f(S)$

$$\Sigma''_{w',s'} = \bigcup \{g'_{w,s}(\Sigma_{w,s}) |\ s \in f^{-1}(s')\ , w \in (f^*)^{-1}(w')\} =$$

$$\bigcup \{g_{w,s}(\Sigma_{w,s}) |\ s \in f^{-1}(s')\ , w \in (f^*)^{-1}(w')\} = \Sigma^0_{w',s'}.$$

Deci $g' = e_g$ şi $\Sigma'' = \Sigma^0$.

Deci **Sig** cu sistemul $< \mathcal{I}, \mathcal{E} >$ este categorie inclusivă. $\Box$

Teoriile unei instituţii

Fie o instituţie $\mathcal{I}$. Fie $U : Th(\mathcal{I}) \longrightarrow Sign$ functorul uituc definit prin $U(\Sigma, E) = \Sigma$ şi $U(\varphi) = \varphi$.

Reamintim că o prezentare este o pereche (S, Σ) unde $\Sigma \in |Sign|$ şi $E \subseteq Sen(\Sigma)$ şi o teorie este o prezentare (S, Σ) cu $E^\bullet = E$.

Teorema 5.5.8 *Intr-o instituţie $\mathcal{I}$, dacă $Sign$ este o categorie inclusivă, atunci $Th(\mathcal{I})$ este o categorie inclusivă.*

Demonstraţie: Fie $(I, \mathcal{E})$ un sistem de incluziuni în categoria signaturilor $Sign$. Vom defini $(\mathcal{I}_\mathcal{T}, \mathcal{E}_\mathcal{T})$, un sistem de incluziuni în categoria teoriilor după cum urmează.

Fie $\psi : (\Sigma, E) \longrightarrow (\Sigma', E')$ un morfism de teorii.
1) $\psi \in \mathcal{I}_\mathcal{T}$ dacă şi numai dacă $\psi \in I$
2) $\psi \in \mathcal{E}_\mathcal{T}$ dacă şi numai dacă $\psi \in \mathcal{E}$ şi $E' = Sen(\psi)(E)^\bullet$.

Probăm că $\mathcal{I}_\mathcal{T}$ este o categorie de incluziuni.
1) Deoarece

$$\mathcal{I}_\mathcal{T}((\Sigma, E), (\Sigma', E')) \subseteq I(\Sigma, \Sigma')$$

şi $I(\Sigma, \Sigma')$ are cel mult un element deducem că $\mathcal{I}_\mathcal{T}((\Sigma, E), (\Sigma', E'))$ are cel mult un element.
2) Fie $\psi \in \mathcal{I}((\Sigma, E), (\Sigma', E'))$ şi $\varphi \in \mathcal{I}((\Sigma, E), (\Sigma', E'))$. Deducem $I(\Sigma, \Sigma') \neq \emptyset$ şi $I(\Sigma, \Sigma')) \neq \emptyset$, deci $\Sigma = \Sigma'$. Din $\psi \in \mathcal{I}(\Sigma, \Sigma)$ şi $\varphi \in \mathcal{I}(\Sigma, \Sigma)$ deducem $\psi = \varphi = 1_\Sigma$. Deoarece $1_\Sigma : (\Sigma, E) \longrightarrow (\Sigma, E')$ şi $1_\Sigma : (\Sigma, E') \longrightarrow (\Sigma, E)$ sunt morfisme de teorii deducem $E = E'$. Deci $(\Sigma, E) = (\Sigma', E')$.

Deoarece pentru orice teorie (Σ, E), $1_\Sigma : (\Sigma, E) \longrightarrow (\Sigma, E)$ este morfism de teorii din $\mathcal{I}_\mathcal{T}$ rezultă că $\mathcal{I}_\mathcal{T}$ are aceleaşi obiecte ca şi $Th(\mathcal{I})$ deci $\mathcal{I}_\mathcal{T}$ este o subcategorie de incluziuni a lui $Th(\mathcal{I})$.

Fie $\psi \in \mathcal{E}_\mathcal{T}((\Sigma, E), (\Sigma_1, E_1))$ şi $\varphi \in \mathcal{E}_\mathcal{T}((\Sigma_1, E_1), (\Sigma_2, E_2))$. Probăm că $\psi; \varphi \in \mathcal{E}_\mathcal{T}((\Sigma, E), (\Sigma_2, E_2))$. Din ipoteză $E_1 = Sen(\varphi)(E)^\bullet$ şi $E_2 = Sen(\varphi)(E_1)^\bullet$. Folosind observaţia 5.4.8 deducem

$$Sen(\varphi)(Sen(\psi)(E)^\bullet)^\bullet = (Sen(\varphi)(Sen(\psi)(E)))^\bullet$$

Prin urmare

$$Sen(\varphi)(E_1)^\bullet = ((Sen(\psi); Sen(\varphi))(E))^\bullet$$

aşadar $E_2 = (Sen(\psi; \varphi)(E))^\bullet$ deci

$$\psi; \varphi \in \mathcal{E}_\mathcal{T}((\Sigma, E), (\Sigma_2, E_2)).$$

Deoarece pentru orice teorie (Σ, E), $1_\Sigma : (\Sigma, E) \longrightarrow (\Sigma, E)$ este morfism de teorii din $\mathcal{E}_\mathcal{T}$ rezultă că $\mathcal{E}_\mathcal{T}$ are aceleaşi obiecte ca şi $Th(\mathcal{I})$.

Probăm existenţa descompunerii. Fie $\psi : (\Sigma, E) \longrightarrow (\Sigma', E')$ un morfism de teorii. Descompunem ψ în categoria inclusivă a signaturilor

$$\begin{tikzcd} \Sigma \arrow[rr, "\psi"] \arrow[dr, "e_\psi"'] & & \Sigma' \\ & \psi(\Sigma) \arrow[ur, "i_\psi"'] & \end{tikzcd}$$

Formăm diagrama de mai jos. Observăm că

$$e_\psi : (\Sigma, E) \longrightarrow (\psi(\Sigma), (Sen(e_\psi)(E))^\bullet)$$

este morfism de teorii din $\mathcal{E}_\mathcal{T}$. Deoarece

$$Sen(i_\psi)((Sen(e_\psi)(E))^\bullet) \subseteq (Sen(i_\psi)[Sen(e_\psi)(E)])^\bullet = (Sen(\psi)(E))^\bullet \subseteq E'^\bullet = E'$$

rezultă că

$$i_\psi : (\psi(\Sigma), (Sen(e_\psi)(E))^\bullet) \longrightarrow (\Sigma', E')$$

este morfism de teorii din $\mathcal{I}_\mathcal{T}$.

$$\begin{tikzcd} (\Sigma, E) \arrow[rr, "\psi"] \arrow[dr, "e_\psi"'] & & (\Sigma', E') \\ & (\psi(\Sigma), (Sen(e_\psi)(E))^\bullet) \arrow[ur, "i_\psi"'] & \end{tikzcd}$$

Probăm unicitatea descompunerii. Fie $\psi = e; i$ o altă descompunere unde e este din $\mathcal{E}_\mathcal{T}$ şi i este din $\mathcal{I}_\mathcal{T}$.

$$\begin{tikzcd} (\Sigma, E) \arrow[rr, "\psi"] \arrow[dr, "e"'] & & (\Sigma', E') \\ & (\Sigma_1, (Sen(e)(E))^\bullet) \arrow[ur, "i"'] & \end{tikzcd}$$

Deoarece descompunerea $\psi = e; i$ are loc şi în categoria inclusivă a signaturilor deducem $\Sigma_1 = \psi(\Sigma)$, $e = e_\psi$ şi $i = i_\psi$ deci descompunerea este unică.

În concluzie $Th(\mathcal{I})$ este o categorie inclusivă. □

Corolar 5.5.9 *Teoriile instituţiei logicii ecuaţionale formează o categorie inclusivă.*

5.5.1 Categorii inclusive cu sume

Definiţia 5.5.10 Spunem că o categorie inclusivă $\mathcal{C}$ are sume dacă subcategoria incluziunilor ei $\mathcal{I}$ are supremuri finite.

Reamintim că întro categorie de incluziuni supremurile coincid cu sumele directe. Continuăm cu mai multe exemple de categorii inclusive cu sumă.

Set

Am văzut că în categoria mulţimilor sumele directe sunt reuniunile disjuncte.

În categoria incluziunilor de mulţimi sumele directe sunt reuniunile. De fapt având de a face cu o clasă parţial ordonată, sumele directe coincid cu supremurile care în cazul mulţimilor coincid cu reuniunile.

Sig

Propoziţie 5.5.11 *Categoria inclusivă a signaturilor algebrice are sume.*

Demonstraţie:

Vom demonstra chiar mai mult, existenţa supremurilor arbitrare.

Considerăm K o mulţime arbitrară de indecşi. Pentru $k \in K$ fie (S_k, Σ^k) o signatură algebrică. Signatura (S, Σ) este definită prin $S = \bigcup\{S_k | k \in K\}$ şi pentru orice $w \in S^*$ şi $s \in S$ prin

$$\Sigma_{w,s} = \bigcup\{\Sigma^k_{w,s} | w \in S^*_k, s \in S_k\}.$$

Evident $(S_k, \Sigma^k) \subseteq (S, \Sigma)$ pentru orice $k \in K$.

Presupunem $(S_k, \Sigma^k) \subseteq (S', \Sigma')$ pentru orice $k \in K$. Deoarece $S_k \subseteq S'$ pentru orice $k \in K$ deducem $S \subseteq S'$.

Fie $w \in S^*$, $s \in S$ şi $\sigma \in \Sigma_{w,s}$. Prin urmare există $k \in K$, $w \in S^*_k$ şi $s \in S_k$ cu proprietatea $\sigma \in \Sigma^k_{w,s}$. Deoarece $(S_k, \Sigma^k) \subseteq (S', \Sigma')$ deducem $\sigma \in \Sigma'_{w,s}$. Am arătat astfel $\Sigma \subseteq \Sigma'$.

În concluzie, (S', Σ') este supremul familiei (S_k, Σ^k). $\Box$

Teoriile unei instituţii

Teorema 5.5.12 *Dacă $Sign$ este o categorie inclusivă cu sume, atunci $Th(\mathcal{I})$ este o categorie inclusivă cu sume.*

Demonstraţie: Păstrăm notaţiile din propoziţia precedentă.

Fie (Σ_k, E_k) o familie finită de teorii. Fie Σ supremul(suma directă) familiei $\{\Sigma_k\}_k$ şi $i_k : \Sigma_k \hookrightarrow \Sigma$ incluziuni. Vom proba că

$$(\Sigma, (\bigcup_k Sen(i_k)(E_k))^\bullet)$$

este supremum în categoria incluziunilor de teorii a familiei (Σ_k, E_k). Evident i_k sunt morfisme de teorii,

Fie (Σ_1, E_1) o teorie şi $j_k \in \mathcal{I}_\mathcal{T}((\Sigma_k, E_k), (\Sigma_1, E_1))$. Deoarece Σ este supremul familiei de signaturi $\{\Sigma_k\}_k$, există o unică incluziune de signaturi $j : \Sigma \hookrightarrow \Sigma_1$ cu proprietatea $i_k; j = j_k$ pentru orice k.

Deoarece familia

$$i_k : (\Sigma_k, E_k) \longrightarrow (\Sigma, (\bigcup_k Sen(i_k)(E_k))^\bullet)$$

are proprietatea din propoziţia 5.4.4 şi $i_k; j$ sunt morfisme de teorii rezultă că

$$j : (\Sigma, (\bigcup_k Sen(i_k)(E_k))^\bullet) \longrightarrow (\Sigma_1, E_1)$$

este morfism de teorii. □

Deoarece categoria signaturilor algebrice Sig este inclusivă cu sume rezultă corolarul următor.

Corolar 5.5.13 *Categoria teoriilor din instituţia logicii ecuaţionale este inclusivă cu sume.*

Din punct de vedere practic, deoarece teoriile sunt implementate prin prezentări ale lor, este utilă următoarea propoziţie.

Propoziţie 5.5.14 *Fie* (Σ_k, E_k) *o familie finită de prezentări şi* $i_k : \Sigma_k \longrightarrow \Sigma$ *supremul în categoria incluziunilor de signaturi. Supremul familiei de teorii* $(\Sigma_k, E_k^\bullet)$ *este prezentat de*

$$(\Sigma, \bigcup_k Sen(i_k)(E_k)).$$

Demonstraţie: Este suficient să invocăm observaţia 5.4.8

$$(\bigcup_k Sen(i_k)(E_k))^\bullet = (\bigcup_k Sen(i_k)(E_k^\bullet))^\bullet.$$

5.5.2 Sume fibrate care conservă incluziunile

Sumele fibrate sunt un caz particular de colimite. Fie I o categorie mică care are trei obiecte împreună cu identităţile lor şi alte două morfisme cu acelaşi sursă. Un functor definit pe I cu valori întro categorie $\mathcal{C}$ este dat de două morfisme cu aceeaşi sursă din categoria $\mathcal{C}$.

$$\begin{tikzcd} & B \\ A \arrow[ur, "i"] \arrow[dr, "f"'] & \\ & C \end{tikzcd}$$

Suma fibrată este colimita unui astfel de functor.

Un cocon pentru un astfel de functor este format din două morfisme

$$j : B \longrightarrow D \text{ şi } g : C \longrightarrow D$$

cu proprietatea

$$i; j = f; g.$$

Urmând direct definiţia coconului ar fi trebuit trei morfisme j, g de mai sus şi $h : A \longrightarrow D$ cu proprietăţile $h = i; j$ şi $h = f; g$. Se observă uşor echivalenţa celor două variante.

Definiţia 5.5.15 Se numeşte **sumă fibrată** a cuplului de morfisme

$$\begin{tikzcd} & B \\ A \arrow[ur, "i"] \arrow[dr, "f"'] & \\ & C \end{tikzcd}$$

un cuplu de morfisme

$$j : B \longrightarrow D \text{ şi } g : C \longrightarrow D$$

cu proprietăţile:

1) $i; j = f; g$ şi

2) pentru orice cuplu de morfisme $j' : B \longrightarrow D'$ şi $g' : C \longrightarrow D'$ cu proprietatea $i; j' = f; g'$ există un unic morfism $u : D \longrightarrow D'$ care satisface

$$j; u = j' \text{ şi } g; u = g'. \square$$

Definiţia 5.5.16 Într-o categorie inclusivă, spunem că sumele fibrate conservă incluziunile dacă:

pentru orice incluziune $i : A \hookrightarrow B$ şi orice morfism $f : A \longrightarrow C$ pentru care perechea i, f are o sumă fibrată, există o sumă fibrată a perechii i, f formată dintr-un morfism $g : B \longrightarrow D$ şi o incluziune $j : C \longrightarrow D$.

$$\begin{tikzcd} & B \arrow[dr, "g"] & \\ A \arrow[ur, "i"] \arrow[dr, "f"'] & & D \\ & C \arrow[ur, hook, "j"'] & \end{tikzcd}$$

Continuăm cu exemple de categorii inclusive în care sumele fibrate conservă incluziunile.

Set

Propoziţie 5.5.17 *În categoria mulţimilor sumele fibrate conservă incluziunile.*

Demonstraţie: Fie $f : A \longrightarrow C$ o funcţie şi $A \subseteq B$. Notăm cu $i : A \hookrightarrow B$ incluziunea lui A în B.

$$\begin{tikzcd}
 & B \arrow[d, "g"] \arrow[drr, "u"] & & \\
A \arrow[ur, "i"] \arrow[dr, "f"'] & C\coprod(B-A) \arrow[rr, "w"'] & & D \\
 & C \arrow[u, hook, "j"'] \arrow[urr, "v"'] & &
\end{tikzcd}$$

Considerăm reuniunea disjunctă $C\coprod(B-A)$ şi definim $g : B \longrightarrow C\coprod(B-A)$ ca o extindere a lui f astfel $g(b) = \begin{cases} b & b \in B - A \\ f(b) & \text{altfel} \end{cases}$.
Observăm că $g(i(a)) = g(a) = f(a) = j(f(a))$, adică $i; g = f; j$.

Considerăm $u : B \longrightarrow D$ şi $v : C \longrightarrow D$ astfel încât $i; u = f; v$. Definim $w : C\coprod(B-A) \longrightarrow D$ astfel

$$w(x) = \begin{cases} v(x) & x \in C \\ u(x) & x \in B - A \end{cases}$$

Prin urmare pentru orice $b \in B$

$$w(g(b)) = \begin{cases} w(b) & b \in B - A \\ w(f(b)) & b \in A \end{cases} = \begin{cases} u(b) & b \in B - A \\ v(f(b)) & b \in A \end{cases} = u(b)$$

deci $g; w = u$. Deoarece pentru orice $x \in C$, $w(j(x)) = w(x) = v(x)$, deducem $j; w = v$.

Unicitatea lui w rezultă din construcţie. $\square$

Sig

Propoziţie 5.5.18 *În categoria signaturilor algebrice sumele fibrate conservă incluziunile.*

Demonstraţie: Fie $i : (S, \Sigma) \longrightarrow (S', \Sigma')$ o incluziune de signaturi şi $(f, g) : (S, \Sigma) \longrightarrow (S'', \Sigma'')$ un morfism de signaturi.

Plecând de la incluziunea de mulţimi $S \hookrightarrow S'$ şi funcţia $f : S \longrightarrow S''$ facem în categoria mulţimilor, după modelul de mai sus, o sumš fibrată care conservă incluziunea.

Definim f' astfel $f'(s) = \begin{cases} f(s) & s \in S \\ s & \text{altfel} \end{cases}$. Observăm că $f'\restriction_S = f; l$.

Variante de definire Σ^0:

1) Dacă $w^0 \in S''^*$ şi $s^0 \in S''$ atunci

$$\Sigma^0_{w^0,s^0} = \Sigma''_{w^0,s^0} \cup \{(w', s', \sigma') | f'^*(w') = w^0,\ f'(s') = s^0,\ \sigma' \in \Sigma'_{w',s'} - \Sigma_{w',s'}\}.$$

2) În caz contrar

$$\Sigma^0_{w^0,s^0} = \{(w', s', \sigma') | f'^*(w') = w^0,\ f'(s') = s^0,\ \sigma' \in \Sigma'_{w',s'} - \Sigma_{w',s'}\}.$$

Vom defini g' pentru orice $w' \in S'^{*}$, $s' \in S'$ şi $\sigma \in \Sigma'_{w',s'}$ prin

$$g'_{w',s'}(\sigma) = \begin{cases} g_{w',s'}(\sigma) & w' \in S^*, s' \in S, \sigma \in \Sigma_{w',s'} \\ (w', s', \sigma) & \text{altfel} \end{cases}$$

Observăm că $g'_{w',s'} : \Sigma'_{w',s'} \longrightarrow \Sigma^0_{f'^*(w'),f'(s')}$. Comutativitatea diagramei $i; (f', g') = (f, g); l$ este evidentă.

$$\begin{tikzcd}
 & (S',\Sigma') \arrow[dr, "(f',g')"] \arrow[drr, "(f_1,g^1)"] & & \\
(S,\Sigma) \arrow[ur, "i"] \arrow[dr, "(f,g)"'] & & (S''\coprod(S'-S),\Sigma^0) \arrow[r, dashed, "\exists!(u,v)"'] & (S_1,\Sigma_1) \\
 & (S'',\Sigma'') \arrow[ur, hook, "l"'] \arrow[urr, "(f_2,g^2)"'] & &
\end{tikzcd}$$

Considerăm morfismele $(f_1, g^1) : (S', \Sigma') \longrightarrow (S^1, \Sigma^1)$, $(f_2, g^2) : (S'', \Sigma'') \longrightarrow (S^1, \Sigma^1)$ cu proprietatea $i; (f_1, g^1) = (f, g); (f_2, g^2)$, prin urmare $f_1{\restriction_\Sigma} = f; f_2$ şi pentru orice $w \in S^*$, $s \in S$ şi $\sigma \in \Sigma_{w,s}$

$$g^1_{w,s}(\sigma) = g^2_{f^*(w),f(s)}(g_{w,s}(\sigma)).$$

Construim morfismul de signaturi $(u, v) : (S'' \coprod(S' - S), \Sigma^0) \longrightarrow (S^1, \Sigma^1)$. Din proprietatea sumei fibrate pentru mulţimi ştim că există o unică funcţie u care face diagramele comutative pe mulţimi, adică $f'; u = f_1$ şi $u{\restriction_{S''}} = f_2$.

Definim pe $v_{w^0,s^0} : \Sigma^0_{w^0,s^0} \longrightarrow \Sigma^1_{u^*(w^0),u(s^0)}$ astfel

1) Dacă $w^0 \in S''^{*}, s^0 \in S'', \sigma \in \Sigma''_{w^0,s^0}$ atunci $v_{w^0,s^0}(\sigma) = g^2_{w^0,s^0}(\sigma)$.
2) Dacă $\sigma \in \Sigma'_{w',s'} - \Sigma$ şi $f'^*(w') = w^0,\ f'(s') = s^0$ atunci $v_{w^0,s^0}(w', s', \sigma') = g^1_{w',s'}(\sigma')$.

Din primul caz este evident că $v{\restriction_{\Sigma''}} = g^2$, deci prima diagramă este comutativă.

Pentru a proba că $(f', g'); (u, v) = (f_1, g^1)$, deoarece $f'; u = f_1$ arătăm că pentru orice $w' \in S'^{*}$ şi $s \in S'$ că

$$g'_{w',s'}; v_{f'^*(w'),f'(s')} = g^1_{w',s'}.$$

Fie $\sigma \in \Sigma'_{w',s'}$. Conform definiţiei lui $g'_{w',s'}$ considerăm două cazuri

a) În cazul în care $\sigma \in \Sigma'_{w',s'} - \Sigma$ avem $v_{f'^*(w'),f'(s')}(g'_{w',s'}(\sigma)) = v_{f'^*(w),f'(s')}(w', s', \sigma) = g^1_{w',s'}(\sigma)$.

b) Dacă $\sigma \in \Sigma_{w,s},\ w \in S^*,\ s \in S$ avem $v_{f'^*(w),f'(s)}(g'_{w,s}(\sigma)) = v_{f^*(w),f(s)}(g_{w,s}(\sigma)) = g^2_{f(w),f(s)}(g_{w,s}(\sigma)) = g^1_{w,s}(\sigma)$.

Unicitatea rezultă uşor. □

Teoriile unei instituţii

Propoziţie 5.5.19 *Fie $\mathcal{I}$ o instituţie în care Sign este categorie inclusivă şi are sume fibrate. Dacă în categoria Sign sumele fibrate conservă incluziunile, atunci în categoria $\mathcal{T}h(\mathcal{I})$ sumele fibrate conservă incluziunile.*

Demonstraţie: În $\mathcal{T}h(\mathcal{I})$ presupunem că incluziunea $i : (\Sigma, E) \longrightarrow (\Sigma_1, E_1)$ şi morfismul $\phi : (\Sigma, E) \longrightarrow (\Sigma_2, E_2)$ au o sumă fibrată. Deoarece în $Sign$ incluziunea $i : \Sigma \longrightarrow \Sigma_1$ şi morfismul

$\phi : \Sigma \longrightarrow \Sigma_2$ au o sumă fibrată şi în categoria signaturilor sumele fibrate conservă incluziunile deducem existenţa unei sume fibrate

$$\begin{tikzcd} & \Sigma_1 \arrow[dr, "\rho"] & \\ \Sigma \arrow[ur, "i"] \arrow[dr, "\phi"'] & & \Sigma' \\ & \Sigma_2 \arrow[ur, hook, "j"'] & \end{tikzcd}$$

în care j este o incluziune de signaturi.

Deoarece functorul $U : \mathcal{T}h(\mathcal{I}) \longrightarrow Sign$ crează slab colimitele, există suma fibrată

$$\begin{tikzcd} & (\Sigma_1, E_1) \arrow[dr, "\rho"] & \\ (\Sigma, E) \arrow[ur, "i"] \arrow[dr, "\phi"'] & & (\Sigma', (\ Sen(\rho)(E_1) \cup Sen(j)(E_2)\ )^{\bullet}) \\ & (\Sigma_2, E_2) \arrow[ur, hook, "j"'] & \end{tikzcd}$$

în care j este o incluziune de teorii. $\square$

Corolar 5.5.20 *În categoria teoriilor din instituţia logicii ecuaţionale sumele fibrate conservă incluziunile.*

5.6 IMPORTURI

Pentru o prezentare cele mai utilizate semantici sunt **semantica iniţială** şi **semantica slabă**. Conform semanticii slabe sunt considerate toate modele prezentării. Conform semanticii iniţiale este considerat doar modelul inţial. Prin urmare in conformitate cu semantica adoptată o prezentare poate da mai multe specificaţii, adică o prezentare şi o clasă de modele ale acesteia. Practica programării cere şi alte metode de definire a specificaţiilor. Printre acestea metoda importurilor.

Pentru definirea semanticii iniţiale este necesar conceptul de săgeată universală.

Definiţia 5.6.1 Fie $F : \mathcal{C} \longrightarrow \mathcal{D}$ un functor, C un obiect din categoria $\mathcal{C}$ şi D un obect din categoria $\mathcal{D}$.

Un morfism $u : D \longrightarrow F(C)$ se numeşte **săgetă universală** dacă pentru orice morfism $f : D \longrightarrow F(C')$

există un unic morfism $f^{\#} : C \longrightarrow C'$ cu proprietatea $u; F(f^{\#}) = f$.

Vom accepta trei feluri de importuri **protecting, extending** şi **using**.

Fie o specificaţie $(\Sigma, E, \mathcal{M})$ importată de prezentarea (Σ', E'). Acceptăm automat că $\mathcal{M} \subseteq Mod^p(\Sigma, E)$ şi că prezentarea importată este inclusă în specificaţia care importă:

$$\alpha : (\Sigma, E) \hookrightarrow (\Sigma', E').$$

Prin urmare $Mod^p(\alpha) : Mod^p(\Sigma', E') \longrightarrow Mod^p(\Sigma, E)$ este un functor uituc, adică uită o parte dintre suporturile şi o parte dintre operaţiile modelelor.

Vom defini specificaţia $(\Sigma', E', \mathcal{M}')$ definind de fapt clasa de modele $\mathcal{M}'$ în funcţie de tipul importului şi de semantica acceptată pentru prezentarea importatoare.

5.6.1 Semantică slabă

Prin definiţie în funţie de tipul importului:

a) **Protecting**

$$\mathcal{M}' = \{M' \in Mod^p(\Sigma', E') \,:\, Mod^p(\alpha)(M') \in \mathcal{M}\}$$

b) **Extending**

$$\mathcal{M}' = \{M' \in Mod^p(\Sigma', E') \,:\, (\exists M \in \mathcal{M}) \text{ şi un morfism injectiv } j : M \longrightarrow Mod^p(\alpha)(M')\}$$

c) **Using**

$$\mathcal{M}' = \{M' \in Mod^p(\Sigma', E') \,:\, (\exists M \in \mathcal{M}) \text{ şi un morfism } j : M \longrightarrow Mod^p(\alpha)(M')\}$$

5.6.2 Semantica iniţială

a) **Protecting**

$$\mathcal{M}' = \{M' \in Mod^p(\Sigma', E') \,:\, Mod^p(\alpha)(M') \in \mathcal{M} \text{ şi } 1_{Mod^p(\alpha)(M')} \text{ este săgeată universală}\}$$

b) **Extending**

$$\mathcal{M}' = \{M' \in Mod^p(\Sigma', E') \,:\, (\exists M \in \mathcal{M}) \text{ şi un morfism injectiv şi universal } j : M \longrightarrow Mod^p(\alpha)(M')\}$$

c) **Using**

$$\mathcal{M}' = \{M' \in Mod^p(\Sigma', E') \,:\, (\exists M \in \mathcal{M}) \text{ şi un morfism universal } j : M \longrightarrow Mod^p(\alpha)(M')\}$$

Dacă o prezentare importă mai multe submodule se utilizează conjucţia condiţiilor de mai sus.

Având în vedere cele de mai sus un program apare ca un graf orientat aciclic conex în care nodurile sunt etichetate cu specificaţii şi săgeţile sunt incluziuni de prezentări. Dacă un modul este importat pe două căi diferite el apare într-un singur nod, fapt pentru care, în general, nu avem de a face cu un arbore.

5.7 PARAMETRIZAREA PROGRAMELOR

Programarea clasică include utilizarea subprogramelor pentru modularizarea programelor mari. Programarea prin rescriere foloseşte pentru acelaşi scop, modularizarea, aşa numita metodă a programării parametrizate.

Aşa numitele argumente, formale respective actuale, din programarea clasică sunt înlocuite de aşa numiţii parametri, formali respectivi actuali. Parametri formali sunt prin definiţie teorii, iar parametri actuali sunt prin definiţie morfisme de prezentări.

Vom ilustra la nivel teoretic tehnica parametrizării programelor.

Fie $M[T1, T2, T3]$ un modul parametrizat formal de trei teorii $T1$, $T2$ şi $T3$. Numărul trei este utilizat pentru acest exemplu. In general orice număr finit este permis. Nu uitaţi că practic, în programe, teoriile sunt reprezentate de prezentări ale lor.

Fie $T1 + T2 + T3$ suma directa în categoria incluziunilor de teorii. Notăm cu $i1 : T1 \hookrightarrow T1 + T2 + T3$, $i2 : T2 \hookrightarrow T1 + T2 + T3$, $i3 : T3 \hookrightarrow T1 + T2 + T3$ şi $i : T1 + T2 + T3 \hookrightarrow M[T1, T2, T3]$ incluzinile care apar.

Fie $M[m1, m2, m3]$ modulul obţinut prin înlocuirea parametrilor formali cu parametri actuali $m1$, $m2$ şi $m3$. Menţionăm că $m1 : T1 \longrightarrow M1$, $m2 : T2 \longrightarrow M2$ şi $m3 : T3 \longrightarrow M3$ sunt morfisme de prezentări. Vom explica în continuare cine este $M[m1, m2, m3]$.

$$\begin{tikzcd}
M[T1,T2,T3] \arrow[r] & M[m1,m2,m3] \\
T1+T2+T3 \arrow[u, hook, "i"] \arrow[r, "m1+m2+m3"] & M1+M2+M3 \arrow[u, hook] \\
T_k \arrow[u, hook, "i_k"] \arrow[r, "m_k"] & M_k \arrow[u, hook, "j_k"]
\end{tikzcd}$$

Fie $M1 + M2 + M3$ suma directă în categoria incluziunilor. Notăm cu $j1 : M1 \hookrightarrow M1 + M2 + M3$, $j2 : M2 \hookrightarrow M1 + M2 + M3$ şi $j3 : M3 \hookrightarrow M1 + M2 + M3$ incluziunile care sunt morfisme structurale ale sumei directe.

Fie $m1 + m2 + m3 : T1 + T2 + T3 \longrightarrow M1 + M2 + M3$ unicul morfism de teorii cu proprietate

$$ik; (m1 + m2 + m3) = mk; jk$$

pentru orice $k \in \{1, 2, 3\}$.

La ultimul pas al construcţiei utilizând morfismul $m1 + m2 + m3 : T1 + T2 + T3 \longrightarrow M1 + M2 + M3$ se face o sumă fibrată care conservă incluziunea $i : T1 + T2 + T3 \hookrightarrow M[T1, T2, T3]$. Se obţine astfel incluziunea $M1 + M2 + M3 \hookrightarrow M[m1, m2, m3]$. Această ultimă incluziune impreună cu incluziunile $jk : Mk \hookrightarrow M1 + M2 + M3$ dau structura programului $M[m1, m2, m3]$.

Bibliography

[1] F. Baader, T. Nipkow. Term Rewriting and All That. Cambridge University Press, 1998

[2] R. Burstall, J. Goguen. Putting theories together to make specifications. In: R. Reddy editor, Proc. Tifth International Joint Conference on Artificial Intelligence, 1977, Cambridge, pp 1045-1058

[3] V. E. Căzănescu, G. Roşu. Weak inclusion systems. *Mathematical Structure in Computer Science,* 7(2):195-206, 1997

[4] R. Diaconescu. *Institution-independent Model Theory* Birkhäuser 2008

[5] R. Diaconescu, K. Futatsugi. *CafeOBJ Report. The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification*, volum 6 in *AMAST Series in Computing*. World Scientific, 1998

[6] R. Diaconescu, J. Goguen, P. Stefaneas. Logical support for modularisation. In Gerard Huet şi Gordon Plotkin, editori, *Logical Environments,* pag 83-130. Cambridge, 1993.

[7] H. Ehrig, B. Mahr. Fundamentals of Algebraic Specification 1: Equations and Initial Semantics. EATCS Monographs on Theoretical Computer Science, vol 6. Springer-Verlag, 1985

[8] H. Ehrig, B. Mahr. Fundamentals of Algebraic Specification 2: Module Specifications and Constraints. EATCS Monographs on Theoretical Computer Science, vol 21. Springer-Verlag, 1990

[9] J. Goguen, Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95-146, 1992

[10] J. Goguen, J. Thatcher, E. Wagner, J. Wright. Initial algebra semantics and continuous algebras, *Journal of the ACM*, 24:68-95, 1977

[11] J. Goguen, J. Thatcher, E. Wagner. An initial algebra approach to the specification, correctness, and implementation of abstract data types, Current trends in Programming Methodology, Vol. IV, Data Structuring(R.T.Yeh, Editor), Prentice-Hall, 1978, 80-149

[12] J. Goguen, T.Winkler, J. Meseguer, K. Futatsugi, J.-P. Jouannaud. Introducing OBJ. In J. Goguen, editor, *applications of Algebraic Specification using OBJ.* Cambridge, 1993

[13] D.E. Knuth, P. Bendix. Simple word problems in universal algebra. In J.Leech editor, Computational Problems in Abstract Algebra, pages 263-97, Pergamon Press, 1970

[14] F. W. Lawvere. Functorial semantics of algebraic theories. *Proceedings, National Academy of Sciences*, 50:869-873, 1993. Summary of Ph.D. Thesis, Columbia University

[15] S. MacLane. *Categories for the working mathematician.* Graduate Texts in Mathematics 5, 2nd edition, Springer-Verlag, 1998.

[16] E. Ohlebusch. *Advanced Topics in Term Rewriting.* Springer Verlag, 2002

[17] Terese, Term Rewriting Systems, Cambridge Universitary Press, 2003