

Cuprins

1	ALGEBRE MULTISORTATE	5
1.1	CONCEPTUL DE ALGEBRĂ MULTISORTATĂ	5
1.1.1	Mulțimi și funcții multisortate	5
1.1.2	Signaturi multisortate	6
1.1.3	Algebre multisortate	7
1.1.4	Morfisme de algebre multisortate	8
1.1.5	Izomorfisme de algebre multisortate	10
1.2	ALGEBRE LIBERE - APLICAȚII	11
1.2.1	Expresii	11
1.2.2	Unicitatea abstracție de un izomorfism a algebrelor libere	13
1.2.3	Tipuri abstracte de date - introducere	14
1.3	SUBALGEBRE	16
1.3.1	Operator de închidere. Familie Moore	16
1.3.2	Părți stabile, Subalgebre	18
1.3.3	Morfisme și părți stabile	20
1.4	EXISTENȚA ALGEBRELOR LIBERE	21
1.4.1	Algebre libere și algebre Peano	21
1.4.2	Algebre Peano	22
1.4.3	Algebra arborilor de derivare	24
1.4.4	Existența algebrelor libere	25
1.5	SEMANTICA ALGEBREI INIȚIALE	27
1.5.1	Semantica unui șir de cifre ca număr natural	27
1.5.2	Un calculator de buzunar	28
1.5.3	Arbori de derivare	30
1.5.4	Scrierea poloneză inversă	32
1.5.5	Compilare	33
1.6	CONGRUENȚE ȘI ALGEBRE CÂT	34
1.6.1	Congruențe	35
1.6.2	Algebre cât	37
1.7	ALGEBRE PROIECTIVE	37
1.7.1	Proiectivitatea algebrelor libere	37
1.7.2	Alte proprietăți	38
1.8	SPRE ABSTRACTIZAREA TIPURILOR DE DATE	39
1.8.1	Ecuatii	39
1.8.2	Ecuatii condiționate	40
1.8.3	Necesitatea utilizării cuantificatorilor în ecuații	40
1.8.4	În primul rând semantica	41
1.8.5	Punctul de vedere local	42
1.8.6	Congruența semantică	42
1.9	TIPURI ABSTRACTE de DATE	44

1.9.1	Tipul abstract al numerelor naturale - continuare	44
1.10	Problema programării prin rescriere	45
2	RESCRIERI	47
2.1	TEORII DEDUCTIVE À la MOISIL	47
2.2	LOGICĂ ECUAȚIONALĂ	48
2.2.1	Reguli de deducție, corectitudine	48
2.2.2	Completitudine	49
2.3	RESCRIERE LOCALĂ	50
2.3.1	Preliminarii	50
2.3.2	Închiderea la contexte	51
2.3.3	Închiderea la preordini compatibile cu operațiile	52
2.3.4	Γ -rescriere	53
2.3.5	Corectitudinea rescrierii	55
2.4	Relația de întâlnire, Forme normale	55
2.4.1	Confluență	55
2.4.2	Completitudinea întâlnirii prin rescriere	56
2.4.3	Forme normale	56
2.4.4	Relații canonice	58
2.5	RESCRIERE ÎN SUBTERMENI	58
2.5.1	Completitudinea rescrierii în subtermeni	59
2.5.2	Rescriere într-un pas	59
2.5.3	Considerații metodologice	60
2.6	UNIFICARE	61
2.6.1	Algoritmul de unificare	61
2.6.2	Terminare	62
2.6.3	Corectitudine	62
2.7	LOCAL CONFLUENȚĂ	63
2.8	RESCRIERE MODULO ECUATII	64
2.8.1	Motivare semantică	64
2.8.2	Rescrierea modulo o relație de echivalență	65
2.9	DEMONSTRAREA ECUAȚIILOR CONDIȚIONATE	68
2.9.1	Preliminarii	68
2.9.2	Schimbarea semnăturii	69
2.9.3	Translatarea ecuațiilor	70
2.9.4	Teorema constantelor	71
2.10	PERECHI CRITICE	73
2.11	TERMINAREA și PROCEDURA KNUTH-BENDIX	78
2.11.1	Terminarea programelor	78
2.11.2	Exemplul teoriei grupurilor	80
2.11.3	Completarea Knuth-Bendix	81
3	PROGRAMARE LOGICĂ ECUAȚIONALĂ	87
3.1	IN PRIMUL RAND SEMANTICA	87
3.2	TEOREMELE LUI HERBRAND	88
3.2.1	Introducere	88
3.2.2	Teoremele lui Herbrand	89
3.3	REGULILE PROGRAMĂRII LOGICE	90
3.3.1	Soluții	90
3.3.2	Reguli de deducție	91
3.4	CORECTITUDINEA REGULILOR PROGRAMĂRII LOGICE	93

3.5	COMPLETITUDINEA PARAMODULATIEI	94
3.5.1	Legături între regulile de deducție	94
3.5.2	Amintiri despre rescriere	97
3.5.3	Prolog	97
3.5.4	Completitudinea	98
3.6	COMPLETITUDINEA NARROWINGULUI	98
3.6.1	Introducere	98
3.6.2	Amintiri despre formele normale	99
3.6.3	Lema de ridicare	99
3.6.4	Epilog	101
3.6.5	Completitudine	102
3.7	REZOLUȚIE Ț LA PROLOG	102
3.7.1	Rezoluția	102
3.7.2	Rezoluție = Narrowing = Paramodulație	103
3.8	EXEMPLE	104
3.8.1	Primul	104
3.8.2	Al doilea	105
4	INSTITUȚII	107
4.1	CATEGORII	107
4.1.1	Morfisme distinse	108
4.1.2	Dualitate	109
4.1.3	Subcategorii	110
4.2	FUNCTORI	111
4.3	CONEXIUNI GALOIS	112
4.3.1	Conexiunea Galois atașată unei relații	113
4.4	INSTITUȚII	113
4.4.1	Consecință semantică	114
4.4.2	Exemple	116
4.5	PREZENTĂRI	118
4.5.1	Instituția prezentărilor	119
4.6	TEORII	120
4.6.1	Instituția teoriilor	120
4.7	INSTITUȚIA LOGICII ECUAȚIONALE MULTISORTATE	121
4.7.1	Categoria signaturilor algebrice Sig	121
4.7.2	Functorul <i>Alg</i>	122
4.7.3	Functorul <i>Sen</i>	123
4.7.4	Incluziuni de semnături	124
4.7.5	Relația și condiția de satisfacere	125
5	MODULARIZARE	127
5.1	LIMITE și COLIMITE ÎNTR-O CATEGORIE	127
5.1.1	Colimite de mulțimi	128
5.1.2	Despre unicitatea colimitelor	129
5.1.3	Crearea colimitelor	130
5.2	COLIMITE DE SIGNATURI ALGEBRICE	131
5.3	FUNCTORUL Alg CONSERVĂ COLIMITELE	135
5.3.1	Alg conservă colimitele	136
5.4	COLIMITE ÎN INSTITUȚII	141
5.4.1	Functorul \bullet	143
5.5	CATEGORII INCLUSIVE	145

5.5.1	Categorii inclusive cu sume	149
5.5.2	Sume fibrante care conservă incluziunile	151
5.6	IMPORTURI	153
5.6.1	Semantică slabă	154
5.6.2	Semantica inițială	154
5.7	PARAMETRIZAREA PROGRAMELOR	155

Chapter 1

ALGEBRE MULTISORTATE

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

Presupunem că cititorul are cunoștințe de teoria cantoriană a mulțimilor și puține cunoștințe de algebră. Pentru înțelegerea exemplelor se presupun cunoștințe 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 și 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 apariția algebrelor multisortate și în particular a mulțimilor multisortate.

Algebrele multisortate au fost generalizate conducând la algebrele ordonat sortate.

1.1.1 Mulțimi și funcții multisortate

Fixăm mulțimea S a sorturilor.

Definiția 1.1.1 O familie de mulțimi $M = \{M_s\}_{s \in S}$ indexată de S se numește **mulțime S -sortată**.

Observăm că aceeași literă este folosită atât pentru întreaga mulțime M cât și pentru toate componentele acesteia, M_s unde $s \in S$.

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

Conceptele uzuale cu mulțimi se extind pe componente la mulțimile S -sortate așa cum se vede din exemplele de mai jos

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

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

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

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

O funcție între două mulțimi S -sortate duce un element din prima mulțime într-un element de același sort din a doua mulțime.

Definiția 1.1.2 O funcție S -sortată

$$f : M \longrightarrow N$$

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

Ca și în cazul mulțimilor S -sortate, operațiile cu funcțiile S -sortate se fac pe componente. Dacă $f : M \longrightarrow N$ și $g = \{g_s\}_{s \in S} : N \longrightarrow P$ sunt funcții 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 $s \in S$ și $x \in M_s$. Semnul $;$ folosit pentru compunere este inspirat din limbajele de programare. Mai observăm scrierea diagramatică $f; g$ utilizată pentru compunere în opoziție cu scrierea clasică $g \circ f$

$$\begin{array}{ccccc} M & \xrightarrow{f} & N & \xrightarrow{g} & P \\ & & \searrow f;g & & \end{array}$$

Compunerea funcțiilor S -sortate este asociativă. Dacă și $h = \{h_s\} : P \longrightarrow R$ este funcție S -sortată, atunci folosind asociativitatea compunerii funcțiilor 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 mulțime S -sortată M funcția 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 funcția identitate a mulțimii M_s . Funcția identitate are efect neutru la compunere. Pentru orice funcție S -sortată $f : M \longrightarrow N$ au loc egalitățile $1_M; f = f = f; 1_N$.

Cele două proprietăți de mai sus ne permit să vorbim de categoria mulțimilor S -sortate (Vezi definiția 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** așa cum vom spune în continuare. Mai mult, de cele mai multe ori, în diferitele construcții sintactice, într-un anumit loc al acestora nu poate fi plasată decât o dată de un anumit sort. Aceasta ar fi explicația 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 și una pentru operații.

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

Fiecare operație este caracterizată de modul acesteia de acțiune. Operația acționează pe un anumit număr fix de date de sorturi precizate și are rezultatul de un sort dat. Ca exemplu pentru operația cu **numele** o notăm cu

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

faptul ca ea are n argumente de sorturi s_1, s_2, \dots, s_n și rezultatul este de sort s . Numele unei operații mai este numit și **simbol de operație**.

Terminologia folosită este următoarea:

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

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

Toate aceste informații privind felul algebrei sunt adunate în conceptul de signatură. Cu S^* notăm mulțimea șirurilor finite formate cu elemente din S .

Definiția 1.1.3 O semnătură

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

este formată dintr-o mulțime S ale cărei elemente se numesc sorturi și o familie de mulțimi

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

Pentru fiecare $s_1 s_2 \dots s_n \in S^*$ și $s \in S$ mulțimea $\Sigma_{s_1 s_2 \dots s_n, s}$ conține numele operațiilor cu n argumente de sorturi s_1, s_2, \dots, s_n și rezultat de sort s .

Menționăm că mulțimile $\Sigma_{s_1 s_2 \dots s_n, s}$ pot avea elemente comune, ceea ce permite modelarea supraîncărcării operațiilor, adică permisiunea ca mai multe operații să aibă același nume, sau altfel spus să fie denumite prin același simbol.

Când nu există pericol de confuzie vom scrie pur și simplu (S, Σ) sau Σ în loc de

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

Cea mai cunoscută semnătură multisortată provenită din algebra clasică este cea corespunzătoare conceptelor de spațiu vectorial sau modul. Simbolurile de operații sunt de trei feluri

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

O consecință deosebită a stilului multisortat este faptul că relațiile pot fi definite ca operații cu sortul rezultat boolean. De exemplu

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

Observăm că $3 \leq 5 = \text{adevăr}$ și $5 \leq 3 = \text{fals}$.

În general prin **relație** se înțelege o operație al cărei sort rezultat este boolean. Practic, pentru a transforma o relație în operație se înlocuiește relația cu așa numita funcția ei caracteristică. Reamintim că funcția caracteristică

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

a submulțimii A a mulțimii 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 studiul clasic al modelelor unde apar atât operații cât și relații poate fi redus la studiul algebrelor multisortate unde apar numai operații.

1.1.3 Algebre multisortate

Algebrele sunt formate în mare din date și operații. Datele sunt de mai multe sorturi, adică pentru fiecare sort s algebra conține o mulțime a datelor de sort s . Familia acestor mulțimi, numită și **suportul algebrei**, constituie o mulțime sortată.

Definiția 1.1.4 O Σ -algebră $\mathcal{A} = (\{A_s\}_{s \in S}, \{A_\sigma\}_{\sigma \in \Sigma})$ este formată dintr-o mulțime S -sortată, numită suportul algebrei, $A = \{A_s\}_{s \in S}$ și o familie de operații $\{A_\sigma\}_{\sigma \in \Sigma}$. Pentru claritate, dacă $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$, adică $\sigma : s_1 s_2 \dots s_n \longrightarrow s$, atunci A_σ este o funcție

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

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 menționăm că pentru o algebră \mathcal{A} și suportul acesteia A folosim aceeași literă cu grafii diferite.

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

În continuare pentru $s_1, s_2, \dots, s_n \in S$ vom mai folosi și notația

$$A_{s_1 s_2 \dots s_n} = A_{s_1} \times A_{s_2} \times \dots \times A_{s_n}.$$

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

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

Asemănător algebrei care abordează pe rând diferite structuri algebrice, trebuie să facem același lucru. Adică trebuie să studiem, dar în același timp, structuri algebrice de naturi diferite. Prin urmare, în continuare, fixăm semnatura (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 aceeași semnătură.

Fixarea semnăturii arată că ne ocupăm de o anumită structură algebrică. Faptul că semnatura este arbitrară arată că 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 funcție multisortată între suporturile celor două algebre care verifică o condiție suplimentară. Pentru a scrie această condiție pentru cazul algebrelor multisortate să plecăm de la conceptul uzual de morfism pentru o structură algebrică bazată pe o operație binară. Funcția $h : A \longrightarrow B$ 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 operației și apoi se egalează rezultatele

- membrul stâng:

- 1) se aplică operația din prima algebră elementelor din prima algebra
- 2) se aplica morfismul h rezultatului obținut

- membrul drept:

- 1) se aplică morfismul h elementelor din prima algebra obținându-se niște elemente din a doua algebră
 - 2) se aplică operația din a doua algebră acestor elemente
- se cere ca rezultatul evaluării celor doi membri să fie egali.

Să facem același lucru pentru două algebre multisortate $\mathcal{A} = (A_s, A_\sigma)$, $\mathcal{B} = (B_s, B_\sigma)$ și o funcție S -sortată $h : A \longrightarrow B$. Condiția de mai sus trebuie pusă pentru fiecare operație cu numele

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

și oricare ar fi elementele $a_1 \in A_{s_1}$, $a_2 \in A_{s_2} \dots a_n \in A_{s_n}$

- membrul stâng:

- 1) se aplică operația din prima algebră elementelor din prima algebra: $A_\sigma(a_1, a_2, \dots, a_n)$
- 2) se aplica morfismul h rezultatului obținut $h_s(A_\sigma(a_1, a_2, \dots, a_n))$

- membrul drept:

- 1) se aplică morfismul h elementelor din prima algebra obținându-se niște elemente din a doua algebră:

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

- 2) se aplică operația 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.

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

Definiția 1.1.5 Funcția 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 , $a_n \in A_{s_n}$

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

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 și notația

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

Cu aceleași notații 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 condiție pentru fiecare nume de operație. În cazul operațiilor fără argumente, așa zisele constante, condiția 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 și că orice morfism de seminele duce elementul neutru la adunare, respectiv la înmulțire, tot în elementul neutru la adunare respectiv la înmulțire.

Cu notațiile de mai sus condiția de morfism pentru operația $\sigma : s_1 s_2 \dots s_n \longrightarrow s$ este echivalentă cu

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

Este deasemenea util să menționăm diferența de notație dintre o funcție S -sortată

$$f : A \longrightarrow B$$

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

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

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

Propoziție 1.1.6 *Compunerea ca funcții S -sortate a două morfisme de Σ -algebre este un morfism de Σ -algebre.*

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

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

$$\begin{aligned} (h; g)_s(A_\sigma(a_1, a_2, \dots, a_n)) &= g_s(h_s(A_\sigma(a_1, a_2, \dots, a_n))) = g_s(B_\sigma(h_{s_1}(a_1), h_{s_2}(a_2), \dots, h_{s_n}(a_n))) \\ &= C_\sigma(g_{s_1}(h_{s_1}(a_1)), g_{s_2}(h_{s_2}(a_2)), \dots, g_{s_n}(h_{s_n}(a_n))) = C_\sigma((h; g)_{s_1}(a_1), (h; g)_{s_2}(a_2), \dots, (h; g)_{s_n}(a_n)). \quad \square \end{aligned}$$

Compunerea morfismelor de Σ -algebre este asociativă.

Morfismul identitate are efect neutru la compunere.

Datorită celor două proprietăți de mai sus putem vorbi de categoria Σ -algebrelor.

1.1.5 Izomorfisme de algebre multisortate

Definiția 1.1.7 Morfismul de Σ -algebre $h : \mathcal{A} \longrightarrow \mathcal{B}$ se numește **izomorfism** dacă există morfismul $g : \mathcal{B} \longrightarrow \mathcal{A}$ cu proprietățile $h; g = 1_{\mathcal{A}}$ și $g; h = 1_{\mathcal{B}}$.

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

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

Datorită unicității sale, conform uzanțelor, morfismul g , denumit și inversul lui h , este notat în continuare cu h^{-1} .

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

Propoziție 1.1.8 *Un morfism este izomorfism dacă și numai dacă are toate componentele bijective.*

Demonstrație: 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 proprietățile $h; h^{-1} = 1_{\mathcal{A}}$ și $h^{-1}; h = 1_{\mathcal{B}}$. Rezultă că pentru orice sort $s \in S$ au loc egalitățile $h_s; h_s^{-1} = 1_{A_s}$ și $h_s^{-1}; h_s = 1_{B_s}$, adică funcția h_s este inversabilă pentru orice $s \in S$, deci toate componentele h_s ale lui h sunt bijectii.

Reciproc, presupunem că toate componentele h_s ale lui h sunt bijectii. Prin urmare pentru orice $s \in S$ există funcția $h_s^{-1} : B_s \longrightarrow A_s$ cu proprietățile $h_s; h_s^{-1} = 1_{A_s}$ și $h_s^{-1}; h_s = 1_{B_s}$. De aici notând $h^{-1} = \{h_s^{-1}\}_{s \in S}$ rezultă pentru orice $s \in S$ că

$$(h; h^{-1})_s = h_s; h_s^{-1} = 1_{A_s} = (1_{\mathcal{A}})_s \quad \text{și} \quad (h^{-1}; h)_s = h_s^{-1}; h_s = 1_{B_s} = (1_{\mathcal{B}})_s$$

deci $h; h^{-1} = 1_{\mathcal{A}}$ și $h^{-1}; h = 1_{\mathcal{B}}$.

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

Fie $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ și $b_1 \in B_{s_1}$, $b_2 \in B_{s_2}$, \dots , $b_n \in B_{s_n}$. Deoarece h este morfism pentru elementele $h_{s_1}^{-1}(b_1), h_{s_2}^{-1}(b_2), \dots, h_{s_n}^{-1}(b_n)$ din \mathcal{A} deducem

$$\begin{aligned} h_s(A_{\sigma}(h_{s_1}^{-1}(b_1), h_{s_2}^{-1}(b_2), \dots, h_{s_n}^{-1}(b_n))) &= B_{\sigma}(h_{s_1}(h_{s_1}^{-1}(b_1)), h_{s_2}(h_{s_2}^{-1}(b_2)), \dots, h_{s_n}(h_{s_n}^{-1}(b_n))) \\ &= B_{\sigma}(b_1, b_2, \dots, b_n). \end{aligned}$$

Aplicând funcția h_s^{-1} ambilor membri deducem

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

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

Propoziție 1.1.9 *Compunerea a două izomorfisme $f : \mathcal{A} \longrightarrow \mathcal{B}$ și $g : \mathcal{B} \longrightarrow \mathcal{C}$ este un izomorfism $f; g : \mathcal{A} \longrightarrow \mathcal{C}$. În plus*

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

Demonstrație: 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 obțin identități. Prin urmare

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

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

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

1.2 ALGEBRE LIBERE - APLICAȚII

După această mică introducere privind algebrele multisortate, trecem la conceptul de algebră liberă datorită importanțelor aplicații ale acestuia în informatică. Cu el se modelează noțiunile de expresie și de evaluarea a unei expresii.

Menționăm ca demonstrația privind existența algebrelor libere prezintă dificultăți de natură tehnică fapt pentru care este prezentată mai târziu și recomandată numai celor care posedă spiritul demonstrațiilor matematice.

Cealaltă se pot mulțumi numai cu explicațiile de mai jos, acceptând, fără demonstrație, existența algebrelor libere.

1.2.1 Expresii

Ce este o expresie?

Conceptul de *expresie* așa cum este el folosit în învățământul preuniversitar nu are o definiție și un înțeles 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 operația $*$ 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 noțiunea de expresie va fi definită în ipoteza că *operațiile cu care lucrăm nu au nici o proprietate suplimentară*.

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

Definiția 1.2.1 Σ -algebra $\mathcal{A} = (A_s, A_\sigma)$ se numește liber generată de $V \subseteq A$ dacă pentru orice Σ -algebră \mathcal{D} și pentru orice funcție sortată $f : V \rightarrow D$, există un unic morfism de Σ -algebre $f^\# : \mathcal{A} \rightarrow \mathcal{D}$ care extinde f .

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

Definiția 1.2.2 Se numește **expresie** un element dintr-o algebră liberă.

Când privim algebra liberă ca o algebră de expresii, mulțimea V de mai sus reprezintă mulțimea variabilelor.

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

În plus noțiunea naivă de expresie ne dă intuiția necesară pentru înțelegerea conceptului de algebră liberă: algebrele libere nu sunt altceva decât algebre de expresii.

Independența de modul de scriere al expresiilor corespunde unicității abstracției de un izomorfism al algebrei libere pentru care este fixată mulțimea V a generatorilor.

Evaluarea expresiilor

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

1. o algebră în care se fac calculele și care are aceeași semnătură cu cea a expresiei
2. valori variabilelor din expresie în algebra în care se fac calculele.

Menționăm că **a da valori variabilelor din mulțimea X în algebra \mathcal{D}** este echivalent cu a da o funcție

$$v : X \longrightarrow D.$$

Pentru orice variabilă x din X valoarea dată lui x este $v(x)$.

Vom nota cu $T_\Sigma(X)$ algebra liber generată de mulțimea 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} și pentru orice funcție $v : X \longrightarrow D$ există, conform definiției algebrelor libere, un unic morfism $v^\# : T_\Sigma(X) \longrightarrow \mathcal{D}$ a cărui restricție la X coincide cu v .

Fixând algebra \mathcal{D} vom constata că există o bijecție naturală între $\text{Alg}_\Sigma(T_\Sigma(X), \mathcal{D})$ mulțimea morfismelor de Σ -algebre de la $T_\Sigma(X)$ la \mathcal{D} și $\text{Set}_S(X, D)$ mulțimea funcțiilor S -sortate de la X la D . Fie

$$r : \text{Alg}_\Sigma(T_\Sigma(X), \mathcal{D}) \longrightarrow \text{Set}_S(X, D)$$

funcția restricție, adică $r(h) : X \longrightarrow D$ este restricția $h|_X$ a morfismului $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ la X . Proprietatea de mai sus a algebrei libere spune că

$$(\forall v \in \text{Set}_S(X, D))(\exists! v^\# \in \text{Alg}_\Sigma(T_\Sigma(X), \mathcal{D}))r(v^\#) = v$$

adică r este bijecție. Existența acestei bijecții ne permite să identificăm elementele celor două mulțimi fără a mai face distincție între un morfism $v^\#$ de la $T_\Sigma(X)$ la \mathcal{D} și v , restricția lui la X .

Dacă mai sus scriam că **a da valori variabilelor din mulțimea X în algebra \mathcal{D}** este echivalent cu **a da o funcție $v : X \longrightarrow \mathcal{D}$** acum putem spune că:

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

O altă consecință 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 funcție de la X la \mathcal{D} .

Definiția 1.2.3 Dacă $e \in T_\Sigma(X)$ este o expresie cu variabile din X și $h : T_\Sigma(X) \longrightarrow \mathcal{D}$ morfismul prin care se dau valori în \mathcal{D} variabilelor, atunci $h(e)$ este rezultatul evaluării expresiei e pentru valorile variabilelor date de funcția $h|_X : X \longrightarrow \mathcal{D}$. \square

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

Pentru valorile $x = 2$, $y = 3$ și $z = 1$ intuitiv obținem $2 * (3 + 1) = 8$ iar cu definiția de mai sus unde

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

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

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

Semantica instrucțiunii de atribuire

Fie X mulțimea variabilelor utilizate întrun program. O instrucțiune de atribuire este de forma $x := e$ unde x este o variabilă și e este o expresie, adică $e \in T_\Sigma(X)$.

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

mulțimea stărilor memoriei, adică mulțimea funcțiilor de la mulțimea variabilelor X la mulțimea datelor D .

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

Semantica unei instrucțiuni, sau mai general a unui program, este o funcție parțială F de la mulțimea S a stărilor la ea însăși. Funcția F este definită pentru starea s a memoriei dacă și numai dacă execuția instrucțiunii începută în starea s a memoriei se termină. Mai mult $F(s)$ este starea memoriei în momentul terminării execuției.

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

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

1.2.2 Unicitatea abstracție de un izomorfism a algebrelor libere

Teorema 1.2.4 *Două algebre liber generate de aceeași mulțime sunt izomorfe.*

Demonstrație: Fie $\mathcal{A} = (A_s, A_\sigma)$ și $\mathcal{B} = (B_s, B_\sigma)$ două Σ -algebre liber generate de X . Notăm cu $i_A : X \longrightarrow A$ și $i_B : X \longrightarrow B$ funcțiile incluziune ale lui X în A , respectiv în B . Demonstrația are patru pași.

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} și \mathcal{B} . La fel vor fi pașii 3 și 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 $i_A; (f; g) = (i_A; f); g = i_B; g = i_A = i_A; 1_A$ și deoarece \mathcal{A} este algebră liber generată de X deducem $f; g = 1_A$.

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

Deci f și g sunt izomorfisme inverse unul altuia.

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

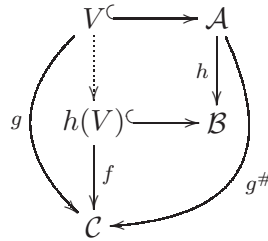
Demonstrație: Fie \mathcal{A} o Σ -algebră liber generată de $V \subseteq A$ și $h : \mathcal{A} \longrightarrow \mathcal{B}$ un izomorfism.

Probăm că Σ -algebra \mathcal{B} este liber generată de $h(V) \subseteq B$. Fie \mathcal{C} o Σ -algebră și $f : h(V) \longrightarrow C$ o funcție S -sortată. Fie $g : V \longrightarrow C$ funcția 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

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



Calculând restricția 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 funcției f .

Vă propunem să demonstrați unicitatea acestei extinderi.

Algebre inițiale

Definiția 1.2.6 O Σ -algebră \mathcal{I} se numește **inițială** dacă pentru orice Σ -algebră \mathcal{A} există un unic morfism

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

Observăm că o algebră este inițială dacă și numai dacă este liber generată de mulțimea vidă. Din cele două propoziții de mai sus rezultă că:

Σ -algebra inițială este unică abstracție făcând de un izomorfism.

Acest fapt are aplicații importante în informatică.

Un **tip de date** se numește **abstract** dacă este unic determinat abstracție făcând de un izomorfism. Se vede prin urmare că dând o semnătură am dat implicit, prin algebra inițială corespunzătoare semnăturii, un tip abstract de date.

Vom da un exemplu cunoscut din algebra de liceu. Se știe că numerele întregi formează un inel inițial. Vă invităm să reflectați asupra următoarei definiții a ideii de număr întreg.

Se numește **număr întreg** un element al inelului inițial.

1.2.3 Tipuri abstracte de date - introducere

Un tip de date se numește **abstract** dacă este unic determinat abstracție 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 inițiale. Mai clar : este suficient să dăm o semnătură și eventual niște axiome(ecuații condiționate sau nu) deoarece algebra inițială, a cărei existență este garantată de teoremele care vor fi prezentate mai târziu, este unic determinată abstracție de un izomorfism, prin urmare este un tip abstract de date.

Tipul numerelor naturale poate fi definit abstract ca fiind semiinelul inițial.

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

Definițiile de mai sus, deși corecte sunt uneori ineficiente, deoarece nu face posibilă execuția de calcule. Prin urmare dorim alte definiții echivalente care permit calculatorului să facă calcule. Vom exemplifica pentru numere naturale fără a intra în prea multe detalii.

Tipul abstract al numerelor naturale

Semiinelul este o mulțime M înzestrată cu două operații binare notate cu $+$ și $*$ cu următoarele proprietăți

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

Mulțimea numerelor naturale cu adunarea și înmulțirea uzuale este un semiinel inițial, adică conceptul de semiinel caracterizează numerele 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 cărui i se atribuie prima punere în evidență a ideilor abstracte privind numerele naturale este F.W. Lawvere[15].

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

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

Elementele algebrei inițiale sunt

$$0, s(0), s(s(0)), s(s(s(0))), s(s(s(s(0)))) , \dots$$

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

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

Propoziție 1.2.7 Fie $\mathcal{N} = (N, 0_N, s_N)$ algebra definită prin: *N* este mulțimea numerelor naturale, 0_N este numărul natural zero și $s_N(n) = n + 1$ pentru orice număr natural *n*. Algebra \mathcal{N} este inițială.

Demonstrație: Fie $\mathcal{A} = (A, 0_A, s_A)$ o altă algebra pentru signatura de mai sus. Definim funcția $h : N \longrightarrow A$ prin inducție

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

Prima egalitate de mai sus și $h(s_N(n)) = s_A(h(n))$ 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 inducție că $g(n) = h(n)$ pentru orice *n* natural.

$$\begin{aligned} g(0_N) &= 0_A = h(0_N) \text{ și} \\ g(n+1) &= g(s_N(n)) = s_A(g(n)) = s_A(h(n)) = h(s_N(n)) = h(n+1). \quad \square \end{aligned}$$

Propoziția anterioară ne arată cum pot fi definite numerele naturale prin metoda algebrei inițiale ca tip abstract de date. Ea dovedește corectitudinea definiției de mai sus.

Deocamdată prin signatura de mai sus calculatorul învață numerele naturale dar nu știe încă să calculeze. Pentru început să-l învățăm să adune. Dacă introducem în signatură o operație binară +

$$\text{op } _+ _ : \text{nat nat} \longrightarrow \text{nat}$$

nu realizăm nimic altceva decât să adăugăm la mulțimea de mai sus a numerelor naturale foarte mult gunoi. De exemplu, deoarece calculatorul nu știe încă să adune, $0 + 0$ este un nou element de care nu avem nevoie. Il învățăm să adune dându-i următoarele două reguli de rescriere precedate de o declarație de variabile

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

Trebuie să remarcăm diferența esențială dintre o regulă de rescriere și o egalitate. O regulă de rescriere se aplică numai de la stânga la dreapta, deoarece simetria este unul dintre marii dușmani ai programării prin rescriere conducând la neterminarea programelor.

Ce părere aveți despre comutativitate?

Să vedem cum efectuează mașina adunarea $2 + 2$, adică:

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

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

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

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

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

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

Corectitudinea acestei definiții precum și a celei care urmează va fi probată mai târziu, în secțiunea 1.9.1.

Calculatorul va ști să și înmulțească dacă mai introducem o operație binară și două reguli de rescriere

$$\begin{aligned} \text{op } _ * _ : \text{nat nat} &\longrightarrow \text{nat} . \\ \text{eq } X * 0 &= 0 . \\ \text{eq } X * s(Y) &= X * Y + X . \end{aligned}$$

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 noțiunea intuitivă dată de o algebră \mathcal{D} a datelor, trebuie să găsim signatura și eventual ecuațiile pentru care \mathcal{D} devine algebră inițială.

Exemplul numerelor naturale este doar un început. Vă propunem de exemplu să definiți relația de ordine ca operație $< : \text{nat nat} \longrightarrow \text{bool}$. Piatra de încercare va fi operația de împărțire.

1.3 SUBALGEBRE

Vom continua prezentarea teoriei algebrelor multisortate cu conceptele de parte stabilă și subalgebră. Deoarece părțile 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 mulțime parțial ordonată, adică relația \leq pe mulțimea P este reflexivă, tranzitivă și antisimetrică.

Definiția 1.3.1 Se numește **operator de închidere** o funcție

$$\bullet : P \longrightarrow P$$

cu următoarele proprietăți:

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

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

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

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

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

Propoziție 1.3.3 *Dacă $\bullet : P \longrightarrow P$ este un operator de închidere, atunci mulțimea elementelor închise*

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

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

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

Deoarece $p^\bullet = p^{\bullet\bullet}$ rezultă că $p^\bullet \in M$. Conform extensivității $p \leq p^\bullet$ deducem că p^\bullet aparține mulțimii M_p . Pentru orice $m \in M_p$ din $p \leq m$ deducem $p^\bullet \leq m^\bullet$ și din $m \in M$ deducem $m^\bullet = m$ deci $p^\bullet \leq m$. \square

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

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

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

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

Deoarece $p^{\bullet\bullet}$ este cel mai mic element din mulțimea $\{m \in M \mid p^\bullet \leq m\}$ și p^\bullet aparține acestei mulțimi 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 $q^\bullet \in M$ și $q \leq q^\bullet$, prin urmare $p \leq q^\bullet$. Deoarece $q^\bullet \in \{m \in M \mid p \leq m\}$ deducem $p^\bullet \leq q^\bullet$. \square

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

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

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

Deci operatorii de închidere $'$ și \bullet coincid. \square

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

Demonstrație: Arătăm pentru orice $p \in P$ că $p \in M$ dacă și 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 mulțimea $\{m \in M \mid p \leq m\}$ deci $p = p^\bullet$ adică $p \in M^\bullet$. \square

Din propozițiile de mai sus se observă echivalența celor două concepte. Fixând mulțimea parțial ordonată (P, \leq) , considerăm mulțimea \mathcal{O} a operatorilor de închidere și mulțimea \mathcal{M} a familiilor Moore din P . Propoziția 1.3.3 ne dă o funcție de la \mathcal{O} la \mathcal{M} . Propoziția 1.3.5 ne dă o funcție de la \mathcal{M} la \mathcal{O} . Celelalte două propoziții ne spun că aceste funcții sunt bijecții inverse una alteia.

Bazându-ne pe motivația de mai sus nu vom mai face nici o distincție între cele două concepte, apariția în discuție a unuia dintre ele introducând automat și pe celălalt.

Următoarea propoziție este utilă în multe exemple.

Propoziție 1.3.8 Într-o latice completă L , M este familie Moore dacă și numai dacă pentru orice $A \subseteq M$, $\inf A \in M$.

Demonstrație: Reamintim că prin definiție orice submulțime A dintr-o latice completă are infimum (cel mai mare minorant), pe care în continuare îl notăm $\inf A$.

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

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

1.3.2 Părți 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 operații din algebră unor elemente din P este tot în P .

Definiția 1.3.9 Fie $\mathcal{A} = (A_s, A_\sigma)$ o Σ -algebră și $P_s \subseteq A_s$ pentru orice $s \in S$. Partea $P = \{P_s\}_{s \in S}$ a lui \mathcal{A} se numește stabilă dacă pentru orice $s_1 s_2 \dots s_n \in S^*$, pentru orice $s \in S$, pentru orice $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$, pentru orice $a_1 \in P_{s_1}, a_2 \in P_{s_2}, \dots, a_n \in P_{s_n}$ elementul $A_\sigma(a_1, a_2, \dots, a_n)$ este în P_s .

Observăm că orice parte stabilă conține 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 părțile stabile ale oricărei algebre formează o familie Moore. Pentru aceasta se poate demonstra că orice intersecție de părți stabile este o parte stabilă și aplica apoi propoziția de mai sus. Lăsăm acest fapt ca exercițiu. Vom prefera o cale mai dificilă, utilizând direct definiția, dar cu rezultat mult mai util în multe cazuri.

Fie $\mathcal{A} = (A_s, A_\sigma)$ o Σ -algebră și $X \subseteq A$. Definim prin inducție șirul de părți ale lui \mathcal{A} astfel

$$X^0 = X \text{ și } X_s^{n+1} = X_s^n \cup \{A_\sigma(a) : w \in S^*, \sigma \in \Sigma_{w, s}, a \in X_w^n\} \text{ pentru orice } n \in \mathbb{N} \text{ și orice } s \in S.$$

Observăm că șirul $\{X^n\}_{n \in \mathbb{N}}$ este crescător. Mai observăm că mulțimea X^{n+1} se obține adăugând la X^n rezultatele aplicării unei singure operații, o singură dată, unor elemente din X^n .

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

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

Propoziție 1.3.10 \overline{X} este partea stabilă generată de X .

Demonstrație: 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 $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ și $a_1 \in \overline{X}_{s_1}$, $a_2 \in \overline{X}_{s_2}$, \dots , $a_n \in \overline{X}_{s_n}$. Pentru orice $1 \leq i \leq n$ din $a_i \in \overline{X}_{s_i}$, adică $a_i \in \bigcup_{n \in \mathbb{N}} X_{s_i}^n$, 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, \dots, k_n . Deoarece șirul $\{X^n\}_n$ este crescător deducem că $a_i \in X_{s_i}^k$ pentru orice $1 \leq i \leq n$. Din definiția șirului $\{X^n\}_n$ deducem $A_\sigma(a_1, a_2, \dots, a_n) \in X_s^{k+1}$. Deoarece $X^{k+1} \subseteq \overline{X}$ deducem că $A_\sigma(a_1, a_2, \dots, a_n) \in \overline{X}_s$ deci \overline{X} este parte stabilă.

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

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

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

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

Definiția 1.3.11 Fie \mathcal{A} o Σ -algebră și $X \subseteq A$. Dacă $\overline{X} = A$ spunem că X generează \mathcal{A} sau că \mathcal{A} este generată de X .

Operatorul de închidere asociat familiei Moore a părților stabile mai are și următoarele proprietăți:

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

Inducție structurală

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

Fie $\mathcal{A} = (A_s, A_\sigma)$ o Σ -algebră, X o mulțime de generatori ai algebrei \mathcal{A} și \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 \mathbf{P} ,
2. pentru orice $s_1 s_2 \dots s_n \in S^*$, $s \in S$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ dacă $a_1 \in A_{s_1}$, $a_2 \in A_{s_2}$, \dots , $a_n \in A_{s_n}$ sunt elemente arbitrare cu proprietatea \mathbf{P} , atunci $A_\sigma(a_1, a_2, \dots, a_n)$ are proprietatea \mathbf{P} .

Să ne convingem de corectitudinea inducției structurale. Fie B submulțimea lui A formată din toate elementele algebrei \mathcal{A} care au proprietatea \mathbf{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 proprietăți 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 \mathbf{P} .

Subalgebre

Conceptul de subalgebră este foarte apropiat de cel de parte stabilă. Diferența principală constă în faptul că subalgebra este o algebră iar partea stabilă este o mulțime.

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

1.3.3 Morfisme și părți stabile

O pereche de morfisme cu același domeniu și același codomeniu se mai numește și săgeată dublă.

Definiția 1.3.12 Fie $f : \mathcal{A} \longrightarrow \mathcal{B}$ și $g : \mathcal{A} \longrightarrow \mathcal{B}$ două morfisme. Numim **nucleu de săgeată dublă** al morfismelor f și g submulțimea lui \mathcal{A} notată $\mathbf{Ker}(f, g)$ și definită pentru orice sort s prin

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

Propoziție 1.3.13 Nucleul de săgeată dublă este o parte stabilă.

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

$$\begin{aligned} f_s(A_\sigma(a_1, a_2, \dots, a_n)) &= B_\sigma(f_{s_1}(a_1), f_{s_2}(a_2), \dots, f_{s_n}(a_n)) = B_\sigma(g_{s_1}(a_1), g_{s_2}(a_2), \dots, g_{s_n}(a_n)) = \\ &= g_s(A_\sigma(a_1, a_2, \dots, a_n)), \text{ deci } A_\sigma(a_1, a_2, \dots, a_n) \in \mathbf{Ker}(f, g)_s. \quad \square \end{aligned}$$

Corolar 1.3.14 Fie $f : \mathcal{A} \longrightarrow \mathcal{B}$ și $g : \mathcal{A} \longrightarrow \mathcal{B}$ două morfisme și X o submulțime a lui \mathcal{A} . Dacă restricțiile lui f și g la X coincid, atunci restricțiile lui f și g la \overline{X} sunt egale.

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

Corolar 1.3.15 Fie $f : \mathcal{A} \longrightarrow \mathcal{B}$ și $g : \mathcal{A} \longrightarrow \mathcal{B}$ două morfisme. Dacă restricțiile lui f și g la o mulțime de generatori ai algebrei \mathcal{A} coincid, atunci $f = g$.

Demonstrație: Fie X o mulțime de generatori ai algebrei \mathcal{A} , adică $\overline{X} = \mathcal{A}$. Presupunem că restricțiile lui f și g la X sunt egale. Din corolarul anterior deducem că restricțiile lui f și g la \overline{X} coincid. Dar $\overline{X} = \mathcal{A}$ implică $f = g$.

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

Demonstrație:

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

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

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

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

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

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

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

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

3. Din $X \subseteq \overline{X}$, 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 proprietății 2 o parte stabilă a lui \mathcal{A} deducem $\overline{X} \subseteq h^{-1}(\overline{h(X)})$. Deoarece imaginea directă este crescătoare $h(\overline{X}) \subseteq h(h^{-1}(\overline{h(X)}))$. 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 EXISTENȚA ALGEBRELOR LIBERE

1.4.1 Algebre libere și algebre Peano

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

Fie M un monoid și $B \subseteq M$. Reamintim două definiții echivalente pentru faptul că **monoidul M este liber generat de submulțimea sa B** .

Definiția 1.4.1 Pentru orice monoid N și orice funcție $f : B \rightarrow N$ există un unic morfism $f^\# : M \rightarrow N$ de monoizi a cărui restricție la B este f .

Definiția 1.4.2 Pentru orice $m \in M$ există și sunt unice numărul natural n și elementele $b_1 \in B$, $b_2 \in B$, \dots , $b_n \in B$ cu proprietatea $m = b_1 b_2 \dots b_n$.

Să observăm diferența esențială dintre cele două definiții echivalente. Observăm că în definiția 1.4.1 nu apar de loc elemente, apărând numai concepte din afara monoidului M . Definiția 1.4.2 în schimb lucrează numai cu elemente din interiorul monoidului.

Comparând definiția dată algebrelor libere cu cele două de mai sus constatăm că definiția algebrelor libere este asemănătoare cu definiția 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 definiția 1.4.2 este cel de algebră Peano, concept echivalent, așa cum am spus mai sus, cu cel de algebră liberă.

Problema esențială este de a demonstra existența 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 și apoi proba că algebrele Peano sunt libere. Deoarece expresiile pot fi scrise în mai multe moduri, fiecare dintre aceste scrieri poate conduce la o demonstrație a existenței algebrei Peano. Deoarece textul se adresează în primul rând unor informaticieni vom prefera reprezentarea expresiilor ca arborii etichetați, local ordonați, în care frunzele sunt etichete cu variabile(generatori) sau nume de constante. Restul nodurilor sunt etichetate cu nume de operații ale căror argumente sunt date de subarborii având rădăcinile drept succesori ai nodului.

Existența algebrelor libere - varianta scurtă

Definiția 1.4.3 Se numește **mulțime S -sortată de variabile** o mulțime S -sortată $X = \{X_s\}_{s \in S}$ cu componentele disjuncte două câte două.

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

Varianta scurtă este scrisă pentru cei care au dificultăți în a înțelege o demonstrație corectă dar mai dificilă. Cei care preferă varianta scurtă pot sării secțiunea următoare. Ceilalți sunt invitați să sară direct la secțiunea următoare.

În această variantă expresiile sunt scrise folosind scrierea uzuală cu paranteze și virgule. Plecăm de la o semnătură (S, Σ) și o mulțime S -sortată de variabile X .

Fie $T_\Sigma(X)$ **cea mai mică** mulțime S -sortată cu următoarele proprietăți:

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

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

1. $T_\sigma = \sigma$ dacă $\sigma \in \Sigma_{\lambda, s}$
2. $T_\sigma(t_1, t_2, \dots, t_n) = \sigma(t_1, t_2, \dots, t_n)$ dacă $n \geq 1$, $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ și $t_i \in T_\Sigma(X)_{s_i}$ 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 și $f : X \longrightarrow A$ o funcție S -sortată.

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

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

Din prima parte a definiției de mai sus se vede că restricția funcției $f^\#$ la X este chiar funcția f .

Din celelalte două părți ale definiției 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 restricție la X este f , deoarece X este o mulțime de generatori ai lui $T_\Sigma(X)$ și deoarece h și $f^\#$ coincid pe X deducem din cololarul 1.3.15 că $h = f^\#$.

Unde-i greșeala?

1.4.2 Algebre Peano

Definiția 1.4.4 O Σ algebră $\mathcal{A} = (A_s, A_\sigma)$ se numește Peano peste $X \subseteq A$ dacă

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

Teorema 1.4.5 *Orice algebră Peano peste X este liber generată de X .*

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

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

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

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

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

Definim prin inducție după $n \in N$ șirul de funcții $h^n : X^n \longrightarrow B$ prin $h^0 = h$ și

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

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

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

Definim funcția $g : A \longrightarrow B$ pentru orice $s \in S$ și 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 $s \in S$ și 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 $\sigma \in \Sigma_{w,s}$ și $a \in A_w$.

Deoarece conform condiției 2 din definiția 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 $\sigma' \in \Sigma_{w',s}$ și $a' \in X_{w'}^n$, astfel încât $A_\sigma(a) = A_{\sigma'}(a')$. Rezultă conform condiției 3 din definiția algebrei Peano că $w = w'$, $\sigma = \sigma'$ și $a = a'$, deci $a \in X_w^n$. Prin urmare

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

deci

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

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

Unicitatea lui g este consecința faptului că X generează \mathcal{A} . \square

Am putea spune că această teoremă este partea comună a tuturor demonstrațiilor privind existența algebrelor libere. Rolul acestei teoreme este de a reduce demonstrația existenței algebrelor libere la existența algebrelor Peano. Cu alte cuvinte demonstrația de natură semantică este redusă la aspectele ei sintactice.

Menționăm că propoziția 1.4.8 este reciproca teoremei de mai sus.

1.4.3 Algebra arborilor de derivare

O gramatică independentă de context posedă două mulțimi disjuncte, una a neterminalelor N și una a terminalelor T precum și mulțimea $P \subseteq N \times (N \cup T)^*$ a producțiilor. Chiar dacă conceptul de gramatică independentă de context verifică și alte condiții, 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 atașa o semnătură:

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

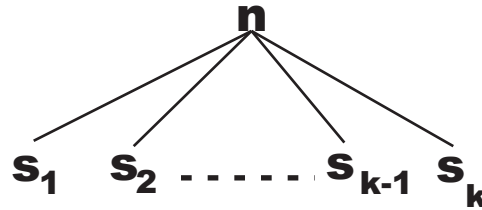
$$(n, t_0 n_1 t_1 \dots n_k t_k) : n_1 n_2 \dots n_k \longrightarrow n$$

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

O algebră a cărei semnătură este cea atașată gramaticii independente de context G se numește G -algebră.

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

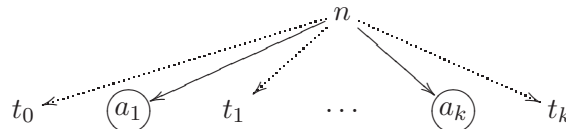
1. este un arbore finit și local ordonat, adică succesorii fiecărui nod sunt într-o ordine totală;
2. are noduri etichetate cu terminale sau neterminale, rădăcina fiind etichetată cu un neterminal;
3. orice nod etichetat cu un terminal nu are nici un succesor;
4. pentru orice nod, dacă este etichetat cu un neterminal n , atunci perechea formată din n și cuvântul format de etichetele succesorilor formează o producție $(n, s_1 s_2 \dots s_{k-1} s_k) \in P$.



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

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

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



Teorema 1.4.6 *Algebra arborilor de derivare ai unei gramaticii independente de context este algebră Peano peste mulțimea vidă.*

Demonstrație: Pentru a proba că \mathcal{A} este generată de mulțimea 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 inducție după adâncimea arborilor că orice arbore este în P . Reamintim că adâncimea unui arbore este lungimea celei mai lungi ramuri din arbore. Aici este esențială finitudinea arborilor de derivare deoarece aceasta implică existența adâncimii oricărui arbore. Fie m un număr natural. Presupunem prin ipoteza de inducție 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 și primul nivel al arborelui a deducem existența unei producții $p = (n, t_0 n_1 t_1 \dots n_k t_k)$ cu proprietatea

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

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

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

A doua condiție din definiția algebrelor Peano

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

este evident adevărată.

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

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

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

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

Reamintim că prin definiția gramaticilor un element nu poate fi în același timp și terminal și neterminal. Deoarece numărul neterminalelor din cele două cuvinte trebuie să fie egal deducem egalitatea $k = k'$. Deoarece primele neterminale din cele două cuvinte trebuie să fie pe aceeași poziție deducem că $t_0 = t'_0$ și $n_1 = s_1$. Rezultă că

$$t_1 n_2 t_2 \dots n_k t_k = t'_1 s_2 t'_2 \dots s_k t'_k.$$

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

În final egalând subarborii cu rădăcinile aflate pe aceeași poziție a primului nivel din arborii egali $A_p(a_1, a_2, \dots, a_k)$ și $A_q(b_1, b_2, \dots, b_{k'})$ rezultă că $a_i = b_i$ pentru orice $1 \leq i \leq k$. \square

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

1.4.4 Existența algebrelor libere

Urmărim să arătăm că pentru orice semnătură Σ și pentru mulțime S -sortată de variabile X există o Σ -algebră liber generată de X pe care în cele ce urmează o vom nota cu $T_\Sigma(X)$.

Construcția algebrei libere care urmează este bazată pe scrierea poloneză a expresiilor, adică un șir de semne de operații și variabile în care semnul de operație este plasat în fața argumentelor sale care trebuie să-l urmeze. Pentru generarea acestor expresii vom folosi o gramatică independentă de context. Producțiile de forma (s, x) unde $s \in S$ și $x \in X_s$ spun că orice variabilă x de sort s este expresie de sort s . Producțiile de forma $(s, \sigma s_1 s_2 \dots s_n)$ unde $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$ spun că dacă e_i este expresie de sort s_i pentru orice $1 \leq i \leq n$, atunci $\sigma e_1 e_2 \dots e_n$ este expresie de sort s .

Fie (S, Σ) o semnătură și X o mulțime S -sortată cu componentele disjuncte două câte două. Fără a micșora generalitatea vom presupune că Σ și X sunt disjuncte.

Considerăm gramatica independentă de context definită prin

1. Mulțimea neterminalelor este S ,
2. Mulțimea terminalelor este $\Sigma \cup X$ și
3. Mulțimea producțiilor este $\{(s, \sigma w) | w \in S^*, s \in S \text{ și } \sigma \in \Sigma_{w, s}\} \cup \{(s, x) | s \in S, x \in X_s\}$.

Notăm cu $\mathcal{A} = (A_s, \{A_{(s, \sigma w)}\}_{\sigma \in \Sigma_{w, s}}, \{A_{(s, x)}\}_{s \in S, x \in X_s})$ algebra arborilor ei de derivare. Ea este, conform teoremei 1.4.6, algebră Peano peste mulțimea vidă.

Notăm cu $\mathcal{B} = (A_s, B_\sigma)$ algebra de semnătură Σ definită prin $B_\sigma = A_{(s, \sigma w)}$ pentru orice $w \in S^*$, $s \in S$ și $\sigma \in \Sigma_{w, s}$. Notăm cu $i : X \rightarrow A$ funcția S -sortată definită pentru orice $s \in S$ și $x \in X_s$ prin $i_s(x) = A_{(s, x)}$. Observăm, folosind a doua condiție din definiția algebrelor Peano, că funcția i are toate componentele injective.

Teorema 1.4.7 Σ -algebra \mathcal{B} este Peano peste $i(X)$.

Demonstrație: Verificăm cele trei condiții din definiția algebrelor Peano.

1. Fie $C \subseteq A$ o parte stabilă a algebrei \mathcal{B} care include $i(X)$. Observăm că C este o parte stabilă a algebrei \mathcal{A} :

- a) C fiind o parte stabilă a algebrei \mathcal{B} , C este închisă la operațiile B_σ , adică la operațiile $A_{(s, \sigma w)}$ pentru orice $w \in S^*$, $s \in S$ și $\sigma \in \Sigma_{w, s}$;
- b) C include $i(X)$, adică $A_{(s, x)} \in C_s$ pentru orice $s \in S$ și $x \in X_s$.

Deoarece \mathcal{A} este Peano peste mulțimea vidă rezultă că $C = A$.

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

2. Fie $w \in S^*$, $s \in S$, $\sigma \in \Sigma_{w, s}$ și $a \in A_w$. Deoarece pentru orice $s' \in S$ și $x \in X_{s'}$, primul succes al rădăcinilor arborilor $A_{(s, \sigma w)}(a)$ și $A_{(s', x)}$ este etichetat cu σ , respectiv x care sunt diferite deducem $A_{(s, \sigma w)}(a) \neq A_{(s', x)}$.

Rezultă că $B_\sigma(a) \notin i(X)$.

3. Fie $\sigma \in \Sigma_{w, s}$, $a \in A_w$, $\sigma' \in \Sigma_{w', s}$ $a' \in A_{w'}$ cu $B_\sigma(a) = B_{\sigma'}(a')$. Din

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

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

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

Propoziție 1.4.8 Orice algebră liberă este Peano

Demonstrație: Fie \mathcal{L} o Σ -algebră liber generată de mulțimea X . Considerăm o Σ -algebră \mathcal{P} Peano peste X . Deoarece și \mathcal{P} este liber generată de X rezultă existența unui izomorfism $i : \mathcal{P} \rightarrow \mathcal{L}$ 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 INIȚIALE

Menționăm aici pe marele informatician, regretatul Joseph Goguen, profesorul și prietenul multor români, care spunea că “semantica algebrei inițiale” este una dintre cele mai frumoase idei ale sale [11].

Metoda semanticii algebrei inițiale este o simplificare a metodei mai clasice a semanticii denotaționale, numită și semantică matematică.

Metoda semanticii algebrei inițiale se aplică pentru limbaje definite printr-o gramatică independentă de context $G = (N, T, P, a)$, unde N este mulțimea neterminalelor, T mulțimea terminalelor, P mulțimea producțiilor și 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 înțelege afirmația de mai sus trebuie să intrăm puțin în amănunte. Fie \mathcal{A} algebra arborilor de derivare. Deoarece \mathcal{A} este algebră inițială 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 definiție $M_a(arb)$.

Menționăm 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 și mai mult ideile de mai sus.

1.5.1 Semantica unui șir de cifre ca număr natural

Vom considera o gramatică G care generează șirurile finite de cifre zecimale, considerate ca terminale. Gramatica are două neterminale $\langle \text{cifra} \rangle$ și $\langle \text{nat} \rangle$ ultima fiind și axioma a gramaticii. Descriem în continuare producțiile gramaticii cărora le dăm un nume

$$\begin{array}{lll} ci & \langle \text{cifra} \rangle & \longrightarrow i \\ n1 & \langle \text{nat} \rangle & \longrightarrow \langle \text{cifra} \rangle \\ n2 & \langle \text{nat} \rangle & \longrightarrow \langle \text{nat} \rangle \langle \text{cifra} \rangle \end{array} \quad \text{pentru orice cifră zecimală } i$$

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 mulțimi. Semantica unei cifre, care nu este nimic altceva decât un semn, este numărul reprezentat de cifră. Prin urmare suportul corespunzător neterminalului $\langle \text{cifra} \rangle$ trebuie să conțină măcar numerele de la zero la nouă. Deci prin definiție

$$S_{\langle \text{cifra} \rangle} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

Deoarece semantica unui șir finit de cifre este numărul natural pe care acest șir îl reprezintă, suportul corespunzător neterminalului $\langle \text{nat} \rangle$ trebuie să conțină măcar mulțimea numerelor naturale. Deci prin definiție

$$S_{\langle \text{nat} \rangle} = \text{este mulțimea numerelor naturale.}$$

Să trecem la definirea operațiilor algebrei semantice.

Deoarece producția ci corespunzătoare cifrei i nu are nici un neterminal în membrul drept, operația corespunzătoare ei este o operație fără argumente, adică un element din $S_{\langle \text{cifra} \rangle}$

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

Producția $n1$ spune că orice cifră este un șir de cifre. Deoarece valoarea numărului reprezentat de un șir de lungime unu este egală cu valoarea numărului reprezentat de singura cifră din șir și semantica lor trebuie să fie aceeași. Deci prin definiție

$$S_{n1} \text{ este funcția incluziune de la } S_{\langle \text{cifra} \rangle} \text{ la } S_{\langle \text{nat} \rangle}.$$

Producția n2 ne spune că prin adăugarea la dreapta unui șir de cifre, să spunem s1, a unei cifre, să spunem c, se obține un nou șir de cifre s2 = s1c. Operația corespunzătoare producției n2 trebuie să fie o funcție

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

definită pentru numele n și m prin

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

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

Cu aceasta semantica este complet definită. Vom exemplifica pentru șirul 023. Arborele de derivare pentru acest șir este

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

Să-i aplicăm morfismul semantic acestui arbore.

$$\begin{aligned} M_{<nat>}(A_{n2}(A_{n2}(A_{n1}(A_{c0}), A_{c2}), A_{c3})) &= S_{n2}(S_{n2}(S_{n1}(S_{c0}), S_{c2}), S_{c3}) = S_{n2}(S_{n2}(S_{n1}(0), 2), 3) = \\ &= S_{n2}(S_{n2}(0, 2), 3) = S_{n2}(10 \times 0 + 2, 3) = S_{n2}(2, 3) = 10 \times 2 + 3 = 23 \end{aligned}$$

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

1.5.2 Un calculator de buzunar

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

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

Remarcăm că butoanele diferite de ON, OFF și E sunt folosite pentru introducerea expresiilor care vor fi evaluate la apăsarea butonului E. Sintaxa expresiilor este definită de producțiile r1-r6 ale gramaticii de mai jos. Expresiile sunt construite pornind de la șirurile finite de cifre zecimale reprezentând numere naturale (producția r1), variabila M (producția r2) și recursiv, alte expresii cuprinse între paranteze (producția r6). Pentru construcția expresiilor sunt folosite numai operațiile de adunare și înmulțire (producțiile r3 și r4). Producția r5 permite construirea unor expresii condiționate unde condiția 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 porțiuni de programe și $< program >$ care este și axiomă a gramaticii. Descriem în continuare producțiile gramaticii cărora le dăm un nume pe care l-am scris la începutul rândului, în fața producției.

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 funcționează calculatorul. Se pornește calculatorul apăsând butonul ON, moment în care se inițializează unica celulă de memorie cu zero. Se introduce o expresie care la apăsarea butonului E se evaluează. Rezultatul evaluării este afișat pe ecran și introdus în unica celulă de memorie în locul vechii valori. Se introduce altă expresie, se apasă butonul E și așa 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 mulțimi, primele două fiind cele de mai sus $S_{<cifra>}$ și $S_{<nat>}$.

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 obține prin evaluarea expresiei M+3. Trebuie să menționăm că fiecare apariție 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 conținutul celulei de memorie. Prin urmare semantica expresiei M+3 este funcția $f : N \longrightarrow N$ definită 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 definiție $S_{<exp>}$ este mulțimea funcțiilor de la mulțimea numerelor naturale la ea însăși.

O instrucțiune, generată de producțiile r7 și r8, comandă evaluarea unui șir finit de expresii. Semantica instrucțiunii este, prin definiție, șirul numerelor care apar pe ecran în timpul execuției instrucțiunii, adică un șir finit și nevid de numere naturale. Mulțimea acestor șiruri o notăm cu N^+ . Rezultatul evaluării depinde evident de valoarea din memorie din momentul începerii execuției instrucțiunii, prin urmare

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

Un program, generat de producția r9, diferă de o instrucțiune prin faptul că se introduce zero în celula de memorie înainte de execuția instrucțiunii pe care o conține, prin urmare șirul de numere care apar pe ecran în timpul execuției programului nu mai depinde de conținutul memoriei, fiind un element din N^+ , deci $S_{<program>} = N^+$.

Folosind următoarele notații

m este un număr natural reprezentând conținutul celulei de memorie

n este un număr natural

e este o funcție de la N la N reprezentând semantica unei expresii

i este o funcție de la N la N^+ reprezentând semantica unei instrucțiuni

definițiile pentru operații, fără a le mai repeta pe cele de mai sus sunt:

$$S_{r1}(n)(m) = n$$

$$S_{r2}(m) = m$$

$$S_{r3}(e1, e2)(m) = e1(m) + e2(m)$$

$$S_{r4}(e1, e2)(m) = e1(m) \times e2(m)$$

$$S_{r5}(e1, e2, e3)(m) = \text{dacă } e1(m) = 0 \text{ atunci } e2(m) \text{ altfel } e3(m)$$

$$S_{r6}(e) = e$$

$$S_{r7}(e)(m) = e(m)$$

$$S_{r8}(e, 1)(m) = e(m)1(e(m))$$

$$S_{r9}(1) = 1(0)$$

1.5.3 Arbori de derivare

Reamintim proprietățile unui arbore de derivare dintr-o gramatică independentă de context:

1. este un arbore finit și 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 și cuvântul format de etichetele s_1, s_2, \dots, s_k ale succesorilor săi formează o producție $(n, s_1 s_2 \dots s_{k-1} s_k) \in P$.

O gramatică pentru expresii

Definim o gramatică independentă de context pentru expresiile construite cu variabilele x, y și z , cu operațiile binare de adunare și înmulțire și cu paranteze. Gramatica trebuie construită respectând următoarele condiții privind expresiile:

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

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

V reprezintă variabilele

F de la factor, reprezintă cele mai “mici” elemente folosite în construcția expresiilor

T de la termen, reprezintă un produs de factori

E de la expresie, reprezintă o sumă de termeni

P de la program.

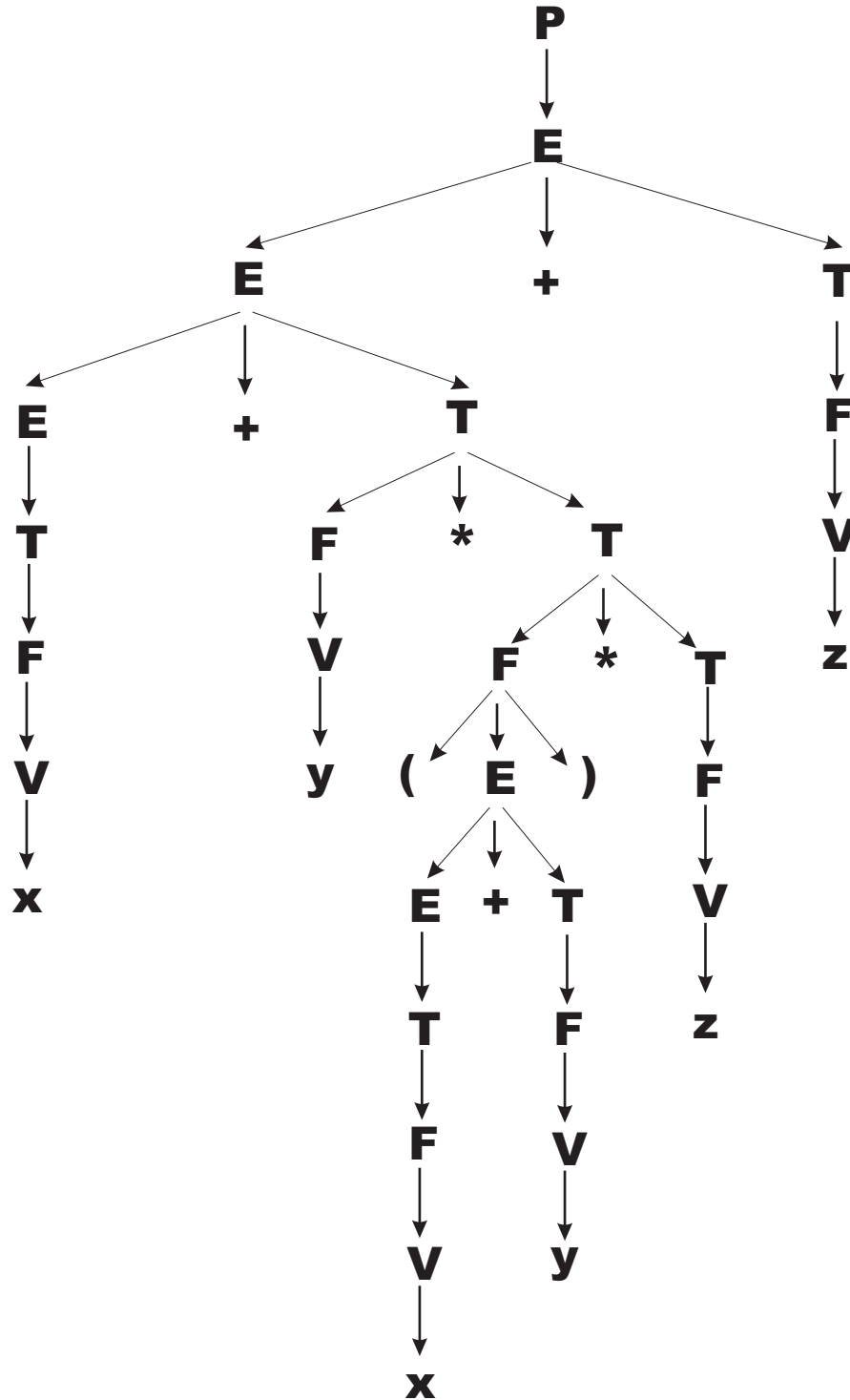
Continuăm cu mulțimea **regulilor (producțiilor)** gramaticii:

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

Observați producțiile 7 și 9 pentru a vedea cum se precizează ordinea de execuție a operațiilor de același fel. Neterminalul din membrul stâng se află în dreapta, respectiv în stânga semnului operației.

Un exemplu

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



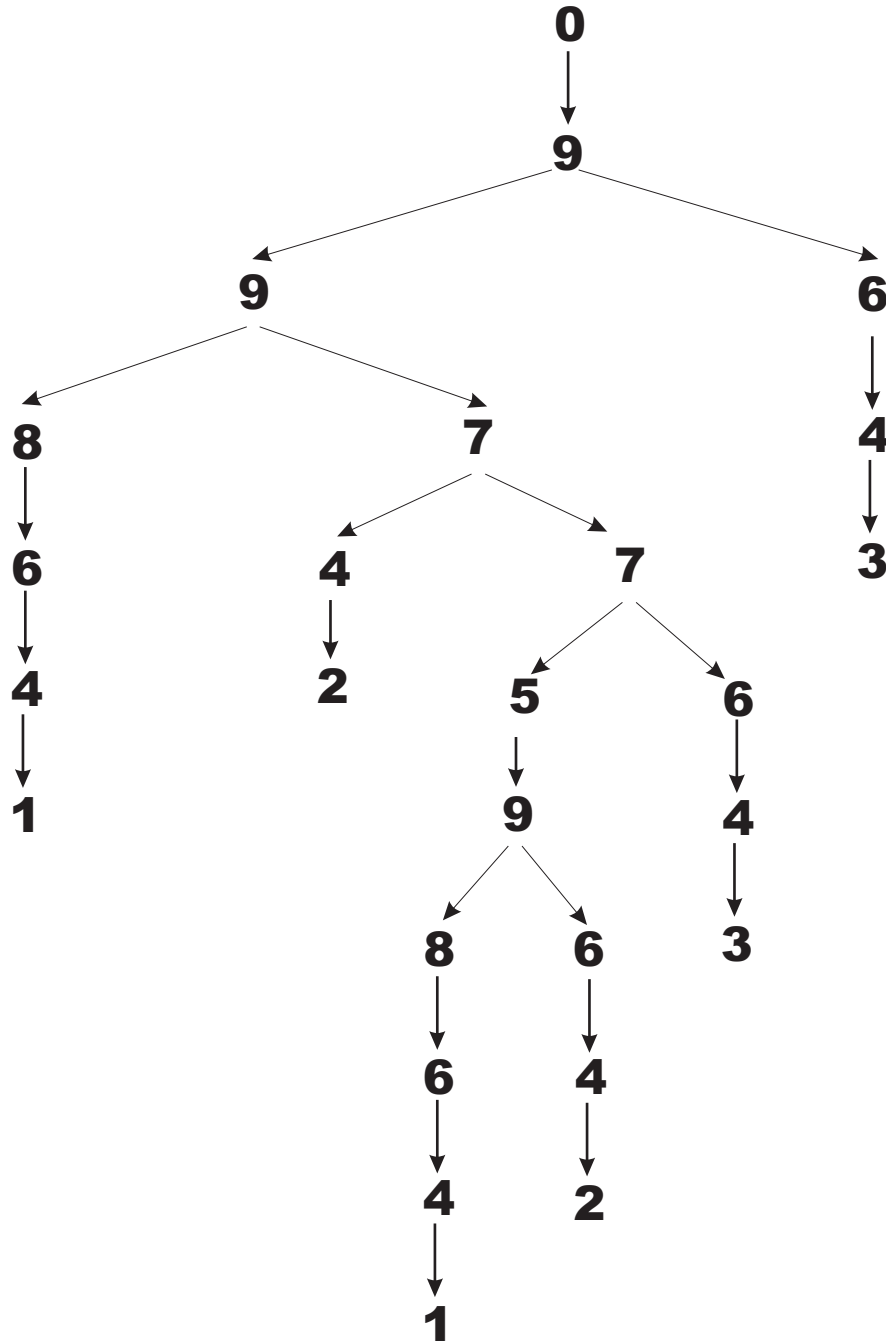
Pentru a reface expresia, arborele se parcurge în inordine.

Dacă veți încerca să construiți alt arbore de derivare pentru aceeași expresie veți constata că acest lucru nu este posibil. Prin urmare modul de construcție al expresiei va impune o anumită ordine în evaluarea acesteia.

Folosind numerele care denumesc producțiile ca nume ale operațiilor 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 $a = 8(6[4(1)])$ și $b = 6(4(3))$.



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 secțiunea următoare.

Algebra semantică va avea toate suporturile egale cu semigrupul liber generat de $\{x, y, z, +, *\}$, sau mai popular mulțimea șirurilor finite nevide de semne din $\{x, y, z, +, *\}$. Operațiile sunt următoarele:

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

$1 = x$, $2 = y$ și $3 = z$,

$$7(\alpha, \beta) = \alpha\beta* \text{ și } 9(\alpha, \beta) = \alpha\beta+.$$

Notăm cu \mathcal{S} algebra semantică de mai sus, cu \mathcal{A} algebra arborilor de derivare și 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 inițiale rezultă că $C(A)$ este scrierea poloneză inversă a expresiei $x + y * (x + y) * z + z$.

Calcululele sunt următoarele

$$C(a) = 8(6[4(1)]) = x \quad \text{și} \quad C(b) = 6(4(3)) = z$$

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

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

Rezultatul final confirmă corectitudinea și eficacitatea metodei.

1.5.5 Compile

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

Vom pune în evidență elementele care apar în programele obținute 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 crește automat cu o unitate. Ori de câte ori se scoate un element din stivă pointerul scade automat cu o unitate. Presupunem că la începutul execuției programului pointerul indică primul loc liber din stivă.

Presupunem că limbajul de asamblare include următoarele instrucțiuni:

Ad R scoate valoarea din vârful stivei, o adună cu conținutul lui R și pune rezultatul în R
Mu R scoate valoarea din vârful stivei, o înmulțește cu conținutul lui R și pune rezultatul în R
ld R scoate valoarea din vârful stivei și o pune în R
st R valoarea din R este pusă în vârful stivei
print tipărește valoarea din vârful stivei

Semantica unei expresii va fi programul care evaluează expresia și imprimă rezultatul evaluării. Programul care evaluează o expresie are ca efect plasarea în vârful stivei a rezultatului evaluării acesteia și avansarea pointerului. Pentru definirea acesteia se va folosi metoda semanticii algebrei inițiale.

Toate cele 5 suporturi ale algebrei semantice, corespunzătoare neterminalelor, sunt identice și coincid cu mulțimea bucăților de program (șiruri de instrucțiuni separate prin ;). Astfel de bucăți de programe vor fi notate cu litere grecești.

Cele 10 operații ale algebrei semantice, corespunzătoare producțiilor sunt:

$$\begin{aligned} 1_S &= \text{st } x \\ 2_S &= \text{st } y \\ 3_S &= \text{st } z \\ 4_S(\alpha) &= \alpha \\ 5_S(\alpha) &= \alpha \\ 6_S(\alpha) &= \alpha \\ 7_S(\alpha, \beta) &= \alpha ; \beta ; \text{ld } R ; \text{Mu } R ; \text{st } R \\ 8_S(\alpha) &= \alpha \\ 9_S(\alpha, \beta) &= \alpha ; \beta ; \text{ld } R ; \text{Ad } R ; \text{st } R \\ 0_S(\alpha) &= \alpha ; \text{print} \end{aligned}$$

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

Conform metodei semanticii algebrei inițiale rezultă că $C(A)$ este programul care evaluează și tipărește rezultatul pentru expresia $x + y * (x + y) * z + z$.

Program

Prezentăm calculele care dovedesc afirmația de mai sus.

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

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

$$\begin{aligned} C(c) &= 7_S[5_S\{9_S(1_S, 6_S[4_S(2_S)])\}, 3_S] = \\ &7_S[9_S(1_S, 2_S), 3_S] = \\ &9_S(1_S, 2_S), 3_S ; \text{ld R} ; \text{Mu R} ; \text{st R} = \\ &1_S ; 2_S ; \text{ld R} ; \text{Ad R} ; \text{st R} ; 3_S ; \text{ld R} ; \text{Mu R} ; \text{st R} = \\ &\text{st x} ; \text{st y} ; \text{ld R} ; \text{Ad R} ; \text{st R} ; \text{st z} ; \text{ld R} ; \text{Mu R} ; \text{st R} \end{aligned}$$

Deoarece

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

deducem

$$\begin{aligned} C(A) &= 0_S(9_S[9_S\{1_S, 7_S(2_S, C(c))\}, 3_S]) = \\ &9_S[9_S\{1_S, 7_S(2_S, C(c))\}, 3_S] ; \text{print} = \\ &9_S\{1_S, 7_S(2_S, C(c))\} ; 3_S ; \text{ld R} ; \text{Ad R} ; \text{st R} ; \text{print} = \\ &1_S ; 7_S(2_S, C(c)) ; \text{ld R} ; \text{Ad R} ; \text{st R} ; 3_S ; \text{ld R} ; \text{Ad R} ; \text{st R} ; \text{print} = \\ &1_S ; 2_S ; C(c) ; \text{ld R} ; \text{Mu R} ; \text{st R} ; \text{ld R} ; \text{Ad R} ; \text{st R} ; 3_S ; \text{ld R} ; \text{Ad R} ; \text{st R} ; \text{print} = \end{aligned}$$

$$\begin{aligned} &\text{st x} ; \text{st y} ; \text{st x} ; \text{st y} ; \text{ld R} ; \text{Ad R} ; \text{st R} ; \text{st z} ; \text{ld R} ; \text{Mu R} ; \text{st R} ; \text{ld R} ; \text{Mu R} ; \text{st R} ; \text{ld R} ; \text{Ad R} \\ &; \text{st R} ; \text{st z} ; \text{ld R} ; \text{Ad R} ; \text{st R} ; \text{print} \end{aligned}$$

Optimizare cod

Programul de mai sus poate fi simplificat. Observăm că grupul de instrucțiuni “st R ; ld R” are efect cumulativ nul ceea ce conduce la programul

$$\begin{aligned} &\text{st x} ; \text{st y} ; \text{st x} ; \text{st y} ; \text{ld R} ; \text{Ad R} ; \text{st R} ; \text{st z} ; \text{ld R} ; \text{Mu R} ; \\ &\text{Mu R} ; \text{Ad R} ; \text{st R} ; \text{st z} ; \text{ld R} ; \text{Ad R} ; \text{st R} ; \text{print} \end{aligned}$$

1.6 CONGRUENȚE ȘI ALGEBRE CÂT

Dacă $f : A \longrightarrow B$ este o funcție relația $\text{Ker}(f)$ definită prin

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

se numește **echivalența nucleară** a lui f , sau mai pe scurt nucleul lui f .

Echivalența nucleară a unei funcții este o relație de echivalență. În cazul multisortat nucleul este luat pe componente, adică pentru fiecare sort în parte.

1.6.1 Congruențe

O relație de echivalență într-o Σ -algebră (A_s, A_σ) este de fapt o familie de relații de echivalențe, câte una pentru fiecare mulțime A_s . O congruență este o relație de echivalență care este compatibilă cu operațiile algebrei. De fapt compatibilitatea trebuie cerută numai pentru operațiile cu argumente căci pentru o constantă $A_\sigma \sim A_\sigma$ rezultă din reflexivitate.

Definiția 1.6.1 O congruență $\sim = \{\sim_s\}_{s \in S}$ în Σ -algebra (A_s, A_σ) este o relație de echivalență 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 congruență este relația de egalitate. Cea mai mare congruență este relația totală.

Propoziție 1.6.2 Dacă $h : \mathcal{A} \longrightarrow \mathcal{B}$ este un Σ -morfism, atunci $\text{Ker}(h)$ este o congruență.

Demonstrație: Fie $s_1 s_2 \dots s_n \in S^*$, $s \in S$ și $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$. Presupunem că $a_i \text{Ker}(h_{s_i}) b_i$ 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

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

așadar $A_\sigma(a_1, a_2, \dots, a_n) \text{Ker}(h_s) A_\sigma(b_1, b_2, \dots, b_n)$. Deci $\text{Ker}(h)$ este congruență.

Propoziție 1.6.3 Orice intersecție de congruențe este tot o congruență.

Demonstrație: Fie $\{\sim^k\}_{k \in K}$ o mulțime de congruențe și

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

Probăm că \sim este congruență.

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

Pentru orice $k \in K$ deoarece \sim^k este congruență, 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

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

Această propoziție ne spune că mulțimea congruențelor unei algebre este o familie Moore. Încercați să caracterizați operatorul de închidere asociat.

Congruențe de grupuri - numai pentru cei știu ce este un subgrup normal

Grupurile sunt privite ca algebre cu trei operații: una binară \times , una unară $*$ și o constantă e . De ce? Există cel puțin două motive.

a) Motivul principal este dispariția cuantificatorului existențial 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 ecuațional 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 condiții se impun pentru conceptul de congruență

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

Practic rămâne numai una deoarece a doua condiție este o consecință a primeia: presupunând $a \sim b$, folosind reflexivitatea din $a^* \sim a^*$, $a \sim b$ și $b^* \sim b^*$ deducem $a^* \times a \times b^* \sim a^* \times b \times b^*$, prin urmare $b^* \sim a^*$, deci $a^* \sim b^*$.

Propoziție 1.6.4 *În orice grup conceptul de congruență este echivalent cu cel de subgrup normal*

Demonstrație:

1. Dată o congruență clasa elementului neutru este un subgrup normal.
2. Dat un subgrup normal N relația definită prin

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

este o congruență.

3. Cele două treceri, de la un concept la celălalt și reciproc, stabilesc o bijecție între mulțimea congruențelor unui grup și mulțimea subgrupurilor sale normale. \square

Congruențe de inele - numai pentru cei care știu ce este un ideal

Pentru simplitate vom prefera cazul comutativ.

Signatura conceptului de inel conține pe lângă cele trei operații 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 condiția de congruență numai pentru cele două operații binare.

Propoziție 1.6.5 *În orice inel comutativ conceptul de congruență este echivalent cu cel de ideal.*

Demonstrație:

- 1 Dată o congruență clasa elementului neutru pentru adunare este un ideal.
- 2 Dat un ideal N relația definită prin

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

este o congruență.

3. Cele două treceri, de la un concept la celălalt și reciproc, stabilesc o bijecție între mulțimea congruențelor unui inel și mulțimea idealelor sale. \square

Concluzie

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

1.6.2 Algebre cât

Proprietatea de universalitate a mulțimii cât.

Propoziție 1.6.6 *Fie \sim o relație de echivalență în mulțimea A . Fie A/\sim mulțimea cât și $\rho : A \longrightarrow A/\sim$ surjecția naturală de factorizare. Pentru orice funcție $f : A \longrightarrow B$ dacă $\sim \subseteq \text{Ker}(f)$, atunci există o unică funcție $f^\# : A/\sim \longrightarrow B$ cu proprietatea $\rho; f^\# = f$.*

Demonstrație: Definim funcția $f^\#$ pentru orice $a \in A$ prin

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

Definiția este corectă deoarece $b \sim a$ implică, folosind ipoteza $\sim \subseteq \text{Ker}(f)$, că $b \text{Ker}(f) a$, adică $f(a) = f(b)$.

Egalitatea din enunț $\rho; f^\# = f$ rezultă direct din definiție.

Unicitatea rezultă din surjectivitatea funcției ρ de factorizare. \square

Proprietatea de universalitate a algebrei cât.

Fie (A_s, A_σ) o Σ -algebră și \sim o congruență. Definim operațiile algebrei cât

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

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

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

Teorema 1.6.7 *Pentru orice morfism de Σ -algebre $f : \mathcal{A} \longrightarrow \mathcal{B}$ dacă $\sim \subseteq \text{Ker}(f)$, atunci există un unic morfism de Σ -algebre $f^\# : A/\sim \longrightarrow \mathcal{B}$ cu proprietatea $\rho; f^\# = f$.*

Demonstrație: Din proprietatea de universalitate a mulțimii cât deducem existența unei unice funcții $f^\# : A/\sim \longrightarrow B$ cu proprietatea $\rho; f^\# = f$. Observăm că $f_s^\#(\rho(a)) = f_s(a)$ pentru orice $s \in S$ și $a \in A_s$. Mai trebuie să demonstrăm că funcția $f^\#$ este un morfism de Σ -algebre.

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

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

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

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

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

1.7 ALGEBRE PROIECTIVE**1.7.1 Proiectivitatea algebrelor libere**

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

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

Demonstrație: Fie \mathcal{A} , \mathcal{B} două Σ -algebre și X o mulțime S -sortată de variabile. Dacă $e : \mathcal{A} \longrightarrow \mathcal{B}$ este un morfism cu toate componentele surjective și $f : T_\Sigma(X) \longrightarrow \mathcal{B}$ 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 acțiunea lui pe variabile.

Fie $s \in S$ și $x \in X_s$. 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 $s \in S$ și orice $x \in X_s$. Este evident faptul că $(g; e)_s(x) = f_s(x)$ pentru orice $s \in S$ și orice $x \in X_s$. Deoarece morfismele $g; e$ și f coincid pe generatorii algebrei libere $T_\Sigma(X)$ rezultă că $g; e = f$ și demonstrația 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 proprietăți

Cei care nu sunt interesați de toate detaliile pot sări această secțiune, dar în continuare pot lua drept definiție pentru Σ -algebre: epimorfism = morfism cu toate componentele surjective.

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

Propoziție 1.7.3 *Fie \mathcal{A} și \mathcal{B} două (S, Σ) -algebre și $h : \mathcal{A} \longrightarrow \mathcal{B}$ un (S, Σ) -morfism. Morfismul h este epimorfism în Alg_Σ dacă și numai dacă h_s este surjectiv pentru orice $s \in S$.*

Demonstrație: Fie $\mathcal{A} = (\{A_s\}_{s \in S}, \{A_\sigma\}_{\sigma \in \Sigma})$ și $\mathcal{B} = (\{B_s\}_{s \in S}, \{B_\sigma\}_{\sigma \in \Sigma})$.

(\Leftarrow) Fie \mathcal{C} o (S, Σ) -algebră și $m, n : \mathcal{B} \longrightarrow \mathcal{C}$ două morfisme astfel încât $h; m = h; n$. Rezultă că $h_s; m_s = h_s; n_s$ pentru orice $s \in S$. Deoarece pentru orice $s \in S$, h_s este surjectiv rezultă că h_s este epimorfism în categoria mulțimilor, deci $m_s = n_s$. Am demonstrat că $m = n$.

(\Rightarrow) Fie $\mathcal{C} = \{C_s\}_{s \in S}$ o mulțime 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 mulțimi.

Fie $\iota : \mathcal{B} \longrightarrow \mathcal{C}$ funcția incluziune de mulțimi S -sortate și $r : \mathcal{B} \longrightarrow \mathcal{C}$ definit prin

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

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

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

Fie $\sigma \in \Sigma_{s_1 \dots s_n, s}$. Definim:

- 1) $C_\sigma(b_1, \dots, b_n) = B_\sigma(b_1, \dots, b_n)$ dacă $b_i \in B_{s_i}$ pentru orice $i \in [n]$;
- 2) $C_\sigma(r_{s_1}(b_1), \dots, r_{s_n}(b_n)) = r_s(B_\sigma(b_1, \dots, b_n))$ dacă $b_i \in B_{s_i}$ pentru orice $i \in [n]$;
- 3) în celelalte situații operația C_σ poate fi definită oricum.

Să demonstrăm că definiția de mai sus nu este contradictorie.

Fie $b_1 \in B_{s_1}, \dots, b_n \in B_{s_n}$ astfel încât $r_{s_i}(b_i) = b_i \in B_{s_i}$ pentru orice $i \in [n]$. Atunci $b_{s_i} \in h_{s_i}(A_{s_i})$ de unde rezultă că $B_\sigma(b_1, \dots, b_n) \in h_s(A_s)$ și deci $B_\sigma(b_1, \dots, b_n) = r_s(B_\sigma(b_1, \dots, b_n))$.

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

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

Această propoziție ne asigură că algebrele proiective corespund conceptului categorial cu același nume.

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

Demonstrație: Într-o algebră definim relația

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

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

Fie \mathcal{P} o subalgebră a unei algebre libere \mathcal{L} . Relația \vdash este bine fondată și în subalgebra \mathcal{P} .

Fie X mulțimea tuturor elementelor din \mathcal{P} care nu sunt rezultatul aplicării vreunei operații din \mathcal{P} unor elemente din \mathcal{P} . Prin urmare pentru orice $s \in S$

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

Vom dovedi ca \overline{X} , subalgebra generată de X , 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 puțin al unei operații. În plus cel puțin unul dintre argumentele acestei operații nu este din \overline{X} deoarece în caz contrar obținem contradicția $p_0 \in \overline{X}$. Prin urmare există $p_1 \in \mathcal{P} - \overline{X}$ astfel încât $p_0 \vdash p_1$. Continuând raționamentul prin inducție ar rezulta că relația \vdash nu este noetheriană în \mathcal{P} , o contradicție.

Observăm că \mathcal{P} este algebră Peano peste X . Chiar din definiția lui X rezultă că rezultatul aplicării unei operații din \mathcal{P} unor elemente din \mathcal{P} nu este în X . Ultima condiție din definiția 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 . \square

Propoziție 1.7.5 (Andrei Popescu) *In Alg_Σ orice algebră proiectivă este liberă.*

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

Deoarece \mathcal{P} este o algebră proiectivă există un morfism $g : \mathcal{P} \longrightarrow T_\Sigma(P)$ cu proprietatea $g; \epsilon = 1_P$. Observăm că g este o injecție. 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 Ecuatii

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

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

Ce sunt acestea? Sunt egalități de două expresii cuantificate universal prin mulțimea variabilelor conținute în cele două expresii. Deci o astfel de axiomă are forma

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

unde l și r sunt elemente din algebra liber generată de mulțimea X de variabile. Semnul $\overset{\circ}{=}$ este utilizat în loc de $=$, între două expresii, pentru a marca faptul că egalitatea poate fi și falsă, fapt pentru care vom folosi denumirea de egalitate formală în loc de egalitate. În cazul multisortat, cei doi membri ai unei egalități formale trebuie să fie de același sort. Uneori scriem $l \overset{\circ}{=}_s r$ 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 și r în algebra \mathcal{D} să fie același indiferent de valorile date în \mathcal{D} variabilelor din X . Ținând 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:

Definiția 1.8.1 Axioma $(\forall X)l \stackrel{\circ}{=} r$ este satisfăcută în algebra \mathcal{D} dacă și 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 notația

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

și 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 **ecuație** în locul celui de axiomă, pentru a ne conforma cu terminologia internațională.

În plus vor intra în joc și așa zisele **ecuații condiționate**. 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 Ecuații condiționate

În logica ecuațională o axiomă poate fi o implicație

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

unde ipoteza este o conjuncție de egalități formale și concluzia o egalitate formală. Toată implicația este cuantificată universal, fapt care nu apare scris mai sus. În acest cadru o axiomă, numită în continuare și ecuație condiționată a logicii ecuaționale, poate fi scrisă sub forma

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

unde a și c sunt elemente de același sort s , iar H este o mulțime **finită** de egalități formale din algebra liber generată de mulțimea X de variabile. Ipoteza implicației este dată de mulțimea 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 funcție 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 evaluării conduce la egalități adevărate, caz în care trebuie ca $h_s(a) = h_s(c)$. Deci Σ -algebra \mathcal{D} satisface ecuația condiționată $(\forall X)a \stackrel{\circ}{=}_s c \text{ if } H$ fapt notat prin

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

dacă și 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 menționăm din nou diferența esențială între semnele $\stackrel{\circ}{=}_s$ și $=$, diferență care va fi menținută constant pe parcursul întregului text. Egalul peste care s-a pus un cerculeț ($\stackrel{\circ}{=}_s$) indică o egalitate formală care poate fi adevărată sau falsă. Egalul $=$ are semnificația uzuală indicând de obicei o egalitate adevărată.

1.8.3 Necesitatea utilizării cuantificatorilor în ecuații

Vom ilustra printr-un exemplu necesitatea utilizării cuantificatorilor în ecuațiile logicii ecuaționale multi-sortate.

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

$$a \xrightarrow{g} bool \xrightleftharpoons[T]{F}$$

unde se vede că sortul rezultat al celor trei simboluri de operație g, F și T este $bool$; operațiile F și T nu au argumente și g are un singur argument de sort a .

Vom lucra cu două Σ -algebre.

Σ -algebra inițială este $\mathcal{I} = (\emptyset, \{F, T\}; I_g, I_F, I_T)$ unde $F \neq T$; $I_F = F$, $I_T = T$ și $I_g : \emptyset \longrightarrow \{F, T\}$ este funcția 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 și $\{g(x), F, T\}$ pentru sortul $bool$.

Are loc relația $\mathcal{I} \models_\Sigma F \stackrel{\circ}{=} T$? 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 puțin două variante.

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

Am arătat că $\mathcal{I} \models_\Sigma (\forall \emptyset) T \stackrel{\circ}{=} F$ este falsă și că $\mathcal{I} \models_\Sigma (\forall x) T \stackrel{\circ}{=} F$ este adevărată. Dacă ometem cuantificatorii obținem: “ $\mathcal{I} \models_\Sigma T \stackrel{\circ}{=} F$ este falsă și $\mathcal{I} \models_\Sigma T \stackrel{\circ}{=} F$ este adevărată.”

Contradicția obținută prin omiterea cuantificatorilor din fața egalității formale $F \stackrel{\circ}{=} T$ arată că în logica ecuațională multisortată prezența cuantificatorilor în ecuații este necesară.

În multe cazuri, de exemplu în limbajele de programare PROLOG, OBJ, Maude, CafeOBJ, etc cuantificatorii din fața ecuațiilor sunt omiși dar se presupune implicit că ecuațiile sunt cuantificate prin mulțimea variabilelor care apar în ecuație. Egalitatea formală $T \stackrel{\circ}{=} F$ este implicit interpretată $(\forall \emptyset) T \stackrel{\circ}{=} F$.

1.8.4 În primul rând semantica

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

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

mulțimea *propozițiilor* sale. Propozițiile sunt de fapt egalități formale care pot fi adevărate sau false.

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

Având în vedere cele de mai sus nu vom mai face nici o distincție între relațiile lui \mathcal{D} și mulțimile de egalități formale între elementele lui \mathcal{D} .

Definiția 1.8.2 O ecuație condiționată este

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

unde X este o mulțime S -sortată de variabile, l și r sunt două elemente de sort s din $T_\Sigma(X)$ iar H o mulțime **finită** de egalități formale din $T_\Sigma(X)$. \square

O ecuație condiționată în care $H = \emptyset$ devine necondiționată și este numită pe scurt *ecuație*. În acest caz scriem doar $(\forall X) l \stackrel{\circ}{=}_s r$ în loc de $(\forall X) l \stackrel{\circ}{=}_s r \text{ if } \emptyset$.

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

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

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

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

În continuare fixăm o mulțime Γ de ecuații condiționate, numite axiome.

Definiția 1.8.4 Spunem că algebra \mathcal{D} *satisface* Γ sau că \mathcal{D} este o Γ -algebră și scriem $\mathcal{D} \models \Gamma$ dacă \mathcal{D} *satisface* toate ecuațiile condiționate din Γ .

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

\square

1.8.5 Punctul de vedere local

Fixăm o algebră \mathcal{A} și lucrăm cu propoziții din $Sen(\mathcal{A})$. Acesta este punctul *local* de vedere al logicii ecuaționale. Cazul în care ne interesează propoziții din algebre diferite este numit *global* dar va fi puțin folosit în această carte. În cazul global o propoziție din \mathcal{D} va fi scrisă $(\forall \mathcal{D})a \stackrel{\circ}{=}_s c$ în loc de $a \stackrel{\circ}{=}_s c$. Notăția fără cuantificatori folosită în cazul local nu contrazice ceea ce am scris despre necesitatea utilizării cuantificatorilor în ecuații. Pur și simplu nu scriem cuantificatorul pentru că fixând algebra \mathcal{A} el va fi mereu același $(\forall \mathcal{A})$ și deci îl știm 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 și la rescrierile modulo ecuații, rămân valabile în acest cadru mai general. Aceasta prezentare unitară a diverselor tipuri de rescriere este de fapt contribuția autorului la modernizarea lecțiilor despre rescriere. Există în lume trei cărți privind rescrierile [2, 17, 18]. Nici una nu prezintă stilul local.

Menționăm 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 . Menționăm și 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 și y iar asociativitatea $(xy)z = x(yz)$ într-o algebră liberă cu trei generatori x, y și z .

1.8.6 Congruența semantică

Privind Γ ca o mulțime de axiome, ne interesează toate consecințele ei semantice. Prin consecință semantică se înțelege orice fapt adevărat în orice Γ -algebră. Logica ecuațională este interesată de consecințele semantice care sunt scrise ca egalități formale.

Prin urmare ecuația $(\forall X)l \stackrel{\circ}{=}_s r$ este o consecință semantică a lui Γ , denumită și tautologie sau propoziție valabilă a logicii ecuaționale, dacă ea este adevărată în orice Γ -algebră, adică $h_s(l) = h_s(r)$ pentru orice Γ -algebră \mathcal{M} și pentru orice morfism $h : T_{\Sigma}(X) \longrightarrow \mathcal{M}$.

Pentru localizare fixăm algebra $T_{\Sigma}(X)$ și pentru generalizare o înlocuim cu o Σ -algebră arbitrară \mathcal{A}

Prin urmare vom lucra într-o Σ -algebră \mathcal{A} și vom grupa într-o relație toate tautologiile(propozițiile valide) logicii ecuaționale din algebra \mathcal{A} .

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

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

Dacă nu există pericol de confuzie vom prefera să scriem \equiv_{Γ} 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 relație de congruență și deoarece orice intersecție de relații de congruențe este o relație de congruență, deducem că \equiv_{Γ} este o relație de congruență.

\equiv_{Γ} este numită **congruență semantică**.

Definim **regula de deducție a substituției** utilizată atât în logica ecuațională cât și în rescrierea termenilor.

Sub $_{\Gamma}$ Pentru orice $(\forall X)l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$ și 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 $h_s(l) \overset{\circ}{=}_s h_s(r)$.

Definiția 1.8.5 O submulțime M a lui $Sen(\mathcal{A})$ se numește **închisă la substituție** sau închisă la **Sub $_{\Gamma}$** dacă pentru orice $(\forall X)l \overset{\circ}{=}_s r$ **if** H în Γ și 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 $h_s(l) \overset{\circ}{=}_s h_s(r) \in M$.

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

Demonstrație: Fie $(\forall X)l \overset{\circ}{=}_s r$ **if** H în Γ și $h : T_{\Sigma}(X) \longrightarrow \mathcal{A}$ 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 $h_s(l) Ker(f) h_s(r)$. \square

Propoziție 1.8.7 Congruența semantică este închisă la substituție.

Demonstrație: Congruența semantică este o intersecție de congruențe închise la substituție. Deoarece o intersecție de congruențe închise la substituții este o congruență închisă la substituții rezultă că \equiv_{Γ} este o congruență închisă la substituție. \square

Propoziție 1.8.8 Dacă \sim este o congruență închisă la substituții, atunci $\mathcal{A}/\sim \models \Gamma$.

Demonstrație: 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 și $h : T_{\Sigma}(X) \longrightarrow \mathcal{A}/\sim$ un morfism astfel încât $h_t(u) = h_t(v)$ pentru orice $u \overset{\circ}{=} v \in H$. 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 $u \overset{\circ}{=} v \in H$ deoarece $\rho_t(f_t(u)) = \rho_t(f_t(v))$ deducem $f_t(u) \sim f_t(v)$.

Deoarece \sim este o congruență închisă la substituții obținem $f_s(l) \sim f_s(r)$. Prin urmare $\rho_s(f_s(l)) = \rho_s(f_s(r))$, de unde $h_s(l) = h_s(r)$. \square

Arătați că \equiv_{Γ} este cea mai mică congruență închisă la substituție.

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

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

Demonstrație: Se aplică propozițiile 1.8.7 și 1.8.8. \square

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

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

Corolar 1.8.11 *Dacă \mathcal{A} este Σ -algebră inițială, atunci \mathcal{A}_Γ este Γ -algebră inițială.*

Corolar 1.8.12 *Pentru orice semnătură Σ și pentru orice mulțime Γ de ecuații condiționate există o Γ -algebră inițială.*

Demonstrație: Se folosește existența Σ -algebrei initiale și corolarul precedent.

1.9 TIPURI ABSTRACTE de DATE

Am văzut în lecțiile precedente că orice semnătură determină prin algebra sa inițială un tip abstract de date. Tipul abstract de date al numerelor naturale a fost determinat de semnătura formată din constanta 0 și operația unară cunoscută sub numele de succesor.

Acum am pus în evidență un instrument mai puternic deoarece orice semnătură împreună cu o mulțime Γ de ecuații condiționate determină prin Γ -algebra inițială, a cărei existență am dovedit-o mai sus, un tip abstract de date.

1.9.1 Tipul abstract al numerelor naturale - continuare

Considerăm semnătura cu un singur sort *nat*, o singură constantă de sort *nat* și o singură operație unară cu argument și rezultat de sort *nat*:

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

Elementele algebrei inițiale sunt

$$0, s(0), s(s(0)), s(s(s(0))), s(s(s(s(0)))), \dots$$

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

Propoziție 1.9.1 *Algebra $(N, 0_N, s_N)$ definită prin: N este mulțimea numerelor naturale, 0_N este numărul natural zero și $s_N(n) = n + 1$ pentru orice număr natural n ; este inițială. (vezi și propoziția 1.2.7)*

Propoziția anterioară ne arată cum pot fi definite numerele naturale prin metoda algebrei inițiale ca tip abstract de date.

Deocamdată prin semnătura de mai sus calculatorul învață numerele naturale dar nu știe încă să calculeze. Să-l învățăm să adune și să înmulțească. 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 .
```

Trebuie să menționăm că egalitățile de mai sus sunt folosite în două moduri:

a) ca ecuații care împreună cu semnătura de patru operații formează o mulțime Γ pentru a defini o structură algebrică și

b) ca reguli de rescriere în timpul execuției programelor.

Propoziție 1.9.2 Algebra $\mathcal{N} = (N, 0_N, s_N, +_N, *_N)$ este Γ -algebră inițială.

Demonstrație: Fie $\mathcal{A} = (A, 0_A, s_A, +_A, *_A)$ o Γ -algebră. Menționăm că algebra \mathcal{A} satisfac ecuațiile 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 propoziția de mai sus.

Pentru a demonstra existența nu avem decât o singură șansă și anume să dovedim că unicul morfism $h : (N, 0_N, s_N) \longrightarrow (A, 0_A, s_A)$ este și morfism de Γ -algebre. Reamintim că

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

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

$$\begin{aligned} h(n +_N 0_N) &= h(n) = h(n) +_A 0_A = h(n) +_A h(0_N) \text{ și} \\ h(n +_N (m+1)) &= h(s_N(n +_N m)) = s_A(h(n +_N m)) = s_A(h(n) +_A h(m)) = h(n) +_A s_A(h(m)) = \\ &= h(n) +_A h(m+1). \end{aligned}$$

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

$$\begin{aligned} h(n *_N 0_N) &= h(0_N) = 0_A = h(n) *_A 0_A = h(n) *_A h(0_N) \text{ și} \\ h(n *_N (m+1)) &= h(n *_N m +_N n) = h(n *_N m) +_A h(n) = (h(n) *_A h(m)) +_A h(n) = h(n) *_A s_A(h(m)) = \\ &= h(n) *_A h(m+1). \quad \square \end{aligned}$$

Propoziția anterioară demonstrează corectitudinea definiției de mai sus. Deoarece algebra \mathcal{N} este inițială rezultă că ea este izomorfă cu Γ -algebra inițială, deci specificația de mai sus caracterizează, prin Γ -algebra sa inițială, tipul de date al numerelor naturale.

Cred că este util să menționăm 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 știm din algebră că numerele naturale formează un semiinel inițial. Caracterizarea numerelor naturale ca semiinel inițial nu ne mulțumește deoarece axiomele conceptului de semiinel nu ne dau un program (calculatorul este incapabil să facă operații cu numerele naturale folosind numai axiomele semiinelului). Axiomele lui Peano dau un program în programarea prin rescriere.

1.10 Problema programării prin rescriere

Principala problemă este : "poate o mașină să demonstreze că $a \equiv_{\Gamma} c$?"

Se știe că în unele cazuri rescrierile ne dau o soluție.

Chapter 2

RESCRIERI

2.1 TEORII DEDUCTIVE À la MOISIL

Fie E o mulțime de propoziții. O *regulă de deducție* pe E este o pereche (H, e) unde H este o submulțime finită a lui E (și se numește ipoteza regulii) și $e \in E$ (și se numește concluzia regulii). Poate mai sugestiv o regulă (H, e) ar fi trebuit scrisă $H \longrightarrow e$ deoarece semnificația ei este “propoziția e este o consecință a propozițiilor din H ”. Mai menționăm că în cazul particular $H = \emptyset$ regula (\emptyset, e) spune că e este o axiomă.

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

Fie R o mulțime de reguli de deducție. O submulțime D a lui E se numește închisă la R dacă și numai dacă D este închisă la orice regulă din R .

Definiția închiderii la substituție, dată în secțiunea 1.8.6, este un caz particular al definiției de mai sus.

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

Demonstrație: Fie $(H, e) \in R$ astfel încât $H \subseteq \bigcap_{i \in I} D_i$. Pentru orice $i \in I$, deoarece D_i este închisă la R și $H \subseteq D_i$ deducem că $e \in D_i$. Deci $e \in \bigcap_{i \in I} D_i$. \square

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

O propoziție $e \in E$ poate fi demonstrată folosind R dacă și numai dacă există o secvență finită de propoziții $e_1 e_2 \dots e_n$ astfel încât $e_n = e$ și pentru orice $i \in [n]$, există $(H, e_i) \in R$ astfel încât $H \subseteq \{e_1, e_2, \dots, e_{i-1}\}$. Vom spune că $e_1 e_2 \dots e_n$ este o *demonstrație* pentru $e_n = e$.

Observația 2.1.2 Dacă α și β sunt demonstrații atunci $\alpha\beta$ este demonstrație.

Notăm cu **Teor**(R) mulțimea propozițiilor care pot fi demonstrate folosind R .

Lemă 2.1.3 Fie $D \subseteq E$. Dacă D este închisă la R atunci **Teor**(R) $\subseteq D$.

Demonstrație: Fie $e \in \mathbf{Teor}(R)$. Prin urmare există o secvență finită de propoziții $e_1 e_2 \dots e_n$ astfel încât $e_n = e$ și pentru oricare i din $[n]$ există $(H, e_i) \in R$ astfel încât $H \subseteq \{e_1, e_2, \dots, e_{i-1}\}$. Probăm prin inducție că $e_i \in D$ pentru orice i din $[n]$. Ipoteza de inducție este $(\forall k < i) e_k \in D$. Deoarece șirul $e_1 e_2 \dots e_n$ este o demonstrație există $(H, e_i) \in R$ cu $H \subseteq \{e_1, e_2, \dots, e_{i-1}\}$. Din ipoteza de inducție 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 **Teor**(R), deci **Teor**(R) $\subseteq D$. \square

Această leamnă ne furnizează o metodă prin care putem arăta că toate propozițiile demonstrabile folosind R au o anumită proprietate și anume este suficient să arătăm că mulțimea propozițiilor cu proprietatea dată este închisă la R .

Teorema 2.1.4 $\mathbf{Teor}(R)$ este cea mai mică mulțime (în raport cu \subseteq) care este închisă la R .

Demonstrație: Demonstrăm că $\mathbf{Teor}(R)$ este închisă la R . În acest scop presupunem $(H, e) \in R$, unde $H = \{f_1, \dots, f_n\}$ și $H \subseteq \mathbf{Teor}(R)$. Pentru orice $i \in [n]$ din $f_i \in \mathbf{Teor}(R)$ deducem existența unei demonstrații $\alpha_i f_i$ unde α_i este un șir de propoziții din E . Atunci $\alpha_1 f_1 \dots \alpha_n f_n$ este o demonstrație deoarece este o concatenare de demonstrații și pentru că $(\{f_1, \dots, f_n\}, e) \in R$ deducem că $\alpha_1 f_1 \dots \alpha_n f_n e$ este o demonstrație pentru e . Deci $e \in \mathbf{Teor}(R)$. Rezultă că $\mathbf{Teor}(R)$ este închisă la R . Din lema 2.1.3 rezultă că $\mathbf{Teor}(R)$ este cea mai mică în sensul incluziunii care este închisă la R . \square

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

2.2 LOGICĂ ECUAȚIONALĂ

Fie Γ o mulțime de ecuații condiționale și o Σ -algebra \mathcal{A} fixată.

Mulțimea propozițiilor adevărate, tautologiile, din \mathcal{A} este chiar congruența semantică

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

Conform tradiției mai scriem

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

Se caută o mulțime corectă și completă de reguli de deducție pentru $\equiv_\Gamma^{\mathcal{A}}$.

2.2.1 Reguli de deducție, corectitudine

Notăm cu $\mathbf{R_E}$ mulțimea următoarelor reguli de deducție pentru logica ecuațională multisortată:

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

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

Observația 2.2.1 Observăm că o mulțime de egalități formale este închisă la regula **R** dacă și numai dacă este o relație reflexivă.

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

Observația 2.2.2 Observăm că o mulțime de egalități formale este închisă la regula **S** dacă și numai dacă este o relație simetrică.

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

Observația 2.2.3 Observăm că o mulțime de egalități formale este închisă la regula **T** dacă și numai dacă este o relație tranzitivă.

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

Observația 2.2.4 *Observăm că o mulțime de egalități formale este închisă la regula $\mathbf{C}\Sigma$ dacă și numai dacă este o relație compatibilă cu toate operațiile algebrei \mathcal{A} .*

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

Teorema 2.2.5 *Regulile de deducție $\mathbf{R_E}$ sunt corecte pentru \equiv_Γ .*

Demonstrație: Deoarece \equiv_Γ este congruență rezultă ea este închisă la primele patru reguli, prin urmare ele sunt corecte pentru \equiv_Γ . Corectitudinea ultimei reguli rezultă din închiderea congruenței semantice la substituții în conformitate cu propoziția 1.8.7. \square

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

Demonstrație: Congruența semantică, adică mulțimea egalităților formale cu proprietatea $\models a \overset{\circ}{=}_s b$, este închisă la $\mathbf{R_E}$ conform teoremei 2.2.5

Mulțimea egalităților formale demonstrabile, adică mulțimea egalităților formale cu proprietatea $\vdash a \overset{\circ}{=}_s b$, este cea mai mică mulțime închisă la $\mathbf{R_E}$ conform teoremei 2.1.4.

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

2.2.2 Completitudine

Pentru orice $s \in S$ și $a, b \in A_s$ se definește relația \sim_Γ în \mathcal{A} prin:

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

Deoarece regulile \mathbf{R} , \mathbf{S} și \mathbf{T} sunt în $\mathbf{R_E}$ deducem că \sim_Γ este închisă la \mathbf{R} , \mathbf{S} și \mathbf{T} deci este o echivalență. Mai mult, deoarece $\mathbf{C}\Sigma$ este în $\mathbf{R_E}$ rezultă că \sim_Γ este închisă la $\mathbf{C}\Sigma$ deci este congruență.

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

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

Demonstrație: Deoarece \mathbf{Sub}_Γ este în $\mathbf{R_E}$ rezultă că \sim_Γ este închisă la substituții prin urmare aplicând propoziția 1.8.8 rezultă concluzia.

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

Demonstrație: Fie $\models a \overset{\circ}{=}_s b$. Din definiția tautologiilor deducem $(\forall h : A \longrightarrow B \models \Gamma) h_s(a) = h_s(b)$. Întrucât $\mathcal{A}_\Gamma \models \Gamma$ și $\eta_{\mathcal{A}} : \mathcal{A} \longrightarrow \mathcal{A}_\Gamma$ 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 congruența \sim_Γ și congruența semantică \equiv_Γ coincid. Prin urmare algebra \mathcal{A}_Γ și morfismul $\eta_{\mathcal{A}}$ coincid cu cele introduse într-un mod numai aparent diferit în secțiunea 1.8, teoremele 1.8.9 și 1.8.10.

Proprietatea de universalitate a algebrei \mathcal{A}_Γ demonstrată atunci, teorema 1.8.10, rămâne adevărată și în noul context.

Următorul pas după o teoremă de completitudine care ne asigură existența unei demonstrații pentru orice tautologie este de a găsi aceste demonstrații. Informaticienii sunt ceva mai pretențioși deoarece doresc ca aceste demonstrații 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 școală. De exemplu în șirul de egalități

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

facem două rescrieri(înlocuiri): $3+4$ esre rescris în 7 și apoi $2 * 7$ este rescris în 14. Aceste rescrieri sunt permise de regulile adunării și înmulțirii. Partea expresiei care rămâne neschimbată deoarece rescrierea are loc în interiorul ei se numește 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 vârful contextului 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 $3+4$ și nu $2+5$? Prin urmare spre deosebire de egalitate care este simetrică, **rescrierea nu este simetrică**.

Deoarece se dorește 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 și așa mai departe.

Eliminarea simetriei dintre regulile de deducție este principala diferență dintre calculul cu egalități reflectat de logica ecuațională. Deoarece eliminarea simetriei duce la pierderea completitudinii, simetria va trebui înlocuită cu altceva pentru a reobține completitudinea.

2.3.1 Preliminarii

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

Fie X o mulțime S -sortată de variabile și $T_\Sigma(X)$ Σ -algebra liber generată de X . Pentru orice $x \in X$ și $\alpha \in T_\Sigma(X)$, vom nota cu

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

Observăm că $nr_x(x) = 1$, $nr_x(y) = 0$ pentru orice $y \in X - \{x\}$ și $nr_x(\sigma(e_1, e_2, \dots, e_k)) = nr_x(e_1) + nr_x(e_2) + \dots + nr_x(e_k)$.

Fie \mathcal{A} o Σ -algebră, \bullet o variabilă de sort s , $\bullet \notin A_s$. Considerăm algebra liber generată de $A \cup \{\bullet\}$, și anume $T_\Sigma(A \cup \{\bullet\})$. Vom presupune fără a micșora generalitatea că

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

Un element c din $T_\Sigma(A \cup \{\bullet\})$ se numește *context* dacă $nr_\bullet(c) = 1$. Dacă $c = \sigma(c_1, c_2, \dots, c_n)$ este un context, atunci există un $1 \leq i \leq n$ astfel încât c_i este context și $c_j \in T_\Sigma(A)$ pentru orice $j \neq i$.

Pentru $d \in A_s$, vom nota cu $\bullet \leftarrow d : T_\Sigma(A \cup \{\bullet\}) \rightarrow \mathcal{A}$ unicul morfism de Σ -algebre cu proprietatea $(\bullet \leftarrow d)(\bullet) = d$ și $(\bullet \leftarrow d)(a) = a$ pentru orice $a \in A$. Menționăm că restricția 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 $T_\Sigma(A \cup \{\bullet\})$ și $a \in A_s$, 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 și al evaluării în \mathcal{A} al expresiei obținute 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 inducție structurală.

Compunerea relațiilor $R \subseteq A \times B$ și $Q \subseteq B \times C$ este relația $R; Q \subseteq A \times C$ definită prin

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

Compunerea relațiilor este asociativă.

Relațiile între elementele aceleiași mulțimi A , adică părțile lui $A \times A$, formează cu operația de compunere un monoid al cărui element neutru este relația de egalitate. Pentru astfel de relații sunt definite puterile lor naturale. Prin definiție R^0 este chiar relația de egalitate. Prin definiție

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

este închiderea reflexivă și tranzitivă a relației R . Pentru o relație notată \longrightarrow în loc de notația \longrightarrow^* se preferă notația \longrightarrow^* . Menționăm că $*$ este operator de închidere în mulțimea părților lui $A \times A$.

2.3.2 Închiderea la contexte

Definiția 2.3.1 O relație Q pe A se numește **închisă la contexte** dacă pentru orice context c și pentru orice pereche de elemente a, d din A_s , $a Q d$ implică $c[a] Q c[d]$.

Observăm că relația de egalitate este închisă la contexte.

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

Demonstrație: Presupunem că relațiile Q și R sunt închise la contexte. Fie c un context.

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

Propoziție 2.3.3 *Dacă relația R este închisă la contexte, atunci închiderea ei reflexivă și tranzitivă R^* este închisă la contexte.*

Demonstrație: Presupunem că R este închisă la contexte. Folosind observația și lema de mai sus deducem că pentru orice n natural relația R^n este închisă la contexte.

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

Introducem regula de deducție

$$\begin{aligned} \text{CA}\Sigma \quad a \stackrel{\circ}{=}_{s_i} d \quad \text{implică} \quad A_\sigma(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n) &\stackrel{\circ}{=}_s A_\sigma(a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n) \\ (\forall \sigma \in \Sigma_{s_1 \dots s_n, s}), \text{ orice } 1 \leq i \leq n, \text{ unde } a_j \in A_{s_j} \text{ pentru orice } j \in \{1, \dots, i-1, i+1, \dots, n\} \\ \text{și } a, d \in A_{s_i}. \end{aligned}$$

Definiția 2.3.4 O relație $\rho \subset A \times A$ se numește **compatibilă pe argumente cu operațiile** algebrei \mathcal{A} dacă este închisă la $\text{CA}\Sigma$.

Propoziție 2.3.5 *O relație este închisă la contexte dacă și numai dacă este compatibilă pe argumente cu operațiile algebrei.*

Demonstrație: Presupunem Q închisă la contexte. Pentru a demonstra compatibilitatea pe argumente cu operația σ aplicăm ipoteza pentru contextul $c = \sigma(a_1, \dots, a_{i-1}, \bullet, a_{i+1}, \dots, a_n)$. Presupunem $(a, d) \in Q$ și deducem din închiderea la contexte că $(c[a], c[d]) \in Q$. Observăm că

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

Analog $c[d] = A_\sigma(a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n)$, prin urmare

$A_\sigma(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n) \stackrel{\circ}{=} A_\sigma(a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n)$ este în Q .

Reciproca se arată prin inducție 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 $c = \sigma(a_1, \dots, a_{i-1}, c', a_{i+1}, \dots, a_n)$ unde $c' \in T_\Sigma(A \cup \{\bullet\})$ este un context și $a_i \in T_\Sigma(A)$

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

La fel $c[d] = A_\sigma(a_1[d], \dots, a_{i-1}[d], c'[d], a_{i+1}[d], \dots, a_n[d])$. Mai observăm că $a_j[a] = a_j[d]$ pentru orice $j \neq i$.

Din ipoteza de inducție $c'[a] Q c'[d]$ și ținând cont de compatibilitatea pe argumente obținem $c[a] Q c[d]$.

Definiția 2.3.6 Dacă Q este o relație pe A vom nota

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

Propoziție 2.3.7 \longrightarrow_Q este închiderea la contexte a lui Q .

Demonstrație: Trebuie să dovedim că \longrightarrow_Q este cea mai mică relație închisă la contexte care include Q .

Pentru a dovedi că \longrightarrow_Q este închisă la contexte, vom prefera să arătăm că este compatibilă pe argumente cu operațiile. Fie $(c[a], c[d])$ în \longrightarrow_Q unde $(a, d) \in Q$, $\sigma \in \Sigma_{t_1 t_2 \dots t_n, t}$ un simbol de operație și $a_i \in A_{t_i}$ niște elemente din \mathcal{A} . Folosind contextul $c' = \sigma(a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_n)$ deducem că $(c'[a], c'[d])$ este în \longrightarrow_Q . Dar calculând ca mai sus

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

prin urmare

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

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

Presupunem $Q \subseteq R$ și 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

Propoziția precedentă ne asigură că \longrightarrow_Q este operator de închidere.

2.3.3 Închiderea la preordini compatibile cu operațiile

Preordine semnifică o relație reflexivă și tranzitivă.

Reamintim regulile de deducție în mulțimea egalităților formale între elementele algebrei \mathcal{A} denumite **Reflexivitate**, **Tranzitivitate** și **Compatibilitate** cu operațiile din Σ :

$$\begin{array}{ll} \mathbf{R} & a \stackrel{\circ}{=}_s a \\ \mathbf{T} & a \stackrel{\circ}{=}_s d \text{ și } d \stackrel{\circ}{=}_s c \text{ implică } a \stackrel{\circ}{=}_s c \\ \mathbf{C}\Sigma & a_i \stackrel{\circ}{=}_{s_i} c_i \text{ pentru orice } i \in [n] \text{ implică} \\ & A_\sigma(a_1, a_2, \dots, a_n) \stackrel{\circ}{=} A_\sigma(c_1, c_2, \dots, c_n) \text{ pentru orice } \sigma \in \Sigma_{s_1 s_2 \dots s_n, s} \end{array}$$

Remarcăm că **C** Σ și **R** implică **CA** Σ .

Reciproca **CA** Σ , **R** și **T** implică **C** Σ este demonstrată de lema următoare.

Lemă 2.3.8 Fie $\rho \subset A \times A$ o relație tranzitivă și reflexivă în algebra \mathcal{A} . Dacă ρ este compatibilă pe argumente cu operațiile algebrei \mathcal{A} , atunci ρ e compatibilă cu operațiile, adică închisă la **C** Σ .

Demonstrație: Fie $\sigma \in \Sigma_{s_1 \dots s_n, s}$ și $a_i, b_i \in A_{s_i}$ astfel încât $a_i \rho_{s_i} b_i$ pentru orice $i \in [n]$.

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

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 obținem

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

Din tranzitivitatea relației ρ deducem $A_\sigma(a_1, a_2, \dots, a_n) \rho_s A_\sigma(b_1, b_2, \dots, b_n)$. \square

Pentru a rezuma desenăm schema

$$\begin{array}{ccc} & \mathbf{R} & \\ \mathbf{C}\Sigma & \xleftrightarrow{\quad} & \mathbf{CA}\Sigma \Leftrightarrow \text{î închiderea la contexte} \\ & \mathbf{R, T} & \end{array}$$

Un comentariu privind regulile de deducție **C** Σ și **CA** Σ este bine venit. Regula **C** Σ este utilizată în logica ecuațională. 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 prezența reflexibilității și a tranzitivității regulile **CA** Σ și **C** Σ sunt echivalente. Deoarece reflexivitatea și tranzitivitatea sunt acceptate atât de logica ecuațională cât și de rescrieri, cele două reguli au aceeași putere.

Propoziție 2.3.9 $\xrightarrow{*}_Q$ este cea mai mică preordine compatibilă cu operațiile care include Q .

Demonstrație: Relația $\xrightarrow{*}_Q$ este prin definiție închiderea reflexivă și tranzitivă a relației \rightarrow_Q . Ea este, prin definiție, reflexivă și tranzitivă. Este și închisă la contexte, conform propoziției 2.3.3 deoarece \rightarrow_Q este închisă la contexte.

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

Fie R o preordine compatibilă cu operațiile care include Q . Deoarece **C** Σ și **R** implică **CA** Σ relația R este compatibilă pe argumente cu operațiile, așadar relația R este închisă la contexte, prin urmare aplicând propoziția 2.3.7, deducem $\rightarrow_Q \subseteq R$. Rezultă că $\xrightarrow{*}_Q \subseteq R$ deoarece R este reflexivă și tranzitivă. \square

Din proprietățile operatorilor de închidere și $R \subseteq Q$ deducem $\rightarrow_R \subseteq \rightarrow_Q$ și $\xrightarrow{*}_R \subseteq \xrightarrow{*}_Q$.

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

2.3.4 Γ -rescriere

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

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

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

Remarcați că **SRew** $_\Gamma$, *rescrierea într-un subtermen*, este o regulă de deducție mai puternică decât **Rew** $_\Gamma$, *rescrierea*, care poate fi obținută din **SRew** $_\Gamma$ pentru $c = \bullet$.

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

Comparând substituția și rescrierea observăm următoarele.

- 1) **Rew** $_\Gamma$ și **R** implică **Sub** $_\Gamma$.
- 2) **Sub** $_\Gamma$, **S** și **T** implică **Rew** $_\Gamma$.

Probăm ultima afirmație. Fie $(\forall X) l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$ și un morfism $h : T_\Sigma(X) \rightarrow \mathcal{A}$ cu proprietatea că pentru orice $u \overset{\circ}{=}_{s'} v \in H$ există $d \in A_{s'}$ cu $h_{s'}(u) \overset{\circ}{=}_{s'} d$ și $h_{s'}(v) \overset{\circ}{=}_{s'} d$. Pentru orice $u \overset{\circ}{=}_{s'} v \in H$

cu **S** deducem $d \stackrel{\circ}{=}_s h_{s'}(v)$ și apoi cu **T** obținem $h_{s'}(u) \stackrel{\circ}{=}_{s'} h_{s'}(v)$. În final aplicăm **Sub** $_{\Gamma}$ și deducem $h_s(l) \stackrel{\circ}{=}_s h_s(r)$.

Pentru a rezuma desenăm schema

$$\text{Sub}_{\Gamma} \begin{array}{c} \xrightarrow{\mathbf{R}} \\ \xleftarrow{\mathbf{S}, \mathbf{T}} \end{array} \text{Rew}_{\Gamma} \begin{array}{c} \xleftarrow{\mathbf{CA}\Sigma} \\ \xrightarrow{\mathbf{S}\mathbf{R}\mathbf{ew}_{\Gamma}} \end{array} \text{SRew}_{\Gamma}$$

Definim prin inducție șirul crescător de mulțimi de egalități formale din \mathcal{A} .

$$Q_0 = \emptyset$$

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

Pentru a ne convinge că șirul de mai sus este crescător putem demonstra prin inducție că $(\forall n) Q_n \subseteq Q_{n+1}$. Evident $Q_0 = \emptyset \subseteq Q_1$. Presupunem $Q_n \subseteq Q_{n+1}$ și 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 $H \in \Gamma$ și $h : T_{\Sigma}(Y) \longrightarrow \mathcal{A}$ cu proprietatea $(\forall u \stackrel{\circ}{=}_t v \in H) h_t(u) \Downarrow_{Q_n} h_t(v)$. Deoarece șirul \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 $\longrightarrow_{\bullet}$ și $\overset{*}{\longrightarrow}_{\bullet}$ sunt operatori de închidere deducem că șirurile \longrightarrow_{Q_n} și $\overset{*}{\longrightarrow}_{Q_n}$ sunt crescătoare. Prin definiție Q este reuniunea șirului crescător definit mai sus

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

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

Șirul \Downarrow_{Q_n} este crescător. În plus $a \Downarrow_Q b$ implică existența unui n natural cu proprietatea $a \Downarrow_{Q_n} b$.

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

Propoziție 2.3.10 $\overset{*}{\Longrightarrow}_{\Gamma}$ este închisă la **SRew** $_{\Gamma}$.

Demonstrație: 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 $u \stackrel{\circ}{=}_t v \in H$ și un context $c \in T_{\Sigma}(A \cup \{\bullet\})_{s'}$. Trebuie arătat că $c[h(l)] \overset{*}{\Longrightarrow}_{\Gamma} c[h(r)]$.

Deoarece H este finită și numărul pașilor 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 $c[h_s(l)] \overset{*}{\Longrightarrow}_Q c[h_s(r)]$, adică $c[h_s(l)] \overset{*}{\Longrightarrow}_{\Gamma} c[h_s(r)]$. \square

Propoziție 2.3.11 $\overset{*}{\Longrightarrow}_{\Gamma}$ este cea mai mică relație închisă la **R**, **T**, **C** Σ și **Rew** $_{\Gamma}$.

Demonstrație: Evident $\overset{*}{\Longrightarrow}_{\Gamma}$ este închisă la **R**, **T** și **C** Σ și, fiind închisă la **SRew** $_{\Gamma}$, este închisă și la **Rew** $_{\Gamma}$.

Fie W o relație închisă la **R**, **T**, **C** Σ și **Rew** $_{\Gamma}$. Demonstrăm prin inducție 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 $H \in \Gamma$, $h : T_{\Sigma}(Y) \longrightarrow \mathcal{A}$ și $h_t(u) \Downarrow_{Q_{n-1}} h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$. Din ipoteza de inducție $Q_{n-1} \subseteq W$. Cum W este închisă la **C** Σ și **R** rezultă că W este închisă la **C** $\mathbf{A}\Sigma$, adică la contexte, deci $\longrightarrow_{Q_{n-1}} \subseteq W$. Folosind faptul că W este închisă la **R** și **T** rezultă că $\overset{*}{\longrightarrow}_{Q_{n-1}} \subseteq W$, prin urmare $\Downarrow_{Q_{n-1}} \subseteq \Downarrow_W$.

Din $(\forall u \stackrel{\circ}{=}_t v \in H) h_t(u) \Downarrow_{Q_{n-1}} h_t(v)$ obținem $(\forall u \stackrel{\circ}{=}_t v \in H) h_t(u) \Downarrow_W h_t(v)$. Folosind închiderea lui W la **Rew** $_{\Gamma}$ rezultă că $h_s(l) \stackrel{\circ}{=}_s h_s(r) \in W$, deci $Q_n \subseteq W$.

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

2.3.5 Corectitudinea rescrierii

Teorema 2.3.12 *Toate regulile de deducție de mai sus sunt corecte pentru congruența semantică \equiv_Γ .*

Demonstrație: Din lecția privind logica ecuațională știm că regulile de deducție **R**, **S**, **T**, **CΣ** și **Sub_Γ** sunt corecte.

Deoarece **CΣ** și **R** implică **CAΣ** deducem că **CAΣ** este corectă.

Deoarece **Sub_Γ**, **S** și **T** implică **Rew_Γ** rezultă că **Rew_Γ** este corectă.

Deoarece **Rew_Γ** și **CAΣ** implică **SRew_Γ** deducem că **SRew_Γ** este corectă. \square

Corolar 2.3.13 $\xRightarrow{*} \subseteq \equiv_\Gamma$. (Corectitudinea rescrierii pentru congruența semantică)

Rescrierea nu este completă pentru congruența semantică deoarece **SRew_Γ** nu captează întreaga forță a simetriei.

2.4 Relația de întâlnire, Forme normale

2.4.1 Confluență

Definiția 2.4.1 O relație \succ pe o mulțime A se numește **confluentă** dacă

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

Propoziție 2.4.2 *Dacă \succ este o preordine confluentă pe mulțimea A , atunci relația \downarrow definită prin*

$$a \downarrow b \iff (\exists c \in A) a \succ c \text{ și } b \succ c$$

și denumită “întâlnire prin \succ ” este cea mai mică echivalență pe A care include \succ .

Demonstrație:

- Deoarece din $a \succ a$ și $a \succ a$ deducem $a \downarrow a$. Deci relația \downarrow este reflexivă.
- Simetria este evidentă.
- Probăm tranzitivitatea. Fie $a \downarrow b$ și $b \downarrow c$. Observăm că există $d, e \in A$ astfel încât $a \succ d$, $b \succ d$, $b \succ e$ și $c \succ e$. Din confluență rezultă existența lui $f \in A$ astfel încât $d \succ f$ și $e \succ f$. Rezultă prin tranzitivitate că $a \succ f$ și $c \succ f$, deci $a \downarrow c$.
- Pentru a dovedi că $\succ \subseteq \downarrow$, presupunând că $a \succ b$ și observând că $b \succ b$ deducem $a \downarrow b$.
- Fie \equiv o relație de echivalență pe A care include \succ . Probăm că \equiv include \downarrow . Presupunând că $a \downarrow b$ rezultă existența lui $c \in A$ astfel încât $a \succ c$ și $b \succ c$. Deducem $a \equiv c$ și $b \equiv c$, deci $a \equiv b$. \square

Observația 2.4.3 *Dacă \succ este o preordine confluentă și compatibilă pe o Σ -algebră multisortată, atunci \downarrow este o congruență.*

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

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

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

2.4.2 Completitudinea întâlnirii prin rescriere

Prin definiție, pentru orice $s \in S$ și orice $a, d \in A_s$:

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

Relația \downarrow_{Γ} este denumită întâlnire prin rescriere.

Propoziție 2.4.4 $\downarrow_{\Gamma} \subseteq \equiv_{\Gamma}$.

Demonstrație: Din $a \downarrow_{\Gamma} d$ deducem existența lui c astfel încât: $a \xrightarrow{*}_{\Gamma} c$ și $d \xrightarrow{*}_{\Gamma} c$. Din aceasta utilizând corectitudinea rescrierii față de congruența semantică deducem că $a \equiv_{\Gamma} c$ și $d \equiv_{\Gamma} c$ deci $a \equiv_{\Gamma} d$. \square

Aceasta dovedește corectitudinea lui \downarrow_{Γ} în raport cu congruența semantică. Pentru a demonstra și completitudinea sa, adică incluziunea inversă, avem nevoie să presupunem că $\xrightarrow{*}_{\Gamma}$ este confluentă. Folosind observația 2.4.3 deducem că \downarrow_{Γ} este o congruență.

Lemă 2.4.5 Dacă $\xrightarrow{*}_{\Gamma}$ este confluentă, atunci congruența \downarrow_{Γ} este închisă la substituție.

Demonstrație: Fie $(\forall X) l \overset{\circ}{=}_s r$ if $H \in \Gamma$ și fie $h : T_{\Sigma}(X) \rightarrow \mathcal{A}$ 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 proprietățile $h_t(u) \xrightarrow{*}_{\Gamma} a_{uv}$ și $h_t(v) \xrightarrow{*}_{\Gamma} a_{uv}$. Deoarece $\xrightarrow{*}_{\Gamma}$ este închisă la **Rew $_{\Gamma}$** deducem că $h_s(l) \xrightarrow{*}_{\Gamma} h_s(r)$. Prin urmare $h_s(l) \downarrow_{\Gamma} h_s(r)$. \square

Fie \mathcal{A}^{Γ} factorizarea lui \mathcal{A} prin \downarrow_{Γ} și fie $\tau : \mathcal{A} \rightarrow \mathcal{A}^{\Gamma}$ morfismul de factorizare. Deoarece congruența \downarrow_{Γ} este închisă la substituție deducem din propoziția 1.8.8 că $\mathcal{A}^{\Gamma} \models \Gamma$.

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

Demonstrație: Folosind propoziția 2.4.4 avem de demonstrat doar $\equiv_{\Gamma} \subseteq \downarrow_{\Gamma}$.

Fie $a \equiv_{\Gamma} c$. Deoarece $\mathcal{A}^{\Gamma} \models \Gamma$ și $\tau : \mathcal{A} \rightarrow \mathcal{A}^{\Gamma}$ 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_{Γ} . \square

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

”poate o mașină să demonstreze că $a \downarrow_{\Gamma} c$?”

2.4.3 Forme normale

La mulțimi

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

Definiția 2.4.7 Elementul $n \in A$ se numește **o formă normală** pentru relația \succ pe mulțimea A dacă

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

În limbajul teoriei mulțimilor partial ordonate, o formă normală este numită element minimal. Unele texte folosesc “canonică” în loc de formă “normală”.

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

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

Observația 2.4.8 Dacă $a \succ d$, atunci $fn(a) = fn(d)$.

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

Observația 2.4.9 Axioma **FN!** implică \succ este confluentă.

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

Observația 2.4.10 Funcția $fn : A \longrightarrow N$ este surjectivă și

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

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

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

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

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

Remarcăm că N este un sistem complet și independent de reprezentanți pentru \downarrow . Independent înseamnă că nu există două forme normale echivalente și diferite, adică $n \downarrow n'$ implică $n = n'$, oricare ar fi formele normale n și n' . Complet 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 $a \downarrow n$.

Observația 2.4.11 Există o unică bijecție $b : A/\downarrow \longrightarrow N$ astfel încât $\rho; b = fn$ unde $\rho : A \longrightarrow A/\downarrow$ este funcția naturală de factorizare.

Demonstrație: Din proprietatea 1.6.6 de universalitate a mulțimii cât rezultă existența și unicitatea funcției $b : A/\downarrow \longrightarrow N$ cu proprietatea $\rho; b = fn$.

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

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

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

Observația 2.4.12 $fn : \mathcal{A} \longrightarrow \mathcal{N}$ este un Σ -morfism.

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

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

prin urmare

$$\begin{aligned} fn_s(A_\sigma(a_1, a_2, \dots, a_n)) &= fn_s(A_\sigma(fn_{s_1}(a_1), fn_{s_2}(a_2), \dots, fn_{s_n}(a_n))) = \\ &= N_\sigma(fn_{s_1}(a_1), fn_{s_2}(a_2), \dots, fn_{s_n}(a_n)). \end{aligned} \square$$

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

Demonstrație: 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 existența și unicitatea morfismului $h : \mathcal{A}/\downarrow \longrightarrow \mathcal{N}$ cu proprietatea $\rho; h = fn$. Ca în cazul mulțimilor rezultă că morfismul h este bijectiv. Din propoziția 1.1.8 rezultă că h este izomorfism.

Teorema 2.4.14 *Dacă \Rightarrow_Γ^* satisface axioma **FN!**, atunci algebra formelor normale este izomorfă cu \mathcal{A}^Γ .*

2.4.4 Relații canonice

În practică pentru a verifica că \Rightarrow_Γ^* satisface axioma **FN!**, preferăm uneori să verificăm că satisface proprietățile de confluență și terminare.

Definiția 2.4.15 Spunem că \succ are proprietatea de **terminare** dacă nu există șiruri $\{a_n\}$ de elemente din A astfel încât pentru orice număr natural n să avem $a_n \succ a_{n+1}$ și $a_n \neq a_{n+1}$. \square

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

Observația 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 $a \succ n$.*

Demonstrație: Raționând prin absurd, a nu este în forma normală; prin urmare există $a_1 \in A$ cu proprietățile $a \succ a_1$ și $a \neq a_1$. Raționând în continuare prin absurd, a_1 nu este în forma normală; prin urmare există $a_2 \in A$ cu proprietățile $a_1 \succ a_2$ și $a_1 \neq a_2$. Continuând acest raționament prin inducție putem construi un șir de elemente care contrazice proprietatea de terminare. \square

Dacă preordinea \succ este confluentă și are proprietatea de terminare spunem că $\langle A, \succ \rangle$ este **canonică**. \square

Propoziție 2.4.17 *Dacă $\langle A, \succ \rangle$ este canonică atunci axioma **FN!** este verificată.*

Demonstrație: Unicitatea va rezulta din confluență. Presupunând că n' și n'' sunt două elemente în formă normală cu proprietatea $a \succ n'$ și $a \succ n''$ din confluență există $d \in A$ astfel încât $n' \succ d$ și $n'' \succ d$. Deoarece n' și n'' în sunt formă normală deducem că $n' = d$ și $n'' = d$, deci $n' = n''$. \square

*Din punct de vedere practic terminarea este necesară pentru a opri execuția calculatorului iar confluența este necesară pentru completitudine. Aceste proprietăți implică **FN!**. Pentru a demonstra că $a \equiv_\Gamma c$, adică $a \downarrow_\Gamma c$, este suficient conform observației 2.4.10 să rescriem a și c în forma lor normală $fn(a)$ și $fn(c)$ 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 propozițiilor valide. Următoarea problemă constă în a arăta cum regulile de deducție pentru rescriere pot fi utilizate în mod practic.

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

2.5.1 Completitudinea rescrierii în subtermeni

Propozițiile 2.3.10 și 2.3.11 ne asigură că rescrierea este cea mai mică preordine închisă la \mathbf{SRew}_Γ compatibilă cu toate operațiile din \mathcal{A} . Teorema următoare ne spune ceva mai mult.

Teorema 2.5.1 *Rescrierea este cea mai mică preordine din \mathcal{A} închisă la \mathbf{SRew}_Γ .*

Demonstrație: Fie R_Γ cea mai mică relație pe \mathcal{A} care este închisă la \mathbf{R} , \mathbf{T} și \mathbf{SRew}_Γ . Din definiția 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{CS} .

Dar R_Γ este tranzitivă și reflexivă, deci este suficient să arătăm că R_Γ este închisă la $\mathbf{CA}\Sigma$, adică pentru fiecare $\sigma \in \Sigma_{s_1 \dots s_n, s'}$ și $a_i \in A_{s_i}$ că $a R_\Gamma d$ implică

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

Demonstrăm că mulțimea

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

este închisă la \mathbf{R} , \mathbf{T} și \mathbf{SRew}_Γ .

Reflexivitatea lui D rezultă din reflexivitatea lui R_Γ .

Presupunem că (a, u) și (u, d) sunt în D . Folosind tranzitivitatea lui R_Γ din $a R_\Gamma u$ și $u R_\Gamma d$ deducem $a R_\Gamma d$ și din $A_\sigma(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n) R_\Gamma A_\sigma(a_1, \dots, a_{i-1}, u, a_{i+1}, \dots, a_n)$ și $A_\sigma(a_1, \dots, a_{i-1}, u, a_{i+1}, \dots, a_n) R_\Gamma A_\sigma(a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n)$ deducem $A_\sigma(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n) R_\Gamma A_\sigma(a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n)$. Prin urmare $(a, d) \in D$. Deci D este tranzitivă.

Probăm că D este închisă la \mathbf{SRew}_Γ . Fie $(\forall X)l \overset{\circ}{=} s r$ if $H \in \Gamma$, un morfism $h : T_\Sigma(X) \rightarrow \mathcal{A}$ cu proprietatea că pentru orice $u \overset{\circ}{=} t v \in H$, există $a_{uv} \in A_t$ astfel încât $(h_t(u), a_{uv}) \in D$ și $(h_t(v), a_{uv}) \in D$. 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}_Γ din $h_t(u) R_\Gamma a_{uv}$ și $h_t(v) R_\Gamma a_{uv}$ pentru orice $u \overset{\circ}{=} t v \in H$ deducem

$$c[h_s(l)] R_\Gamma c[h_s(r)] \quad \text{și} \quad c'[h_s(l)] R_\Gamma c'[h_s(r)]$$

unde $c' = \sigma(a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_n)$ este un context.

Din $c'[h_s(l)] = A_\sigma(a_1, \dots, a_{i-1}, c[h_s(l)], a_{i+1}, \dots, a_n)$ și $c'[h_s(r)] = A_\sigma(a_1, \dots, a_{i-1}, c[h_s(r)], a_{i+1}, \dots, a_n)$ obținem

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

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

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

2.5.2 Rescriere într-un pas

Fie c în $T_\Sigma(A \cup \{\bullet\})_{s'}$ un context. Fie $(\forall X)l \overset{\circ}{=} s r$ if $H \in \Gamma$ și fie $h : T_\Sigma(X) \rightarrow \mathcal{A}$ un morfism astfel încât pentru orice $u \overset{\circ}{=} t v \in H$ există $a_{uv} \in A_t$ cu proprietățile $h_t(u) \overset{*}{\Rightarrow}_\Gamma a_{uv}$ și $h_t(v) \overset{*}{\Rightarrow}_\Gamma a_{uv}$. Dacă $c[h_s(l)] \neq c[h_s(r)]$ spunem că $c[h_s(l)]$ se rescrie într-un pas în $c[h_s(r)]$ și scriem

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

Propoziție 2.5.2 *Rescrierea este închiderea reflexivă și tranzitivă a rescrierii într-un pas.*

Demonstrație: Fie \Rightarrow^* închiderea reflexiv-tranzitivă a rescrierii într-un pas.

Deoarece $\Rightarrow \subseteq \Rightarrow_\Gamma^*$ și \Rightarrow_Γ^* este preordine deducem incluziunea $\Rightarrow^* \subseteq \Rightarrow_\Gamma^*$.

Reciproc, deoarece \Rightarrow^* este reflexivă și tranzitivă este suficient să arătăm că \Rightarrow^* este închisă la \mathbf{SRew}_Γ . Fie $(\forall X) l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$, $h : T_\Sigma(X) \rightarrow \mathcal{A}$ un morfism astfel încât pentru orice $u \stackrel{\circ}{=}_t v \in H$ există $a_{uv} \in A_t$ cu proprietățile $h_t(u) \stackrel{*}{\Rightarrow} a_{uv}$ și $h_t(v) \stackrel{*}{\Rightarrow} a_{uv}$ și 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 proprietățile $h_t(u) \stackrel{*}{\Rightarrow}_\Gamma a_{uv}$ și $h_t(v) \stackrel{*}{\Rightarrow}_\Gamma a_{uv}$. Avem cazurile:

1. Dacă $c[h(l)] = c[h(r)]$ deducem din reflexivitatea lui \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 și tranzitivă a rescrierii într-un pas. \square

În continuare rescrierea într-un pas va fi notată cu \Rightarrow_Γ sau \Rightarrow .

2.5.3 Considerații metodologice

Pentru a demonstra că $a \equiv_\Gamma b$ este suficient, în ipotezele de mai sus, să arătăm că $fn_s(a) = fn_s(b)$ și 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

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

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 această verificare și în caz de reușită se rescrie a în $c[h_s(r)]$. În caz de nereușită se continuă căutarile pentru a reajunge în situația de mai sus. Intrucât în timpul verificării uneia dintre condițiile $h_t(u) \downarrow_\Gamma h_t(v)$ se rescriu atât $h_t(u)$ cât și $h_t(v)$ în forma lor normală, fenomenul de mai sus se poate repeta se apelează la mecanismul numit backtracking.

Dacă Γ conține doar ecuații necondiționate, 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.

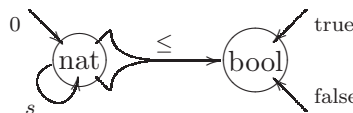
Remarcăm diferența esențială dintre o mulțime Γ de axiome și un program. Un program este o mulțime de axiome care verifică niște condiții și care sunt scrise respectând sintaxa unui limbaj de programare declarativă.

Am pus deja în evidență niște condiții suficiente pe care trebuie să le îndeplinească o mulțime Γ de axiome pentru a deveni program: rescrierea generată trebuie să fie confluentă și să aibă proprietatea de terminare.

Practic, de exemplu, într-o ecuație 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 și $bool$; trei constante $true$, $false$ de sort $bool$ și 0 de sort nat ; două simboluri de operații $s : nat \rightarrow nat$ și $\leq : nat \ nat \rightarrow bool$. Axiomele acceptate cu x și 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 $x = s(0)$ și $y = s(z)$.

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

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

Deoarece am obținut true putem rescrie $s(0) \leq s(z) \Rightarrow_{\Gamma} \text{true}$.

Deoarece prin rescrierea lui $s(0) \leq s(z)$ am obținut true putem rescrie $s(s(0)) \leq s(s(z)) \Rightarrow_{\Gamma} \text{true}$.

2.6 UNIFICARE

În semantica operațională a limbajelor de specificație 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 substituție(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 și 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 **substituție** vom înțelege un morfism între două algebre libere.

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

De fapt mulțimea finită de egalități formale este privită ca un sistem de ecuații. Și de aici poate proveni denumirea de ecuație folosită ca un sinonim pentru egalitățile formale. Acest subcapitol descrie modul de rezolvare a unui astfel de sistem chiar în algebra liberă din care provin termenii ecuațiilor. Cuvântul unificator înseamnă de fapt soluție în algebra liberă din care provin termenii ecuațiilor.

Algoritmul prezentat aici este ales pentru simplitatea sa. Menționăm existența unor algoritmi mai eficienți.

2.6.1 Algoritmul de unificare

Vom lucra cu două liste: soluție și de rezolvat. Inițial lista soluție este vidă și lista de rezolvat conține mulțimea ecuațiilor de unificat.

Algoritmul de unificare constă în execuția nedeterministă a următorilor trei pași.

1. **Scoate.** Se scoate din lista de rezolvat orice ecuație de forma $t \stackrel{\circ}{=} t$.
2. **Descompune.** Orice ecuație din lista de rezolvat de forma

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

se înlocuiește cu ecuațiile $a_1 \stackrel{\circ}{=} b_1, a_2 \stackrel{\circ}{=} b_2, \dots, a_n \stackrel{\circ}{=} b_n$,

3. **Elimină.** Orice ecuație 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 soluție. În toate celelalte ecuații x se substituie cu t .

Algoritmul este oprit cu concluzia inexistenței unui unificator în următoarele două cazuri.

- 1) Existența în lista de rezolvat a unei ecuații de forma $f(a_1, a_2, \dots, a_n) \stackrel{\circ}{=} g(b_1, b_2, \dots, b_m)$ cu $f \neq g$,
- 2) Existența în lista de rezolvat a unei ecuații de forma $x \stackrel{\circ}{=} t$ sau $t \stackrel{\circ}{=} x$ în care x este o variabilă care apare în t și este diferită de t .

Oprirea normală a algoritmului se face prin epuizarea listei de rezolvat, caz în care, așa cum vom proba mai jos, lista soluție 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 în funcție de pasul algoritmului utilizat are următoarea comportare
 - (a) **Scoate**: rămâne egal sau se micșorează,
 - (b) **Descompune**: rămâne egal,
 - (c) **Elimină**: se micșorează
2. numărul aparițiilor simbolurilor (semnelor) care apar în lista de rezolvat, care se micșorează în cele două cazuri care ne mai interesează **Scoate** și **Descompune**.

2.6.3 Corectitudine

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

Deoarece pentru pasul **Scoate** afirmația este evidentă ne referim doar la ceilalți pași.

Descompune: Observăm că pentru orice substituție s egalitatea

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

este echivalentă cu

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

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

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

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

deoarece $(x \leftarrow t; s)(x) = s(t) = s(x)$ și $(x \leftarrow t; s)(y) = s(y)$ pentru orice altă variabilă y . Prin urmare pentru o astfel de substituție

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

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

Algoritmul este oprit cu concluzia inexistenței unui unificator deoarece în cele două situații menționate se constată că mulțimea unificatorilor este vidă.

Să observăm că variabilele care apar în membrul stâng al ecuațiilor din lista soluție sunt diferite două câte două și nu mai apar în nici una dintre celelalte ecuații din cele două liste.

Faptul poate fi dovedit prin inducție.

Menționăm că aplicarea primilor doi pași ai algoritmului nu modifică lista soluție și nu produc apariții noi de variabile în cele două liste.

Fie $x \stackrel{\circ}{=} t$ ecuația introdusă în lista soluție prin aplicarea pasului **Elimină**. Deoarece variabilele din membrul stâng al listei soluție precedentă nu apar în celelalte ecuații rezultă că variabila x este diferită de celelalte variabile care apar în membrul stâng al ecuațiilor din lista soluție. Prin urmare, variabilele din membrul stâng al noii liste soluție sunt diferite între ele.

În plus prin substituirea lui x cu t în celelalte ecuații variabila x dispăre din ele deoarece x nu apare în t . Deoarece nici x și nici variabilele din membrul stâng al listei soluție precedentă nu apar în t rezultă că după efectuarea substituției lui x cu t variabilele din membrul stâng al listei soluție nu apar în restul ecuațiilor.

Să presupunem că algoritmul s-a terminat prin epuizarea listei de rezolvat. Să notăm cu k numărul ecuațiilor din lista soluție și cu $x_i \stackrel{\circ}{=} t_i$ pentru $i \in [k]$ ecuațiile din ea.

Fie U substituția definită prin

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

Definiția 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 substituție s compunerea $U; s$ este tot un unificator. Vom arăta că orice alt unificator este de această formă. Fie u un alt unificator, adică $u(x_i) = u(t_i)$ pentru orice $i \in [k]$. Observăm că $U; u = u$, căci $u(U(x_i)) = u(t_i) = u(x_i)$ pentru orice $i \in [k]$ și $u(U(y)) = u(y)$ 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 substituție. În plus observăm că el este idempotent deoarece $U; U = U$.

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

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

Reamintim că relația de rescriere este confluentă dacă

$$(\forall a, b, c) [(a \xRightarrow{*} b \text{ și } a \xRightarrow{*} c) \text{ implică } (\exists d) (b \xRightarrow{*} d \text{ și } c \xRightarrow{*} d)].$$

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

$$(\forall a, b, c) [(a \Rightarrow b \text{ și } a \Rightarrow c) \text{ implică } (\exists d) (b \xRightarrow{*} d \text{ și } c \xRightarrow{*} d)].$$

Reamintim că rescrierea are proprietatea de terminare, dacă nu există șiruri $\{a_n\}_n$ cu proprietatea $a_n \Rightarrow a_{n+1}$ pentru orice n număr natural.

Propoziție 2.7.2 Dacă rescrierea este local confluentă și are proprietatea de terminare atunci rescrierea este confluentă.

Demonstrație: Reamintim că proprietatea de terminare ne asigură pentru orice a existența unei forme normale n cu proprietatea $a \xRightarrow{*} n$. Vom demonstra unicitatea formei normale fapt ce implică confluența. Notăm cu N mulțimea formelor normale. Fie

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

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

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

Deoarece a nu este formă normală există elemntul b cu proprietățile $a \Rightarrow b$ și $b \xRightarrow{*} n_1$. La fel există c cu proprietățile $a \Rightarrow c$ și $c \xRightarrow{*} n_2$. Deoarece rescrierea este local confluența există elementul d cu proprietățile $b \xRightarrow{*} d$ și $c \xRightarrow{*} d$. Deoarece terminarea implică existența formei normale deducem $\exists n \in N$ cu proprietatea $d \xRightarrow{*} n$.

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

Dacă $n \neq n_1$ deoarece $b \xRightarrow{*} n_1$ și $b \xRightarrow{*} n$ deducem $b \in M$. Dacă $n \neq n_2$ deoarece $c \xRightarrow{*} n$ și $c \xRightarrow{*} n_2$ deducem $c \in M$.

Probăm unicitatea formei normale. Raționând prin absurd deducem că M este nevidă. Plecând de la $a_1 \in M$ aplicând observația de mai sus deducem existența lui $a_2 \in M$ cu proprietatea $a_1 \Rightarrow a_2$. Repetând raționamentul printr-o inducție deducem existența unui șir $\{a_n\}_n$ cu proprietatea $a_n \Rightarrow a_{n+1}$ pentru orice n natural. Deci rescrierea nu are proprietatea de terminare, în contradicție cu ipoteza. \square

2.8 RESCRIERE MODULO ECUATII

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

Deoarece unele ecuații, de exemplu comutativitatea, nu pot fi folosite ca axiome pentru rescriere, s-a găsit o altă cale pentru utilizarea acestora și anume rescrierea modulo ecuații. Mulțimea E reprezintă ecuațiile necondiționate care în loc de a fi folosite ca axiome vor fi utilizate pentru rescrierea modulo ecuații.

2.8.1 Motivare semantică

Fie E o mulțime de Σ -ecuații și Γ o mulțime de Σ -ecuații condiționate.

Propoziție 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 $\mathcal{A} \models \Gamma$.

Demonstrație: 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 și

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

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 $h_s(l) = h_s(r)$.

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

Corolar 2.8.2 Orice subalgebră a unei Γ -algebre este Γ -algebră.

Propoziție 2.8.3 Fie $e : \mathcal{A} \longrightarrow \mathcal{B}$ un morfism din categoria Σ -algebrelor cu toate componentele surjective și E o mulțime de ecuații necondiționate. Dacă $\mathcal{A} \models E$, atunci $\mathcal{B} \models E$.

Demonstrație: Fie $(\forall X)l \stackrel{\circ}{=}_s r \in E$ și $h : T_\Sigma(X) \longrightarrow \mathcal{B}$ un morfism. Deoarece Σ -algebra liberă $T_\Sigma(X)$ este *proiectivă*, conform propoziției 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 ecuație $(\forall X)l \stackrel{\circ}{=}_s r$ din E , adică \mathcal{B} satisface E . \square

Reamintim că $\mathcal{A}_\Gamma = \mathcal{A}/_{\equiv_\Gamma}$. Pentru $a, c \in \mathcal{A}_s$ reamintim deasemenea că $a \equiv_\Gamma c$ dacă și 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 $\rho : \mathcal{A} \longrightarrow \mathcal{A}_E$ și $j : \mathcal{A}_E \longrightarrow \mathcal{A}_{E, \Gamma}$ morfismele canonice de factorizare.

Observația 2.8.4 Algebrele $\mathcal{A}_{\Gamma \cup E}$ și $\mathcal{A}_{E, \Gamma}$ sunt izomorfe.

Demonstrație: Ideea demonstrației este de a arăta că $\mathcal{A}_{E, \Gamma}$ are proprietățile care caracterizează abstracție de un izomorfism Σ -algebra $\mathcal{A}_{\Gamma \cup E}$.

Evident $\mathcal{A}_{E, \Gamma} \models \Gamma$. Deoarece morfismul $j : \mathcal{A}_E \rightarrow \mathcal{A}_{E, \Gamma}$ are toate componentele surjective și $\mathcal{A}_E \models E$ rezultă din propoziția 2.8.3 că $\mathcal{A}_{E, \Gamma} \models E$, deci $\mathcal{A}_{E, \Gamma} \models \Gamma \cup E$.

Pentru restul demonstrației puteți utiliza figura 2.1.

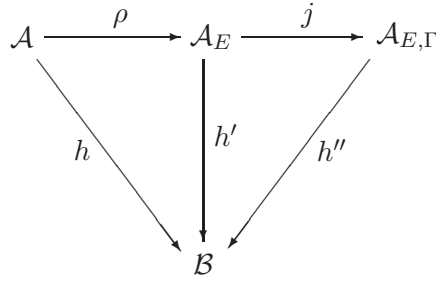


Figure 2.1:

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 $h' : \mathcal{A}_E \rightarrow \mathcal{B}$ astfel încât $\rho; h' = h$. Deoarece $\mathcal{B} \models \Gamma$ rezultă că există un unic morfism $h'' : \mathcal{A}_{E, \Gamma} \rightarrow \mathcal{B}$ astfel încât $j; h'' = h'$. Atunci $(\rho; j); h'' = h$.

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

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

Demonstrație: Fie $t : \mathcal{A} \rightarrow \mathcal{A}_{\Gamma \cup E}$ morfismul canonic de factorizare și $i : \mathcal{A}_{E, \Gamma} \rightarrow \mathcal{A}_{\Gamma \cup E}$ 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 și ținând 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 și ținând cont de egalitatea $\rho; j; i^{-1}$ deducem $j_s(\rho_s(a)) = j_s(\rho_s(c))$ deci $\rho_s(a) \equiv_{\Gamma} \rho_s(c)$. \square

Propoziția 2.8.5 ne dă o metodă de a evita ecuațiile E care nu pot fi folosite ca axiome în rescriere.

Presupunem că avem de demonstrat $a \equiv_{\Gamma \cup E} c$ în \mathcal{A} . Propoziția 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 dori să folosim metoda dată în lecțiile anterioare, pentru a o demonstra vom ajunge la rescrieri în \mathcal{A}_E .

2.8.2 Rescrierea modulo o relație de echivalență

Rescrierea în \mathcal{A}_E se numește *rescriere de clase*. Deoarece clasele de echivalență ale relației \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 aceleiași clase și 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 relație de echivalență** \sim ce poate fi cu ușurință formalizat. Conceptul de rescriere modulo o relație de echivalență \sim se obține din conceptul clasic de rescriere la care se adaugă o nouă regulă de deducție numită regula claselor.

Cl_~ $a \sim_s b$ implică $a \stackrel{\circ}{=}_s b$

De remarcat faptul că \mathbf{Cl}_\sim este o regulă deductivă mai puternică decât \mathbf{R} , prin urmare în rescrierea modulo \sim regula \mathbf{Cl}_\sim va înlocui pe \mathbf{R} .

Utilizarea fără restricții a regulii claselor duce la pierderea proprietății de terminare. Un element poate fi rescris în altul din aceeași clasă, acesta la rândul lui poate fi rescris în alt element echivalent și așa mai departe. Observăm că un șir finit de astfel de rescrieri poate fi înlocuit cu una singură deoarece toate elementele care apar în rescriere sunt din aceeași clasă. Prin urmare, restricția de a nu folosi de două ori consecutiv regula claselor nu restrânge puterea acestei reguli și este suficientă pentru implementare.

Preliminarii

Fie un morfism $h : \mathcal{A} \longrightarrow \mathcal{B}$ și \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 $nr_\bullet^B(c)$ numărul de apariții ale lui \bullet în c . Deoarece $h^\bullet; nr_\bullet^B = nr_\bullet$, rezultă că pentru c din $T_\Sigma(A \cup \{\bullet\})$, $h^\bullet(c)$ este context dacă și 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 și el surjectiv pe componente fapt care se demonstrează prin inducție structurală probând că pentru orice $c' \in T_\Sigma(B \cup \{\bullet\})$ există $c \in T_\Sigma(A \cup \{\bullet\})$ cu proprietatea $h^\bullet(c) = c'$. În plus pentru fiecare context $c' \in T_\Sigma(B \cup \{\bullet\})$ există un context $c \in T_\Sigma(A \cup \{\bullet\})$ astfel încât $h^\bullet(c) = c'$.

Corectitudinea

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

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

Demonstrație: Vom arăta că mulțimea de propoziții din \mathcal{A}

$$D = \{a \stackrel{\circ}{=} d : h_s(a) \xRightarrow{*B}_\Gamma h_s(d)\}$$

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

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

\mathbf{T} . Fie $a \stackrel{\circ}{=} s b \in D$ și $b \stackrel{\circ}{=} s c \in D$. Prin urmare $h_s(a) \xRightarrow{*B}_\Gamma h_s(b)$ și $h_s(b) \xRightarrow{*B}_\Gamma h_s(c)$. Datorită tranzitivității $h_s(a) \xRightarrow{*B}_\Gamma h_s(c)$, deci $a \stackrel{\circ}{=} c \in D$.

\mathbf{SRew}_Γ . Fie $(\forall X)l \stackrel{\circ}{=} s r$ if $H \in \Gamma$ și $f : T_\Sigma(X) \longrightarrow \mathcal{A}$ un morfism astfel încât pentru orice $u \stackrel{\circ}{=} t v$ există $a_{uv} \in A_t$ cu proprietățile $f_t(u) \stackrel{\circ}{=} t a_{uv} \in D$ și $f_t(v) \stackrel{\circ}{=} t a_{uv} \in D$. Prin urmare $h_t(f_t(u)) \xRightarrow{*B}_\Gamma h_t(a_{uv})$ și $h_t(f_t(v)) \xRightarrow{*B}_\Gamma h_t(a_{uv})$.

Pentru orice context c din $T_\Sigma(A \cup \{\bullet\})$, deoarece $\xRightarrow{*B}_\Gamma$ este închis la \mathbf{SRew}_Γ , folosind contextul $h^\bullet(c)$ și morfismul $f; h : T_\Sigma(X) \longrightarrow \mathcal{B}$ obținem

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

Deoarece

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

și analog

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

rezultă că

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

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

Deoarece D este închisă la regulile $\mathbf{CI}_{Ker(h)}$, \mathbf{T} și \mathbf{SRew}_{Γ} rezultă că

$$\xRightarrow{h} \subseteq D$$

fapt care implică concluzia. \square

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

Corolar 2.8.7 Dacă $a \xrightarrow{*_{\Gamma}} d$ atunci $h(a) \xrightarrow{*_{\Gamma}^{\mathcal{B}}} h(d)$.

Demonstrație: Putem aplica propoziția folosind $\xrightarrow{*_{\Gamma}} \subseteq \xRightarrow{h}$.

Completitudinea

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

Demonstrație: Fie mulțimea

$$D = \{h_s(a) \stackrel{\circ}{=} h_s(d) : a \xRightarrow{h} d\}.$$

Observăm că $h_s(u) \stackrel{\circ}{=} h_s(v) \in D$ implică $u \xRightarrow{h} v$. Din ipoteză există a, b astfel încât $h_s(u) = h_s(a)$, $a \xRightarrow{h} d$ și $h_s(d) = h_s(v)$, prin urmare $u \xRightarrow{h} v$.

Probăm că D este închisă la \mathbf{R}, \mathbf{T} și \mathbf{SRew}_{Γ} .

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

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

Deoarece $h(v) = h(u)$ deducem $v \xRightarrow{h} u$. Folosind și $a \xRightarrow{h} v$ și $u \xRightarrow{h} d$ deducem prin \mathbf{T} că $a \xRightarrow{h} d$, 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 $f : T_{\Sigma}(X) \rightarrow \mathcal{B}$ astfel încât pentru orice $u \stackrel{\circ}{=}_t v \in H$ există $b_{uv} \in B_t$ cu proprietățile $f_t(u) \stackrel{\circ}{=} b_{uv} \in D$ și $f_t(v) \stackrel{\circ}{=} b_{uv} \in D$. 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 și 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

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

deducem

$$g_t(u) \xrightarrow{h} a_{uv} \text{ și } g_t(v) \xrightarrow{h} a_{uv}.$$

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

Prin urmare

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

Dar,

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

și

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

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

Corolar 2.8.9 *a se rescrie modulo o echivalență \sim în b, dacă și numai dacă clasa de echivalență a lui a se rescrie în clasa de echivalență a lui b.*

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

Comentariu. În acest moment cititorul poate înțelege de ce am preferat să ne bazăm lecțiile 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) și în momentul când ajung la rescrierile modulo ecuații să expună aceeași teorie privind acest alt tip de rescriere. În concluzie rescrierile într-o algebra unifică rescrierile într-o algebră liberă cu rescrierile modulo ecuații.

2.9 DEMONSTRAREA ECUAȚIILOR CONDIȚIONATE

Pentru două mulțimi de Σ -ecuații condiționate Γ și Γ' , vom scrie $\Gamma \models_\Sigma \Gamma'$ dacă și 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 **consecință semantică** a lui Γ .

Până acum ne-am ocupat numai de ecuațiile necondiționate adevărate în orice Γ -algebră. În această subcapitol vom da o cale prin care se poate demonstra că o ecuație condiționată γ este adevărată în orice Γ -algebră, adică $\Gamma \models_\Sigma \gamma$.

2.9.1 Preliminarii

Se arată ușor că

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

Întradevăr pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{B}$ din $\mathcal{B} \models_\Sigma \{(\forall X)u \stackrel{\circ}{=}_t v \mid u \stackrel{\circ}{=}_t v \in H\}$ deducem $h_t(u) = h_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in H$. Prin urmare din $\mathcal{B} \models_\Sigma (\forall X)l \stackrel{\circ}{=}_s r \text{ if } H$ 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 $\mathcal{B} \models_\Sigma (\forall X)l \stackrel{\circ}{=}_s r$.

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

Lemă 2.9.1

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

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

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

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

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

Corolar 2.9.2

$$\Gamma \models_{\Sigma} (\forall \emptyset)l \stackrel{\circ}{=} r \text{ if } H \text{ dacă și numai dacă } \Gamma \cup \{(\forall \emptyset)u \stackrel{\circ}{=} v \mid u \stackrel{\circ}{=} v \in H\} \models_{\Sigma} (\forall \emptyset)l \stackrel{\circ}{=} r.$$

Demonstrație: Demonstrația se va face printr-un șir de echivalențe. Plecăm de la

$$\Gamma \models_{\Sigma} (\forall \emptyset)l \stackrel{\circ}{=} r \text{ if } H$$

care prin definiție este echivalentă cu

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

Conform lemei afirmația de mai sus este echivalentă cu

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

care este echivalentă cu

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

adică cu

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

Lema și corolarul precedent ne permit să spargem o ecuație condiționată de demonstrat în părțile ei componente. Din păcate acest fapt este posibil numai pentru ecuațiile condiționate fără variabile. Teorema constantelor ne va permite să înlocuim demonstrarea unei ecuații condiționate cu variabile cu o ecuație condiționată fără variabile.

2.9.2 Schimbarea semnăturii

Prima etapă constă în înlocuirea ecuației condiționate cu o alta cu o alta “echivalentă” dintr-o algebră inițială. Pentru aceasta este necesară schimbarea semnăturii prin transformarea variabilelor din ecuația condiționată de demonstrat în constante (simboluri de operații fără argumente). Această tehnică de demonstrație a mai fost folosită în subcapitolul 1.4.4.

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

Vom nota cu $j : \Sigma \longrightarrow \Sigma_X$ morfismul incluziune de semnături. Gândiți-vă la el ca la o funcție incluziune de mulțimi deoarece conceptul de morfism de semnături 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 puneți mai mult accent pe cuvântul uituc. Un alt exemplu de functor uituc, de la inele la grupuri, este cel care uită înmulțirea 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.

În categoria Σ_X -algebrelor Alg_{Σ_X} vom utiliza o notație specială și anume o Σ_X -algebră va fi scrisă ca o pereche (\mathcal{A}, a) unde \mathcal{A} este o Σ -algebră și $a : X \longrightarrow \mathcal{A}$ este o funcție S -sortată. Prin Σ -algebra \mathcal{A} se dau suporturile și operațiile corespunzătoare simbolurilor din Σ iar prin funcția a se dau operațiile corespunzătoare simbolurilor din X , adică operația 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 și numai dacă $h : \mathcal{A} \longrightarrow \mathcal{B}$ este morfism de Σ -algebre și $a; h = b$. Explicit egalitatea $a; h = b$ spune că

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

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

Având în vedere notațiile de mai sus observăm că

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

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

Un cititor mai puțin experimentat poate considera cele două egalități de mai sus drept definiții. 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 semnăturii Σ cu Σ_X impune translatarea Σ -axiomelor în Σ_X -axiome.

2.9.3 Translatarea ecuațiilor

Vom nota cu Y o mulțime 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 $Mod(j)(T_{\Sigma_X}(Y)) = T_{\Sigma}(X \cup Y)$.

Observația 2.9.3 Pentru orice Σ_X -algebră (\mathcal{M}, m) funcția

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

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

Demonstrație: Observăm că $F(h)$ este restricția 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 proprietățile

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

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 restricția lui g la $T_{\Sigma}(Y)$ coincide cu u .

Probăm injectivitatea lui F . Fie $h_1 : T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m)$ și $h_2 : T_{\Sigma_X}(Y) \longrightarrow (\mathcal{M}, m)$ două Σ_X -morfisme cu proprietatea $F(h_1) = F(h_2)$. Deoarece h_1 și 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 și h_2 sunt morfisme definite pe $T_{\Sigma}(X \cup Y)$ deducem că $h_1 = h_2$. \square

Faptele expuse până acum ca și lema următoare au un caracter tehnic. Deoarece semnatura a fost schimbată este necesar ca axiomele să fie rescrise în noua semnătură. Mulțimea Y reprezintă variabilele cuantificate universal dintr-o axiomă. Ele nu sunt afectate prin trecerea la noua semnătură. Schimbarea semnăturii axiomelor este mai mult formală caci în noile axiome constantele obținute din variabilele din X nu sunt practic folosite. Aceasta este motivul pentru care în lema și cololarul care urmează funcția m , prin care sunt date constantele, apare numai în membrul stâng, membrul drept fiind independent de aceasta.

Fiecărei Σ -ecuații condiționate

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

ii atașăm o altă Σ_X -ecuație condiționată

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

Funcția J duce o Σ -ecuație necondiționată într-o Σ_X -ecuație necondiționată.

Lemă 2.9.4 Satisfacerea. Pentru orice Σ_X -algebră (\mathcal{M}, m)

$$(\mathcal{M}, m) \models_{\Sigma_X} J(\gamma) \text{ dacă și numai dacă } \mathcal{M} \models_{\Sigma} \gamma.$$

Demonstrație: Observăm că $(\mathcal{M}, m) \models_{\Sigma_X} J(\gamma)$ este, prin definiție, echivalentă cu

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

prin urmare echivalentă cu

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

adică echivalentă cu

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

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

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

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

Pentru o mulțime Γ de Σ -ecuații condiționate notăm $J(\Gamma) = \{J(\gamma) \mid \gamma \in \Gamma\}$.

Corolar 2.9.5 Pentru orice Σ_X -algebră (\mathcal{M}, m) și orice mulțime Γ de Σ -ecuații condiționate

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

2.9.4 Teorema constantelor

Teorema 2.9.6 Teorema constantelor:

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

Demonstrație: La început vom menționa forme echivalente pentru cei doi membri ai echivalenței din enunț.

Membrul stâng

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

este echivalent succesiv cu:

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

Membrul drept

$$J(\Gamma) \models_{\Sigma_X} (\forall \emptyset)l \overset{\circ}{=}_s r \text{ **if** } H$$

este echivalent succesiv cu:

1. Oricare ar fi Σ_X -algebra (\mathcal{M}, m) , dacă $(\mathcal{M}, m) \models_{\Sigma_X} J(\Gamma)$, atunci $(\mathcal{M}, m) \models_{\Sigma_X} (\forall \emptyset)l \overset{\circ}{=}_s r$ **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 $u \overset{\circ}{=}_t v \in H$, atunci $h_s(l) = h_s(r)$.

Presupunem că $\Gamma \models_{\Sigma} (\forall X)l \overset{\circ}{=}_s r$ **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 $u \overset{\circ}{=}_t v \in H$.

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

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

Deoarece $(\mathcal{M}, h_{/X}) \models_{\Sigma_X} J(\Gamma)$, $h : T_{\Sigma_X} \longrightarrow (\mathcal{M}, h_{/X})$ este un Σ_X -morfism și $h_t(u) = h_t(v)$ pentru orice $u \overset{\circ}{=}_t v \in H$ din ipoteză deducem $h_s(l) = h_s(r)$. \square

Această teoremă ne permite să eliminăm cuantificatorul universal prin înlocuirea variabilelor cuantificate prin simboluri de operații constante. Astfel primul obiectiv de a ne muta într-o algebră inițială a fost atins.

Teorema următoare nu face decât să tragă concluziile, furnizând o cale prin care putem încerca demonstrarea ecuațiilor condiționate. Problema care rămâne este demonstrarea corectitudinii programului obținut folosind noul set de axiome în ipoteza că programul dat de axiomele Γ este corect.

Teorema 2.9.7 Teorema deducției:

$$\Gamma \models_{\Sigma} (\forall X)l \overset{\circ}{=}_s r \text{ **if** } H \text{ dacă și numai dacă } J(\Gamma) \cup \{(\forall \emptyset)u \overset{\circ}{=}_t v \mid u \overset{\circ}{=}_t v \in H\} \models_{\Sigma_X} (\forall \emptyset)l \overset{\circ}{=}_s r.$$

Demonstrație: Conform teoremei constantelor

$$\Gamma \models_{\Sigma} (\forall X)l \overset{\circ}{=}_s r \text{ **if** } H$$

este echivalentă cu

$$J(\Gamma) \models_{\Sigma_X} (\forall \emptyset)l \overset{\circ}{=}_s r \text{ **if** } H$$

care conform corolarului 2.9.2 este echivalentă cu

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

În concluzie pentru demonstrarea Σ -ecuației condiționate $(\forall X)l \overset{\circ}{=}_s r$ **if** H putem proceda astfel

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

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

2.10 PERECHI CRITICE

Toate algebrele sunt libere. Presupunem în plus că Γ conține numai ecuații necondiționate.

Mai menționăm că membrul stâng al unei ecuații nu poate fi format numai dintr-o variabilă, deoarece în caz contrar orice expresie ar putea fi rescrisă. O consecință 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 $a \xrightarrow{*}_\Gamma d$, atunci $c[a] \xrightarrow{*}_\Gamma c[d]$.

Demonstrație: Menționăm că lema arată mai mult decât închiderea la contexte, căci în enunț nu se presupune nimic despre c .

Probăm prin inducție structurală după c .

Dacă $c = \bullet$ concluzia coincide cu ipoteza.

Dacă $c \in A$ utilizăm reflexivitatea.

Dacă $c = \sigma(c_1, c_2, \dots, c_n)$ din ipoteza de inducție $c_i[a] \xrightarrow{*}_\Gamma c_i[d]$ pentru orice i , prin urmare aplicând \mathbf{CS} deducem

$$A_\sigma(c_1[a], c_2[a], \dots, c_n[a]) \xrightarrow{*}_\Gamma A_\sigma(c_1[d], c_2[d], \dots, c_n[d]),$$

prin urmare deoarece

$$\begin{aligned} A_\sigma(c_1[a], c_2[a], \dots, c_n[a]) &= A_\sigma(\bullet \leftarrow a(c_1), \bullet \leftarrow a(c_2), \dots, \bullet \leftarrow a(c_n)) = \\ &= \bullet \leftarrow a(\sigma(c_1, c_2, \dots, c_n)) = \bullet \leftarrow a(c) = c[a] \text{ și} \\ A_\sigma(c_1[d], c_2[d], \dots, c_n[d]) &= A_\sigma(\bullet \leftarrow d(c_1), \bullet \leftarrow d(c_2), \dots, \bullet \leftarrow d(c_n)) = \\ &= \bullet \leftarrow d(\sigma(c_1, c_2, \dots, c_n)) = \bullet \leftarrow d(c) = c[d] \text{ tragem concluzia} \\ c[a] &\xrightarrow{*}_\Gamma c[d]. \quad \square \end{aligned}$$

Lemă 2.10.2 Pentru orice $c \in T_\Sigma(A \cup \{\bullet\})$ dacă $a \downarrow_\Gamma d$, atunci $c[a] \downarrow_\Gamma c[d]$.

Demonstrație: Deoarece există m astfel încât $a \xrightarrow{*}_\Gamma m$ și $d \xrightarrow{*}_\Gamma m$ din lema precedentă deducem $c[a] \xrightarrow{*}_\Gamma c[m]$ și $c[d] \xrightarrow{*}_\Gamma c[m]$ de unde rezultă concluzia. \square

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

Demonstrație: Se face prin inducție structurală după c . \square

Definiția 2.10.4 Fie $(\forall X_1)l_1 \stackrel{\circ}{=}_{s'} r_1$ și $(\forall X_2)l_2 \stackrel{\circ}{=}_{s''} r_2$ 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 nu este o variabilă. Fie u un cel mai general unificator al lui d și al lui l_2 . Elementele

$$u(r_1) \text{ și } u(c[r_2])$$

formează o **pereche critică**. \square

Pe scurt notăm mulțimea perechilor critice

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

Propozițiile 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 propoziții să devină fără variabile comune.

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

Demonstrație: Observăm că

$$u(l_1) \Rightarrow_{\Gamma} u(r_1) \text{ și } u(l_1) = u^{\bullet}(c)[u(d)] = u^{\bullet}(c)[u(l_2)] \Rightarrow_{\Gamma} u^{\bullet}(c)[u(r_2)] = u(c[r_2]).$$

Deoarece rescrierea este local confluentă deducem că

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

Teorema 2.10.6 *Dacă $\mathbf{PC}_{\Gamma} \subseteq \downarrow_{\Gamma}$, atunci rescrierea este local confluentă.*

Demonstrație: Fără a micșora generalitatea vom presupune că toate mulțimile de variabile cu care lucrăm sunt disjuncte două câte două și că variabilele nou introduse sunt distincte între ele și nu aparțin altor mulțimi de variabile.

Presupunem că

$$a \Rightarrow a_1 \text{ și } a \Rightarrow a_2.$$

Prin urmare pentru $i \in [2]$ există $(\forall X_i)l_i \stackrel{\circ}{=} r_i \in \Gamma$, contextul $c_i \in \mathcal{A}[z_i]$ și morfismul $h_i : T_{\Sigma}(X_i) \rightarrow \mathcal{A}$ astfel încât $a = c_i[h_i(l_i)]$ și $a_i = c_i[h_i(r_i)]$.

Cei doi subarbori $h_1(l_1)$ și $h_2(l_2)$ ai lui a pot fi disjuncți (cazul 1 de mai jos) sau incluși unul în celălalt (cazurile 2 și 3 de mai jos). În cazurile 2 și 3 vom presupune fără a restrânge generalitatea că $h_2(l_2)$ este subarbore a lui $h_1(l_1)$.

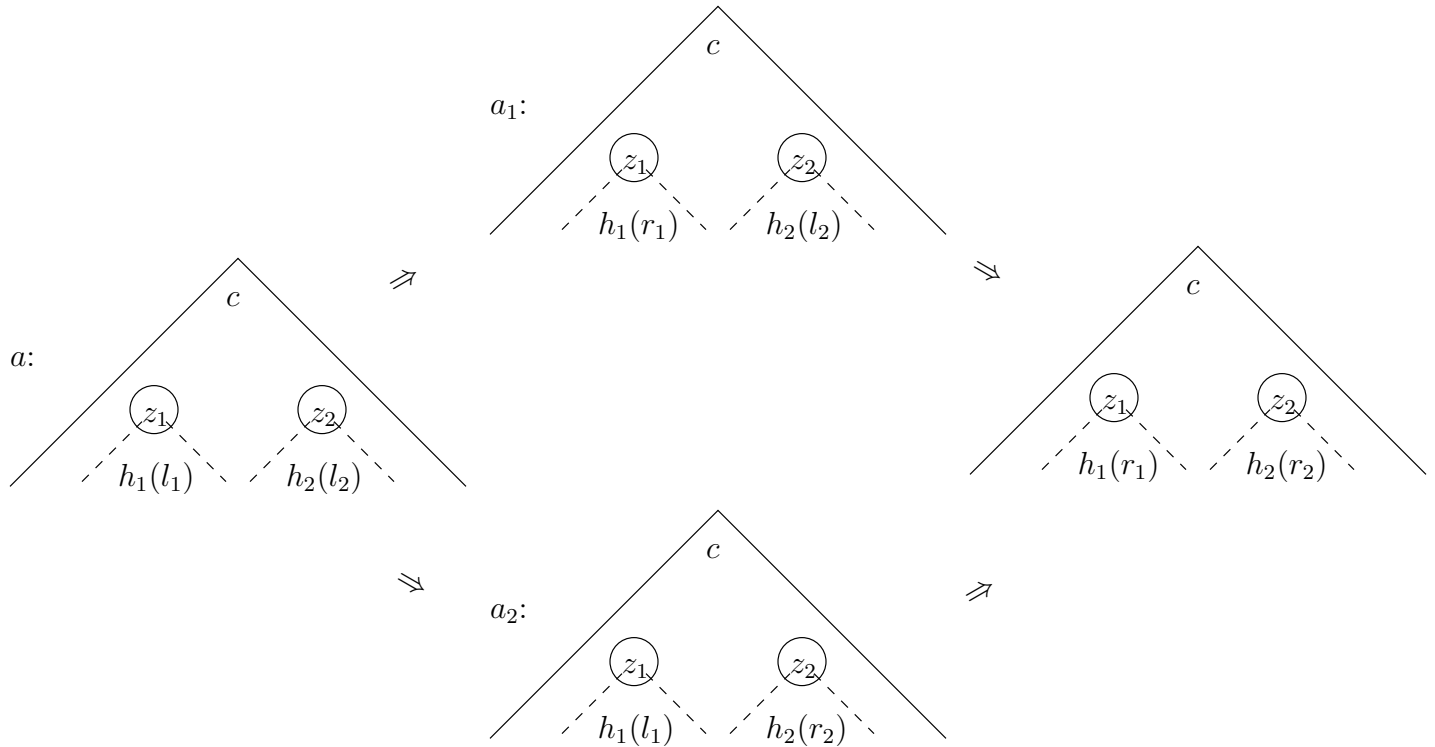


Figure 2.2: Cazul 1. Subarbori disjuncți

Cazul 1 (fără suprapunere). Există $c \in T_{\Sigma}(A \cup \{z_1, z_2\})$ cu câte o apariție a lui z_1 și z_2 astfel încât

$$c_1 = (z_2 \leftarrow h_2(l_2))(c) \text{ și } c_2 = (z_1 \leftarrow h_1(l_1))(c).$$

Deoarece z_1 nu apare în $h_2(l_2)$ și 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

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

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).

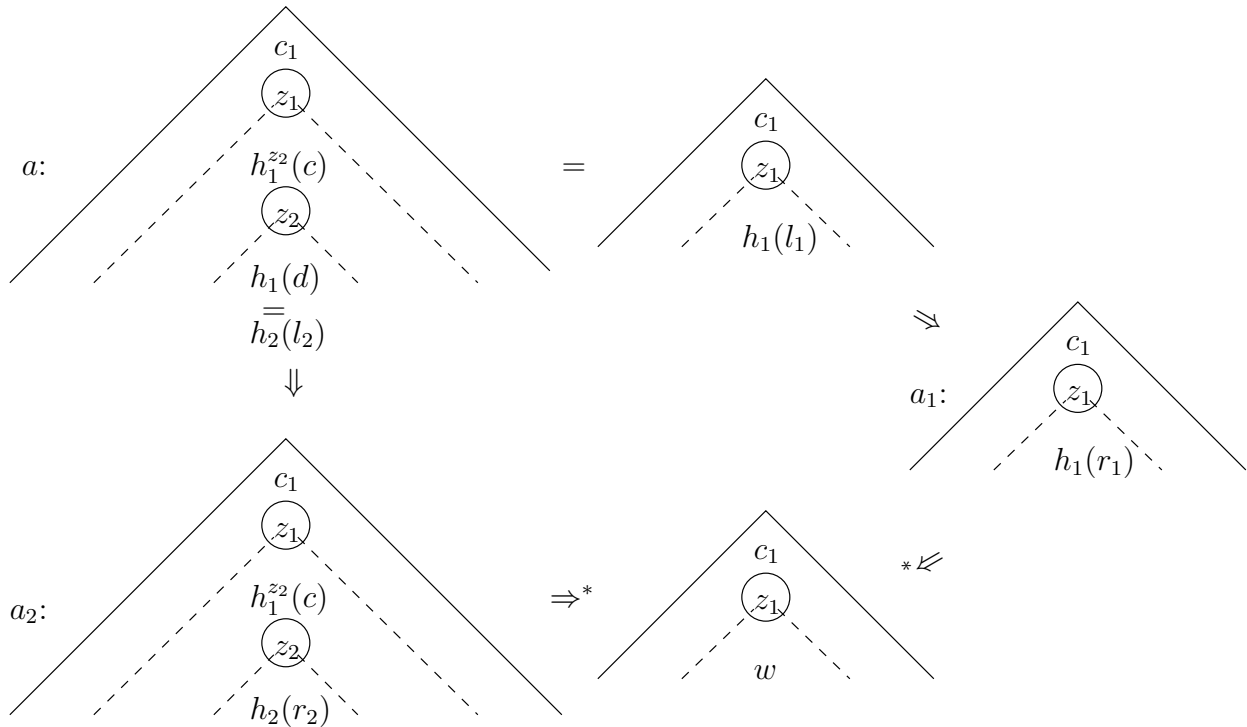


Figure 2.3: Cazul 2. Cu suprapunere critică

Cazul 2(cu suprapunere critică, adică $h_2(l_2)$ este imaginea prin h_1 al unui subtermen d al lui l_1 care nu este variabilă).

Prin urmare

$l_1 = c[d]$ unde c este un context cu variabila distinsă z_2 ,

$$h_1(d) = h_2(l_2) \quad \text{și} \quad c_2 = (z_1 \leftarrow h_1^{z_2}(c))(c_1).$$

Ultima egalitate este motivată de următoarele calcule

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

Fie

$$h : T_{\Sigma}(X_1 \cup X_2) \longrightarrow \mathcal{A}$$

unicul morfism care extinde atât pe h_1 cât și pe h_2 . Observăm că h este un unificator pentru d și l_2 . Fie $u : T_{\Sigma}(X_1 \cup X_2) \longrightarrow \mathcal{T}$ un cel mai general unificator al lui d și l_2 și $t : \mathcal{T} \longrightarrow \mathcal{A}$ unicul morfism cu proprietatea $u; t = h$.

Doarece d nu este variabilă rezultă că $u(r_1)$ și $u(c[r_2])$ formează o pereche critică. Din ipoteză deducem $u(r_1) \downarrow_{\Gamma} u(c[r_2])$. Aplicând morfismul t obținem $t(u(r_1)) \downarrow_{\Gamma} t(u(c[r_2]))$, deci

$$h(r_1) \downarrow_{\Gamma} h(c[r_2]).$$

Prin urmare există w astfel încât

$$h_1(r_1) \xrightarrow{*}_{\Gamma} w \text{ și } h(c[r_2]) \xrightarrow{*}_{\Gamma} w.$$

Din prima rescriere deducem $a_1 = c_1[h_1(r_1)] \xrightarrow{*}_{\Gamma} c_1[w]$.

Utilizând a doua rescriere deducem

$$\begin{aligned} a_2 &= c_2[h_2(r_2)] = \{(z_1 \leftarrow h_1^{z_2}(c))(c_1)\}[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))((z_1 \leftarrow h_1^{z_2}(c))(c_1)) \text{ deoarece } z_2 \text{ nu apare } c_1 \\ &= (z_1 \leftarrow \{z_2 \leftarrow h_2(r_2)(h_1^{z_2}(c))\})(c_1) = (z_1 \leftarrow \{h_1^{z_2}(c)[h_2(r_2)]\})(c_1) = (z_1 \leftarrow h(c[r_2]))(c_1) = \\ &= c_1[h(c[r_2])] \xrightarrow{*}_{\Gamma} c_1[w], \end{aligned}$$

deci $a_1 \downarrow_{\Gamma} a_2$.

Cazul 3(cu suprapunere necritică, adică $h_2(l_2)$ este subexpresie în imaginea prin h_1 a unei variabile x din l_1).

Notăm cu $c \in T_{\Sigma}(A \cup \{z_2\})$ contextul în care se găsește $h_2(l_2)$ 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 $u(x) = x$ și care acționează 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

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

deci

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

Pentru a continua analizăm contextul c_2 și observăm că în acesta apar subarbori de două tipuri provenind din variabila x din l_1

1) Apariția lui x care crează subtermenul $h_2(l_2)$ care este rescris pentru a-l obține pe a_2 și care în c_2 este înlocuită cu subarborii c

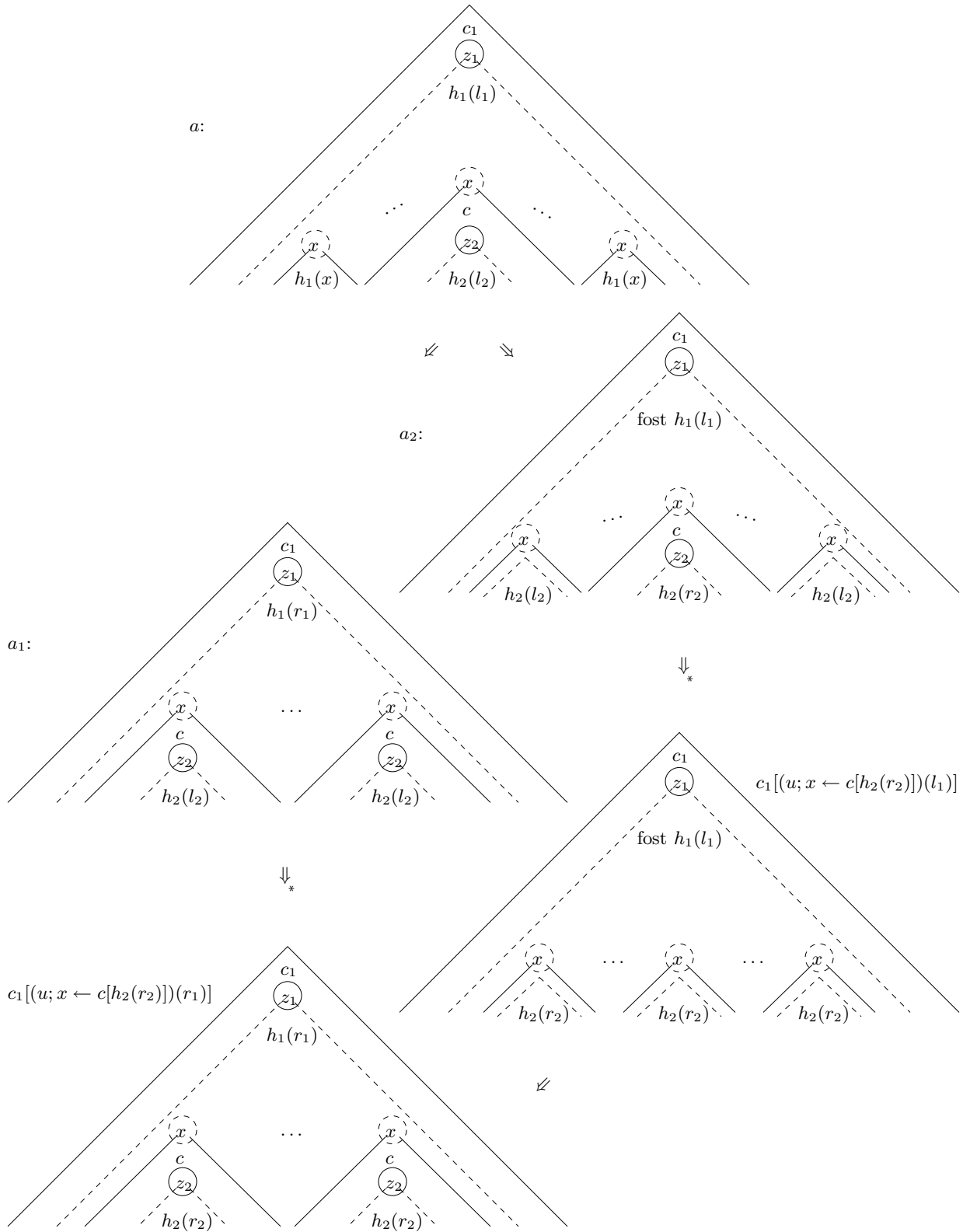


Figure 2.4: Cazul 3. Cu suprapunere necritică

2) Restul aparițiilor lui x din l_1 care în c_2 dau naștere unui subarbore $h_1(x)$. Pentru a diferenția aceste apariții ale lui x în l procedăm după cum descriem în cele de mai jos.

Punând în evidență în l_1 apariția lui x care prin aplicarea lui h_1 dă naștere subtermenului $h_2(l_2)$ care este

rescris pentru a da naștere lui a_2 și substituind-o cu o variabilă nouă z obținem 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 proprietățile $w(x_1) = h_1(x_1)$ pentru orice $x_1 \in X_1$ și $w(z) = c$. 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 $x_1 \in X_1 - \{x\}$, $p(x) = x$ și $p(z) = c[h_2(r_2)]$. Observăm că

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

$$\begin{aligned} a_2 &= c_2[h_2(r_2)] = (z_2 \leftarrow h_2(r_2))(c_1[w(l)]) = c_1[(w; z_2 \leftarrow h_2(r_2))(l)] = c_1[(p; x \leftarrow h_1(x))(l)] \\ &= c_1[(z_2 \leftarrow h_2(l_2))((p; x \leftarrow c)(l))] = (z_2 \leftarrow h_2(l_2))(c_1[(p; x \leftarrow c)(l)]) = (c_1[(p; x \leftarrow c)(l)])(h_2(l_2)). \end{aligned}$$

Prin urmare

$$a_2 \xRightarrow{*}_\Gamma (c_1[(p; x \leftarrow c)(l)])(h_2(r_2)) = (z_2 \leftarrow h_2(r_2))(c_1[(p; x \leftarrow c)(l)]) = c_1[(p; x \leftarrow c; z_2 \leftarrow h_2(r_2))(l)].$$

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

$$a_2 \xRightarrow{*}_\Gamma c_1[(z \leftarrow x; u; x \leftarrow c[h_2(r_2)])(l)] = c_1[(u; x \leftarrow c[h_2(r_2)])(l_1)] \xRightarrow{*}_\Gamma c_1[(u; x \leftarrow c[h_2(r_2)])(r_1)]$$

deci

$$a_1 \downarrow_\Gamma a_2. \square$$

2.11 TERMINAREA și PROCEDURA KNUTH-BENDIX

2.11.1 Terminarea programelor

Terminarea este o proprietate impusă de practica. Din păcate nu dispunem încă de metode puternice pentru a proba această proprietate. Dăm în cele ce urmează câteva fapte care ne permit uneori să probăm terminarea.

Reamintim că prin însăși definiția rescrierii într-un pas, $a \Rightarrow_\Gamma d$ implică $a \neq d$. Rescrierea $\xRightarrow{*}_\Gamma$ are proprietatea de terminare dacă nu există șiruri $\{a_n\}_{n \in \mathbb{N}}$ cu proprietatea $a_n \Rightarrow_\Gamma a_{n+1}$ pentru orice $n \in \mathbb{N}$.

Teorema 2.11.1 *Următoarele proprietăți sunt echivalente*

1. Rescrierea are proprietatea de terminare,
2. închiderea tranzitivă a rescrierii într-un pas $\xRightarrow{+}_\Gamma$ este o relație de ordine strictă și bine fondată,
3. există o relație de ordine \geq pe A compatibilă cu structura algebrică cu proprietățile:

(a) varianta ei strictă $>$ este bine fondată,

(b) pentru orice $(\forall X)l =_s r$ if $H \in \Gamma$ și orice $h : T_\Sigma(X) \rightarrow \mathcal{A}$ dacă $h(H) \subseteq \downarrow_\Gamma$, atunci $h(l) \geq h(r)$

4. există o relație bine fondată \vdash pe o mulțime M și o funcție $\rho : A \longrightarrow M$ cu proprietățile

- (a) pentru orice $(\forall X)l =_s r$ **if** $H \in \Gamma$ și orice $h : T_\Sigma(X) \rightarrow \mathcal{A}$ dacă $h(H) \subseteq \downarrow_\Gamma$ și $h(l) \neq h(r)$, atunci $\rho(h(l)) \vdash \rho(h(r))$
- (b) pentru orice $n \geq 1$ și orice $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$, $\rho(a) \vdash \rho(d)$ și $A_\sigma(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n) \neq A_\sigma(a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n)$ implică $\rho(A_\sigma(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n)) \vdash \rho(A_\sigma(a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n))$,

5. există o relație bine fondată \vdash pe o mulțime M și $\rho : A \longrightarrow M$ o funcție cu proprietățile

- (a) pentru orice $(\forall X)l =_s r$ **if** $H \in \Gamma$ și orice $h : T_\Sigma(X) \rightarrow \mathcal{A}$ dacă $h(H) \subseteq \downarrow_\Gamma$ și $h(l) \neq h(r)$, atunci $\rho(h(l)) \vdash \rho(h(r))$,
- (b) pentru orice context $c \in \mathcal{A}[z]_{s'}$ și pentru orice $a, d \in A_s$ dacă $\rho(a) \vdash \rho(d)$ și $c[a] \neq c[d]$, atunci $\rho(c[a]) \vdash \rho(c[d])$,

6. există o relație bine fondată \vdash pe mulțimea M și $\rho : A \longrightarrow M$ o funcție cu proprietatea

- (a) $a \Rightarrow_\Gamma d$ implică $\rho(a) \vdash \rho(d)$.

Demonstrație:

1 \longrightarrow 2. E suficient să probăm că $a \not\Rightarrow_\Gamma a$ este falsă pentru orice $a \in A$. Presupunând prin absurd existența lui $a \in A$ cu proprietatea $a \Rightarrow_\Gamma a$ se poate construi un șir care contrazice terminarea.

2 \longrightarrow 3. Rescrierea \Rightarrow_Γ^* verifică condițiile cerute.

3 \longrightarrow 4. Pentru $M = A$ și ρ aplicația identică, observăm că relația $>$ verifică condițiile cerute.

4 \longrightarrow 5. Probăm (b) prin inducție structurală după contextul c . Observăm că elementele algebrei \mathcal{A} nu sunt contexte.

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

Dacă $c = \sigma(a_1, a_2, \dots, a_n)$ unde $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s'}$. Deoarece c este context rezultă că $n \geq 1$. În plus există un $i \in [n]$ astfel încât a_i este context și restul argumentelor sunt din A . Presupunem $c[a] \neq c[d]$. Deoarece $c[a] = A_\sigma(a_1, \dots, a_{i-1}, a_i[a], a_{i+1}, \dots, a_n)$ și $c[d] = A_\sigma(a_1, \dots, a_{i-1}, a_i[d], a_{i+1}, \dots, a_n)$ rezultă că $a_i[a] \neq a_i[d]$. Din ipoteza de inducție deducem că $\rho(a_i[a]) \vdash \rho(a_i[d])$.

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

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

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

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 $c[h_s(l)] \neq c[h_s(r)]$ și 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 \longrightarrow 1. Presupunem prin absurd că \Rightarrow_Γ^* nu are proprietatea de terminare. Prin urmare există șirul $\{a_n\}_n$ cu proprietatea $a_n \Rightarrow_\Gamma a_{n+1}$ pentru orice n natural. Aplicând funcția ρ elementelor acestui șir obținem un alt șir care contrazice bine fondarea relației \vdash . \square

Se poate observa diferența esențială 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} și relația 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 relație bine fondată vom folosi relația “strict mai mare” uzuală $>$ pe mulțimea numerelor naturale. Algebra pentru care facem demonstrația este presupusă a fi o algebră liberă. Această ipoteză permite definirea funcției ρ este astfel:

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

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

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

Verificăm prima ipoteză a echivalențelor 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 $x = \rho(h(A)) - 2$ și $y = \rho(h(B)) - 2$ 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 ușor.
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 echivalența 4 este imediată.

2.11.3 Completarea Knuth-Bendix

Este vorba de o procedură care ne permite în anumite condiții destul de restrictive să găsim un program pentru o prezentare ecuațională dată. Condițiile sunt restrictive deoarece se presupune existența unei relații care să verifice condiția (b) din condițiile echivalente 4 sau 5 din teorema 2.11.1. În aceste condiții atenția se va concentra asupra veridicității condiției (a) din aceleași condiții echivalente.

Un exemplu

Vom începe cu un exemplu care va arăta cum se obține programul pentru teoria grupurilor din specificația uzuală a grupurilor.

Specificația uzuală de la care plecăm este inclusă în programul de mai sus.

$$e * A = A \quad (2.2)$$

$$A' * A = e \quad (2.3)$$

$$(A * B) * C = A * (B * C) \quad (2.4)$$

În acest exemplu ne bazăm mai mult pe intuiție, încercând să realizăm întâlnirea prin rescrierea perechilor critice. Căutăm cel mai general unificator u pentru membrul stâng l_2 al unei ecuații E_2 și un subtermen d al unui membru stâng l_1 al unei ecuații E_1 , posibil aceeași. Prin urmare există un context c astfel încât $l_1 = c[d]$. Apoi rescriem $u(l_1)$ la două forme normale n_1 și n_2 începând cu

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

după care adăugăm una dintre ecuațiile $n_1 = n_2$ sau $n_2 = n_1$ cu intenția de a asigura local confluența. Vom specula existența funcției ρ și a relației $>$ pe numere naturale din exemplul dat în secțiunea 2.11.2, pentru a alege între cele două egalități și anume dacă $\rho(n_1) > \rho(n_2)$ alegem $n_1 \stackrel{\circ}{=} n_2$ pentru a o adăuga celorlalte reguli.

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

$$\begin{aligned} (A' * A) * C &\xrightarrow{2.3} e * C \xrightarrow{2.2} C \\ (A' * A) * C &\xrightarrow{2.4} A' * (A * C) \end{aligned}$$

prin urmare adăugăm regula

$$A' * (A * C) = C \quad (2.5)$$

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

$$\begin{aligned} e' * (e * A) &\xrightarrow{2.5} A \\ e' * (e * A) &\xrightarrow{2.2} e' * A \end{aligned}$$

prin urmare adăugăm regula

$$e' * A = A \quad (2.6)$$

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) și calculând formele normale obținem

$$\begin{aligned} B'' * (B' * B) &\xrightarrow{2.3} B'' * e \\ B'' * (B' * B) &\xrightarrow{2.5} B \end{aligned}$$

prin urmare adăugăm regula

$$B'' * e = B \quad (2.7)$$

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) și calculând formele normale obținem

$$\begin{aligned} (B'' * e) * A &\xrightarrow{2.4} B'' * (e * A) \xrightarrow{2.2} B'' * A \\ (B'' * e) * A &\xrightarrow{2.7} B * A \end{aligned}$$

prin urmare adăugăm regula

$$B'' * A = B * A \quad (2.8)$$

Mai târziu vom observa că această regulă este superfluă.

Unificând membrul stâng al lui (2.7) cu membrul stâng al lui (2.8) și calculând formele normale obținem

$$\begin{aligned} B'' * e &\xrightarrow{2.7} B \\ B'' * e &\xrightarrow{2.8} B * e \end{aligned}$$

prin urmare adăugăm regula

$$B * e = B \quad (2.9)$$

Unificând membrul stâng al lui (2.7) cu membrul stâng al lui (2.9) și calculând formele normale obținem

$$\begin{aligned} B'' * e &\xrightarrow{2.7} B \\ B'' * e &\xrightarrow{2.9} B'' \end{aligned}$$

prin urmare adăugăm regula

$$B'' = B \quad (2.10)$$

Mărind lista regulilor cu noi reguli de rescriere, este posibil ca unele reguli să devină inutile, fiind posibilă înlăturarea lor fără a pierde din forța 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 și l_2 pot fi rescrise în aceeași formă normală, atunci vom elimina regula $l_1 \stackrel{\circ}{=} l_2$.

În acest moment regulile (2.7) și (2.8) pot fi eliminate deoarece

$$\begin{aligned} B'' * e &\xrightarrow{2.10} B * e \xrightarrow{2.9} B \quad \text{și} \\ B'' * A &\xrightarrow{2.10} B * A. \end{aligned}$$

Lucrăm în continuare cu regulile (2.2, 2.3, 2.4, 2.5, 2.6, 2.9 și 2.10).

Unificând membrul stâng al lui (2.6) cu membrul stâng al lui (2.9) și calculând formele normale obținem

$$\begin{aligned} e' * e &\xrightarrow{2.6} e \\ e' * e &\xrightarrow{2.9} e' \end{aligned}$$

prin urmare adăugăm regula

$$e' = e \quad (2.11)$$

În acest moment regula (2.6) poate fi eliminată deoarece

$$e' * A \xrightarrow{2.11} e * A \xrightarrow{2.2} A$$

Lucrăm în continuare cu regulile (2.2, 2.3, 2.4, 2.5, 2.9, 2.10 și 2.11).

Unificând membrul stâng al lui (2.10) cu subtermenul A' din membrul stâng al lui (2.3) și calculând formele normale obținem

$$\begin{aligned} B'' * B' &\xrightarrow{2.3} e \\ B'' * B' &\xrightarrow{2.10} B * B' \end{aligned}$$

prin urmare adăugăm regula

$$B * B' = e \quad (2.12)$$

Unificând membrul stâng al lui (2.12) cu subtermenul $A * B$ din membrul stâng al lui (2.4) și calculând formele normale obținem

$$\begin{aligned} (A * A') * C &\xrightarrow{2.4} A * (A' * C) \\ (A * A') * C &\xrightarrow{2.12} e * C \xrightarrow{2.2} C \end{aligned}$$

prin urmare adăugăm regula

$$A * (A' * C) = C \quad (2.13)$$

Unificând membrul stâng al lui (2.4) cu membrul stâng al lui (2.12) și calculând formele normale obținem

$$\begin{aligned} (A * B) * (A * B)' &\xrightarrow{2.4} A * (B * (A * B)') \\ (A * B) * (A * B)' &\xrightarrow{2.12} e \end{aligned}$$

prin urmare adăugăm regula

$$A * (B * (A * B)') = e \quad (2.14)$$

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) și calculând formele normale obținem

$$\begin{aligned} A' * (A * (B * (A * B)')) &\xrightarrow{2.14} A' * e \xrightarrow{2.9} A' \\ A' * (A * (B * (A * B)')) &\xrightarrow{2.5} B * (A * B)' \end{aligned}$$

prin urmare adăugăm regula

$$B * (A * B)' = A' \quad (2.15)$$

Mai târziu vom observa că această regulă este superfluă.

În acest moment regula (2.14) poate fi eliminată deoarece

$$A * (B * (A * B)') \xrightarrow{2.15} A * A' \xrightarrow{2.12} e$$

Lucrăm în continuare cu regulile (2.2, 2.3, 2.4, 2.5, 2.9, 2.10, 2.11, 2.12, 2.13 și 2.15).

Unificând membrul stâng al lui (2.15) cu subtermenul $A * C$ din membrul stâng al lui (2.5) și calculând formele normale obținem

$$\begin{aligned} B' * (B * (A * B)') &\xrightarrow{2.15} B' * A' \\ B' * (B * (A * B)') &\xrightarrow{2.5} (A * B)' \end{aligned}$$

prin urmare adăugăm regula

$$(A * B)' = B' * A' \quad (2.16)$$

În acest moment regula (2.15) poate fi eliminată deoarece

$$B * (A * B)' \xrightarrow{2.16} B * (B' * A') \xrightarrow{2.13} A'$$

Regulile rămase (2.2, 2.3, 2.4, 2.5, 2.9, 2.10, 2.11, 2.12, 2.13 și 2.16) coincid cu programul prezentat în secțiunea 2.11.2.

Despre problema punerii celor două expresii în membrul stâng, respectiv drept, al noilor ecuații 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 secțiunea 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 cunoaștem.

Singura motivație corectă este că alegerea este făcută pentru ca regula creată să verifice condiția necesară pentru demonstrarea terminării, fapt dat de funcția ρ și relația $>$ pentru numere naturale.

Procedura Knuth-Bendix, varianta redusă

Toate ecuațiile utilizate în aceasta secțiune sunt necondiționate.

Procedura Knuth-Bendix se aplică unei specificații ecuaționale E . Vom prezenta la început, din motive didactice, o variantă incompletă.

Procedura lucrează cu o mulțime de ecuații și o mulțime de reguli. Inițial $E_0 = E$ și $\Gamma_0 = \emptyset$.

La fiecare pas se alege o ecuație care la terminarea pasului se va elimina din mulțimea ecuațiilor. Fiecare termen al ecuației se aduce la o formă normală folosind mulțimea regulilor. Vom considera două cazuri.

A) Dacă cele două forme normale coincid pasul constă doar în eliminarea ecuației.

B) Dacă cele două forme normale sunt diferite, ele vor alcătui o nouă regulă care se adaugă mulțimii regulilor. Din noua regulă și din noua mulțime de reguli se calculează toate perechile critice posibile care se adaugă la mulțimea ecuațiilor.

Dacă procedura se termină prin epuizarea mulțimii ecuațiilor, mulțimea finală de reguli este local confluentă și o specificație echivalentă cu cea inițială.

Observația 2.11.2 Dacă $\Gamma_1 \subseteq \Gamma$, atunci $\equiv_{\Gamma_1}^A \subseteq \equiv_{\Gamma}^A$ pentru orice algebră \mathcal{A} .

Demonstrație: Fie $a \equiv_{\Gamma_1}^A b$.

Fie $h : \mathcal{A} \rightarrow \mathcal{B} \models \Gamma$. Deoarece $\mathcal{B} \models \Gamma_1$ deducem $h(a) = h(b)$. Deci $a \equiv_{\Gamma}^A b$. \square

Pentru două mulțimi de ecuații necondiționate vom folosi notația

$$\Gamma_1 \subseteq \equiv_{\Gamma}$$

cu semnificația

$$u \equiv_{\Gamma}^{T_{\Sigma}(X)} v \text{ pentru orice } (\forall X)u = v \in \Gamma_1$$

Observația 2.11.3 $\Gamma \subseteq \equiv_{\Gamma}$

Demonstrație: Fie $(\forall X)l \overset{\circ}{=}_s r \in \Gamma$. Fie $h : T_{\Sigma}(X) \rightarrow \mathcal{A} \models \Gamma$. Deoarece $\mathcal{A} \models (\forall X)l \overset{\circ}{=}_s r$ deducem $h_s(l) = h_s(r)$. Deci $l \equiv_{\Gamma}^{T_{\Sigma}(X)} r$.

Propoziție 2.11.4 Dacă $\Gamma_1 \subseteq \equiv_{\Gamma}$, atunci

- 1) Orice Γ -algebră este Γ_1 -algebră și
- 2) $\equiv_{\Gamma_1}^A \subseteq \equiv_{\Gamma}^A$.

Demonstrație: 1) Presupunem $\mathcal{A} \models \Gamma$.

Fie $(\forall X)u \overset{\circ}{=} v \in \Gamma_1$ și $h : T_{\Sigma}(X) \rightarrow \mathcal{A}$. Deoarece $u \equiv_{\Gamma}^{T_{\Sigma}(X)} v$ și $h : T_{\Sigma}(X) \rightarrow \mathcal{A} \models \Gamma$ deducem $h(u) = h(v)$. Deci $\mathcal{A} \models \Gamma_1$.

2) Presupunem în Σ -algebra \mathcal{A} că $a \equiv_{\Gamma_1}^A d$ și probăm că $a \equiv_{\Gamma}^A d$. Fie $h : \mathcal{A} \rightarrow \mathcal{M} \models \Gamma$. Prin urmare $h : \mathcal{A} \rightarrow \mathcal{M} \models \Gamma_1$, deci din $a \equiv_{\Gamma_1}^A d$ deducem $h(a) = h(d)$. \square

Menționăm utilitatea concluziei a doua. Pentru a demonstra egalitatea $\equiv_{\Gamma} = \equiv_{\Gamma_1}$ este suficient să probăm incluziunile $\Gamma \subseteq \equiv_{\Gamma_1}$ și $\Gamma_1 \subseteq \equiv_{\Gamma}$.

Observația 2.11.5 Dacă (w, v) este o pereche critică pentru Γ , atunci $w \equiv_{\Gamma} v$.

Demonstrație: Presupunem $w = u(r_1)$ și $v = u(c[r_2])$ unde $(l_1 = c[d], r_1)$ și (l_2, r_2) sunt reguli cu mulțimi de variabile disjuncte din Γ și $u = \mathbf{CGU}(d, l_2)$. Deoarece $u(l_1) \Rightarrow_{\Gamma} w$ și $u(l_1) \Rightarrow_{\Gamma} v$ din corectitudinea relației $\xRightarrow{*}_{\Gamma}$ deducem $w \equiv_{\Gamma} v$. \square

Propoziție 2.11.6 Echivalența semantică asociată reuniunii mulțimii ecuațiilor cu mulțimea regulilor este un invariant al procedurii.

Demonstrație: Să presupunem că la începutul pasului mulțimea de ecuații este $E \cup \{(\forall X)l = r\}$ cu ecuația aleasă $(\forall X)l = r$ și mulțimea de reguli Γ . Fie l' forma normală a lui l și r' forma normală a lui r .

În cazul A) $l \downarrow_{\Gamma} r$ implică $l \downarrow_{E \cup \Gamma} r$, prin urmare din corectitudinea relației $\downarrow_{E \cup \Gamma}$ deducem $l \equiv_{E \cup \Gamma} r$. Prin urmare $E \cup \{(\forall X)l = r\} \cup \Gamma \subseteq \equiv_{E \cup \Gamma}$ deci din propoziția 2.11.4 deducem $\equiv_{E \cup \{(\forall X)l=r\} \cup \Gamma} \subseteq \equiv_{E \cup \Gamma}$. Incluziunea contrară fiind evidentă obținem concluzia.

Trecem la cazul B). Fie P mulțimea perechilor critice dintre $l' \stackrel{\circ}{=} r'$ și $\Gamma \cup \{l' \stackrel{\circ}{=} r'\}$. Din observația 2.11.5 deducem $P \subseteq \equiv_{\Gamma \cup \{l' \stackrel{\circ}{=} r'\}}$.

Din $l \xRightarrow{*}_{\Gamma} l'$ și $r \xRightarrow{*}_{\Gamma} r'$ deducem $l \equiv_{\Gamma} l'$ și $r \equiv_{\Gamma} r'$. Folosind $l \equiv_{\Gamma \cup (\forall X)l \stackrel{\circ}{=} r} r$ și $l' \equiv_{\Gamma \cup (\forall X)l' \stackrel{\circ}{=} r'} r'$ deducem $l \equiv_{\Gamma \cup (\forall X)l' \stackrel{\circ}{=} r'} r$ și $l' \equiv_{\Gamma \cup (\forall X)l \stackrel{\circ}{=} r} r'$, prin urmare

$$l \equiv_{E \cup P \cup \Gamma \cup (\forall X)l' \stackrel{\circ}{=} r'} r \text{ și } l' \equiv_{E \cup \Gamma \cup (\forall X)l \stackrel{\circ}{=} r} r'.$$

Din $l' \equiv_{\Gamma \cup (\forall X)l \stackrel{\circ}{=} r} r'$ deducem $\Gamma \cup (\forall X)l' \stackrel{\circ}{=} r' \subseteq \equiv_{\Gamma \cup (\forall X)l \stackrel{\circ}{=} r}$, așadar $\equiv_{\Gamma \cup (\forall X)l' \stackrel{\circ}{=} r'} \subseteq \equiv_{\Gamma \cup (\forall X)l \stackrel{\circ}{=} r}$, deci

$$P \subseteq \equiv_{E \cup \Gamma \cup (\forall X)l \stackrel{\circ}{=} r}.$$

Din cele de mai sus deducem

$$E \cup P \cup \Gamma \cup (\forall X)l' \stackrel{\circ}{=} r' \subseteq \equiv_{E \cup \Gamma \cup (\forall X)l \stackrel{\circ}{=} r} \text{ și }$$

$$E \cup \Gamma \cup (\forall X)l \stackrel{\circ}{=} r \subseteq \equiv_{E \cup P \cup \Gamma \cup (\forall X)l' \stackrel{\circ}{=} r'},$$

deci

$$\equiv_{E \cup \Gamma \cup (\forall X)l \stackrel{\circ}{=} r} = \equiv_{E \cup P \cup \Gamma \cup (\forall X)l' \stackrel{\circ}{=} r'}. \square$$

Propoziție 2.11.7 Dacă procedura se termină, mulțimea finală de reguli generează o rescriere local confluentă.

Demonstrație:

Observăm că pe parcursul algoritmului mulțimea regulilor devine din ce în ce mai mare, fără a fi scoase reguli din această mulțime. Prin urmare orice rescriere posibilă într-un anumit moment al algoritmului rămâne posibilă până la terminarea algoritmului.

Conform teoremei din lecția precedentă este suficient să probăm că mulțimea perechilor critice este inclusă în relația de întâlnire prin rescriere.

Fie (a, b) o pereche critică pentru mulțimea finală de reguli. Din pasul B al algoritmului rezultă că perechea critică (a, b) este adăugată mulțimii ecuațiilor. Pentru ca ecuația $a \stackrel{\circ}{=} b$ să dispară din mulțimea ecuațiilor se calculează $fn(a)$ forma normală a lui a și $fn(b)$ forma normală a lui b . Dacă $fn(a) = fn(b)$ rezultă că a și b se întâlnesc prin rescriere. În caz contrar se adaugă noua regulă $fn(a) \stackrel{\circ}{=} fn(b)$ sau $fn(a) \stackrel{\circ}{=} fn(b)$ ceea ce face ca a și b să se întâlnească prin rescriere.

Pentru a obține concluzia este suficient să mai aplicăm teorema perechilor critice pentru cazul ecuațional.

\square

Procedura Knuth-Bendix

Forma completă a procedurii Knuth-Bendix se obține 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 criteriul se referă numai la algebra $T_\Sigma(X)$ a termenilor.

Criteriul A: există o relație de ordine compatibilă cu structura algebrică, a cărei variantă strictă este noetheriană.

Criteriul B: există o relație noetheriană \vdash pe o mulțime M și o funcție $\rho : A \longrightarrow M$ cu proprietatea pentru orice $n \geq 1$ și $\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}$,

$$\rho(a) \vdash \rho(d) \text{ implică } \rho(A_\sigma(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n)) \models \rho(A_\sigma(a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n)),$$

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 ECUAȚIONALĂ

3.1 IN PRIMUL RAND SEMANTICA

Reamintim câteva concepte introduse în capitolul precedent pe care le vom folosi în continuare.

Fixăm o semnătură multisortată (S, Σ) .

O *ecuație condiționată* este

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

unde X este o mulțime S -sortată de variabile, l și r sunt două elemente de sort s din $T_\Sigma(X)$ iar H o mulțime finită de egalități formale din $Sen(T_\Sigma(X)) = T_\Sigma(X) \times T_\Sigma(X)$.

Algebra \mathcal{D} *satisfacă* ecuația condiționată $(\forall X)l \stackrel{\circ}{=}_s r \text{ if } H$, fapt notat prin

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

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 mulțime Γ de ecuații condiționate, numite axiome sau clauze.

Spunem că algebra \mathcal{D} *satisfacă* Γ sau că \mathcal{D} e o Γ -algebră și scriem $\mathcal{D} \models \Gamma$ dacă \mathcal{D} *satisfacă* toate ecuațiile condiționate din Γ .

Vom lucra într-o Σ -algebră \mathcal{A} .

Congruența semantică \equiv_Γ a lui \mathcal{A} , notată și \equiv_Γ^A atunci când este pericol de confuzie, a fost definită prin

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

Fie $G \subseteq Sen(T_\Sigma(X))$ o mulțime de egalități formale din $T_\Sigma(X)$. Reamintim că

$$\Gamma \models (\forall X)G$$

dacă și numai dacă orice Γ -algebră *satisfacă* $(\forall X)G$, adică $h_s(u) = h_s(v)$ pentru orice morfism $h : T_\Sigma(X) \longrightarrow \mathcal{M} \models \Gamma$ și orice $u \stackrel{\circ}{=}_s t \in G$.

Observația 3.1.1 $\Gamma \models (\forall X)G$ *dacă și numai dacă* $G \subseteq \equiv_\Gamma^{T_\Sigma(X)}$.

Demonstrație: $\Gamma \models (\forall X)G$ 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 *satisfacă* Γ și 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 $u \stackrel{\circ}{=}_s v \in G$,
- $G \subseteq \equiv_\Gamma^{T_\Sigma(X)}$. \square

Propoziție 3.1.1 Fie $h : \mathcal{A} \rightarrow \mathcal{B}$ un Σ -morfism. Dacă $a \equiv_{\Gamma}^{\mathcal{A}} c$, atunci $h(a) \equiv_{\Gamma}^{\mathcal{B}} h(c)$.

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

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

Această propoziție arată de fapt corectitudinea regulii logicii ecuaționale globale: dacă $h : \mathcal{A} \rightarrow \mathcal{B}$ este un morfism de Σ -algebre, atunci $(\forall \mathcal{A})a \overset{\circ}{=} a'$ implică $(\forall \mathcal{B})h_s(a) \overset{\circ}{=} h_s(a')$.

Corolar 3.1.1 Dacă $\Gamma \models (\forall X)G$ și $h : T_{\Sigma}(X) \rightarrow T_{\Sigma}(Y)$ este un Σ -morfism, atunci $\Gamma \models (\forall Y)h(G)$.

Demonstrație: Din ipoteză, conform observației, deducem că $G \subseteq \equiv_{\Gamma}^{T_{\Sigma}(X)}$. Aplicând Propoziția 3.1.1, deducem $h(G) \subseteq \equiv_{\Gamma}^{T_{\Sigma}(Y)}$, deci conform observației $\Gamma \models (\forall Y)h(G)$. \square

Fie \mathcal{A}_{Γ} factorizarea lui \mathcal{A} prin congruența sa semantică \equiv_{Γ} și $\eta_{\mathcal{A}} : \mathcal{A} \rightarrow \mathcal{A}_{\Gamma}$ morfismul de factorizare. Reamintim principalele proprietăți ale acestei algebre

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

Pentru orice Γ -algebră \mathcal{B} și pentru orice morfism $h : \mathcal{A} \rightarrow \mathcal{B}$ există și este unic un morfism $h^{\#} : \mathcal{A}_{\Gamma} \rightarrow \mathcal{B}$ astfel încât $\eta_{\mathcal{A}}; h^{\#} = h$.

Dacă \mathcal{A} este Σ -algebră inițială, atunci \mathcal{A}_{Γ} este Γ -algebră inițială.

3.2 TEOREMELE LUI HERBRAND

3.2.1 Introducere

Morfismele de algebre se extind natural pentru perechi de elemente sau egalități formale și pentru mulțimi de perechi de elemente sau mulțimi de egalități formale. Dacă $h : \mathcal{A} \rightarrow \mathcal{B}$ este un Σ -morfism, atunci pentru orice $a, b \in \mathcal{A}$ prin definiție $h((a, b)) = (h(a), h(b))$ sau $h_s(a \overset{\circ}{=} b) = (h_s(a) \overset{\circ}{=} h_s(b))$. De asemenea în locul notației $h_s(a) = h_s(b)$ vom mai utiliza $h_s((a, b)) \in \Delta_B$.

Dacă G este o mulțime de egalități formale din \mathcal{A} prin definiție

$$h(G) = \{h_s(a) \overset{\circ}{=} h_s(b) : a \overset{\circ}{=} b \in G\}.$$

În plus $h(G) \subseteq \Delta_B$ înseamnă $h_s(a) = h_s(b)$ pentru orice $a \overset{\circ}{=} b \in G$.

Programarea logică ecuațională se ocupă de rezolvarea unui sistem finit de ecuații format din egalități formale în care variabilele sunt privite ca necunoscute. De aici provine și denumirea de ecuații folosită încă din capitolele precedente. De fapt rezolvarea unui astfel de sistem a fost făcută în algebra liberă din care sunt termenii ecuațiilor așa cum a fost prezentat în subcapitolul 2.6 privind unificarea.

Se caută soluții în anumite Σ -algebre \mathcal{D} . Adică pentru o astfel de algebră \mathcal{D} se caută valori date variabilelor în \mathcal{D} pentru care egalitățile formale devin egalități adevărate în \mathcal{D} . Urmează formalizarea celor de mai sus.

Notăm cu $T_{\Sigma} = T_{\Sigma}(\emptyset)$ Σ -algebra inițială (adică obiectul inițial în categoria Σ -algebrelor) și cu $T_{\Sigma, \Gamma}$ Γ -algebra inițială (adică obiectul inițial în categoria Γ -algebrelor, subcategorie a categoriei Σ -algebrelor).

Definiția 3.2.1 Fie \mathcal{A} o Σ -algebră și G o mulțime de egalități formale între elemente din $T_{\Sigma}(X)$.

$\mathcal{A} \models (\exists X)G$ dacă există morfismul $h : T_{\Sigma}(X) \rightarrow \mathcal{A}$ astfel încât $h_s(l) = h_s(r)$ pentru orice $l \overset{\circ}{=} r$.

Notăm $\Gamma \models (\exists X)G$ dacă $\mathcal{A} \models (\exists X)G$ pentru orice Γ -algebră \mathcal{A} .

Programarea logică ecuațională își pune următoarea problemă $\Gamma \models (\exists X)G$.

Teoremele lui Herbrand se referă la posibilitatea rezolvării acestor ecuații 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 mulțime de egalități formale între elemente din $T_\Sigma(X)$. Următoarele afirmații sunt echivalente:*

1. $\Gamma \models (\exists X)G$;
2. $T_{\Sigma,\Gamma} \models (\exists X)G$;
3. Există $\psi : T_\Sigma(X) \rightarrow T_\Sigma$ astfel încât $\Gamma \models (\forall \emptyset)\psi(G)$.

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

(2) \Rightarrow (3) Conform ipotezei, există morfismul $h : T_\Sigma(X) \rightarrow T_{\Sigma,\Gamma}$ astfel încât $h_s(l) = h_s(r)$ pentru orice $l \overset{\circ}{=} r \in G$. Pentru că $T_\Sigma(X)$ este algebră liberă și deci proiectivă conform propoziției 1.7.2, deoarece morfismul η_{T_Σ} este surjectiv pe componente există morfismul $\psi : T_\Sigma(X) \rightarrow T_\Sigma$ astfel încât $\psi; \eta_{T_\Sigma} = h$.

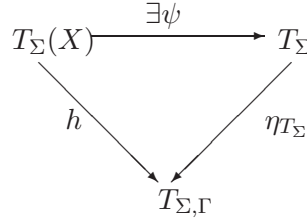


Figure 3.1: Proiectivitatea Σ -algebrei $T_\Sigma(X)$

Prin urmare $\eta_{T_\Sigma}(\psi(l)) = \eta_{T_\Sigma}(\psi(r))$ pentru orice $l \overset{\circ}{=} r \in G$. Deoarece $T_{\Sigma,\Gamma}$ se obține prin factorizarea algebrei T_Σ la congruența semantică \equiv_Γ 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 observația 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 $\alpha_\mathcal{M} : T_\Sigma \rightarrow \mathcal{M}$ unicul morfism existent între cele două algebre. Arătăm că morfismul

$$\psi; \alpha_\mathcal{M} : T_\Sigma(X) \rightarrow \mathcal{M}$$

este soluție, 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 definiția satisfacerii morfismul $\alpha_\mathcal{M} : T_\Sigma \rightarrow \mathcal{M}$ 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 $\mathcal{M} \models (\exists X)G$ și demonstrația se încheie. \square

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

A treia afirmație din teoremă ne arată că rezolvarea problemei se poate face utilizând numai algebre libere. Ea face legătura cu programarea prin rescriere și cu conceptul de soluție.

A treia afirmație mai arată că pentru existența soluției în T_Σ este necesar ca suporturile din Σ -algebra inițială T_Σ , corespunzătoare sorturilor variabilelor cuantificate existențial, să fie nevide.

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

Din a treia afirmație a teoremei lui Herbrand se mai vede că soluția $\psi : T_\Sigma(X) \rightarrow T_\Sigma$ 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 ecuația $x = f(y)$ unde evident $X = \{x, y\}$. Soluția este dată chiar de această ecuație, adică în $T_\Sigma(\{y\})$ chiar dacă suportul corespunzător lui y este vid în T_Σ . În continuare ne vor interesa și astfel de soluții, chiar dacă ecuația $x = f(y)$ are soluție numai în algebrele în care suportul corespunzător sortului lui y este nevid.

3.3 REGULILE PROGRAMĂRII LOGICE

În continuare vom considera Γ o mulțime de ecuații condiționate și G o mulțime finită de ecuații, egalități formale, din $T_\Sigma(X)$. *Problema programării logice* este $\Gamma \models (\exists X)G$.

3.3.1 Soluții

Revenim asupra conceptului de soluție deoarece în cele ce urmează vom utiliza o variantă puțin modificată, fapt datorat contextului în care suntem: soluțiile 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 existența morfismului f care face diagrama comutativă rezultă din proiectivitatea Σ -algebrei libere $T_\Sigma(X)$.

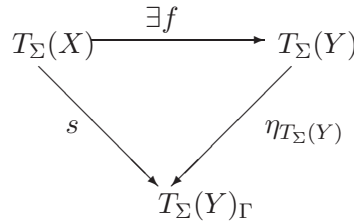


Figure 3.2: Folosind proiectivitatea Σ -algebrei $T_\Sigma(X)$

Propoziție 3.3.1 *Cu notațiile din figura 3.2 următoarele afirmații sunt echivalente*

1. $s_t(u) = s_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in G$
2. $f_t(u) \equiv_\Gamma f_t(v)$ pentru orice $u \stackrel{\circ}{=}_t v \in G$

Demonstrație: Egalitatea $s_t(u) = s_t(v)$ este, datorită comutativității 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 egale de surjecția naturală $\eta_{T_\Sigma(Y)}$ dacă și numai dacă sunt echivalente semantic. \square

Prima condiție spune că s este soluție pentru $(\exists X)G$ în $T_\Sigma(Y)_\Gamma$. Vom prefera să folosim pentru conceptul de soluție a doua condiție din propoziția de mai sus, fapt ce conduce la următoarea definiție în care facem o generalizare înlocuind algebra liberă $T_\Sigma(Y)$ cu o algebră.

Definiție 3.3.1 Σ -morfismul $f : T_\Sigma(X) \longrightarrow \mathcal{A}$ se numește **soluție** pentru $(\exists X)G$ dacă $f(G) \subseteq \equiv_\Gamma^{\mathcal{A}}$.

Observația 3.3.2 Dacă Δ este o mulțime de egalități adevărate din $T_\Sigma(X)$, atunci identitatea lui $T_\Sigma(X)$ este soluție pentru $(\exists X)\Delta$.

Demonstrație: $1_{T_\Sigma(X)}(\Delta) \subseteq \Delta_{T_\Sigma(X)} \subseteq \equiv_\Gamma^{T_\Sigma(X)}$.

Observația 3.3.3 *Un Σ -morfism $f : T_\Sigma(X) \rightarrow \mathcal{A}$ este soluție **soluție** pentru $(\exists X)G$ dacă $\Gamma \models (\forall \mathcal{A})f(G)$, adică pentru orice morfism $h : \mathcal{A} \rightarrow \mathcal{M} \models_\Sigma \Gamma$, $h(f(l)) = h(f(r))$ pentru orice $l \stackrel{\circ}{=} r \in G$.*

Demonstrație: Este suficient să folosim definiția congruenței semantice. Vezi și demonstrația observației 3.1.1

Propoziție 3.3.1 *Compunerea unei soluții cu orice Σ -morfism este soluție.*

Demonstrație: Fie $s : T_\Sigma(X) \rightarrow T_\Sigma(Y)$ o soluție pentru $(\exists X)G$ și $h : T_\Sigma(Y) \rightarrow T_\Sigma(Z)$ un morfism. Vrem să arătăm că $s;h : T_\Sigma(X) \rightarrow T_\Sigma(Z)$ este soluție pentru $(\exists X)G$, adică $\Gamma \models (\forall Z)(s;h)(G)$.

Deoarece $s : T_\Sigma(X) \rightarrow T_\Sigma(Y)$ este soluție pentru $(\exists X)G$, atunci $\Gamma \models (\forall Y)s(G)$. Corolarul 3.1.1 implică $\Gamma \models (\forall Z)h(s(G))$.

Deci $\Gamma \models (\forall Z)(s;h)(G)$ și astfel $s;h$ este soluție pentru $(\exists X)G$. \square

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

În general, soluțiile sunt construite în mai multe etape, apărând în final ca o compunere de morfisme. Morfismele care apar în procesul de calcul și care, sperăm, ca în final să furnizeze o soluție sunt numite morfisme calculate.

3.3.2 Reguli de deducție

Fie \mathcal{A} o Σ -algebră și \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 numește context dacă numărul aparițiilor 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 $(\bullet \leftarrow d)(\bullet) = d$ și $(\bullet \leftarrow d)(a) = a$, pentru orice $a \in A$. Pentru orice t din $T_\Sigma(A \cup \{\bullet\})$ și $a \in A_s$, vom prefera să scriem $t[a]$, în loc de $(\bullet \leftarrow a)(t)$.

Datorită faptului că regulile programării logice lucrează pe mulțimi de egalități formale, vom defini noțiunea de context extins. Fie \mathcal{A} o Σ -algebră, $c \in T_\Sigma(A \cup \{\bullet\})_s$ un context și $v \in A_s$. O egalitate formală de forma $c \stackrel{\circ}{=} v$ sau $v \stackrel{\circ}{=} c$ se numește context extins. Un context extins $c \stackrel{\circ}{=} v$ (respectiv $v \stackrel{\circ}{=} c$) va fi notat cu C . Să observăm că

$$(c \stackrel{\circ}{=} v)[a] = (\bullet \leftarrow a)(c \stackrel{\circ}{=} v) = (c[a] \stackrel{\circ}{=} c[v]) = (c[a] \stackrel{\circ}{=} v).$$

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 $h^\bullet(\bullet) = \bullet$ și $h^\bullet(a) = h(a)$, 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 și că

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

În continuare vom prezenta regulile de deducție folosite în programarea logică. Aceste reguli ne permit să trecem de la o mulțime G de egalități formale la o altă mulțime G' de egalități formale, obținând și un morfism calculat. Aplicarea acestor reguli va înceta în momentul în care ajungem la o mulțime de egalități adevărate, în particular la mulțimea vidă. În acest moment, putem compune toate morfismele calculate găsite, în ordinea apariției lor, și astfel vom obține o soluție pentru problema inițială. Această afirmație va fi probată mai târziu în ipoteza că toate regulile de deducție utilizate sunt corecte.

Menționăm în continuare mai multe reguli de deducție utile în programarea logică.

Regula morfismului: Dacă $G \subseteq \text{Sen}(T_\Sigma(X))$ și $\theta : T_\Sigma(X) \rightarrow T_\Sigma(Y)$, atunci

$$G \rightarrow_m \theta(G),$$

cu morfismul calculat θ .

Regula reflexiei extinse: Dacă $G \subseteq \text{Sen}(T_\Sigma(X))$ și $\theta : T_\Sigma(X) \rightarrow T_\Sigma(Y)$ astfel încât $\theta_s(l) = \theta_s(r)$, atunci

$$G \cup \{l \stackrel{\circ}{=}_s r\} \rightarrow_{re} \theta(G),$$

cu morfismul calculat θ .

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

$$G \cup \{l \stackrel{\circ}{=}_s r\} \rightarrow_r \theta(G),$$

cu morfismul calculat θ .

Observăm că regula reflexiei permite eliminarea din sistem a egalităților adevărate, caz în care morfismul calculat este o identitate.

Regula pararescrierii: Fie $G \subseteq \text{Sen}(T_\Sigma(X))$, $(\forall Y)l \stackrel{\circ}{=}_s r$ if $H \in \Gamma$ și morfismul $\theta : T_\Sigma(Y) \rightarrow T_\Sigma(X)$. Dacă C este un context extins cu variabila distinsă \bullet de sort s , atunci

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

Menționăm că pentru pararescriere, morfismul calculat este morfismul identitate.

Regula paramodulației extinse: Fie $(\forall Y)l \stackrel{\circ}{=}_s r$ if $H \in \Gamma$. Considerăm X astfel încât $X \cap Y = \emptyset$, $G \subseteq \text{Sen}(T_\Sigma(X))$ și morfismul $\theta : T_\Sigma(X \cup Y) \rightarrow T_\Sigma(Z)$ 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

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

cu morfismul calculat $\theta|_X$, restricția lui θ la $T_\Sigma(X)$.

Regula paramodulației: Fie $(\forall Y)l \stackrel{\circ}{=}_s r$ if $H \in \Gamma$. Considerăm X astfel încât $X \cap Y = \emptyset$, $G \subseteq \text{Sen}(T_\Sigma(X))$ și morfismul $\theta : T_\Sigma(X \cup Y) \rightarrow T_\Sigma(Z)$ 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

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

cu morfismul calculat $\theta|_X$, restricția lui θ la $T_\Sigma(X)$.

Comentariu Dorința exprimată mai sus de a obține o soluție cât mai generală face ca regulile de deducție utilizate de semantica operațională a programării logice să fie mai restrictive. Mai precis reflexia este reflexia extinsă cu condiția suplimentară ca θ să fie cel mai general unificator pentru l și r , iar paramodulația este paramodulația extinsă în care θ este cel mai general unificator pentru l și a . Conform uzanțelor 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 și nici a regulii morfismului deoarece ele pot conduce la pierderea parțială a generalității soluției. Aceste regulile au un rol teoretic.

3.4 CORECTITUDINEA REGULILOR PROGRAMĂRII LOGICE

Definiție 3.4.1 Fie R o regulă de deducție. Să presupunem că aplicând regula R obținem $G \rightarrow_R G'$ cu morfismul calculat $\theta : T_\Sigma(X) \rightarrow T_\Sigma(Y)$. Spunem că regula R este o **regulă corectă** dacă este îndeplinită următoarea condiție: dacă s este o soluție pentru G' , atunci $\theta; s$ este soluție pentru G .

Observăm că aplicarea unei reguli corecte $G \rightarrow_R G'$ reduce problema găsirii unei soluții pentru $(\exists X)G$ la găsirea unei soluții pentru $(\exists Y)G'$.

Dacă se aplică numai reguli corecte ajungându-se, în final la o mulțime formată doar din egalități adevărate, în particular la mulțimea vidă de ecuații, atunci compunerea tuturor morfismelor calculate este o soluție a problemei inițiale. Această afirmație rezultă din faptul că morfismul identitate este soluție pentru orice mulțime de egalități adevărate, inclusiv mulțimea vidă.

În continuare vom arăta că regulile de deducție considerate în secțiunile anterioare sunt corecte.

Deoarece $\models (\forall X)l \stackrel{\circ}{=} l$, orice soluție pentru G este soluție și pentru $G \cup \{l \stackrel{\circ}{=} l\}$. Prin urmare eliminarea egalităților adevărate este o regulă corectă.

Propoziție 3.4.1 Regula morfismului este corectă.

Demonstrație: 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 soluție pentru $\theta(G)$, adică $\Gamma \models (\forall Z)s(\theta(G))$. Trebuie să arătăm că $\theta; s$ este soluție pentru G , adică $\Gamma \models (\forall Z)(\theta; s)(G)$, ceea ce este evident. \square

Propoziție 3.4.2 Regula reflexiei extinse este corectă.

Demonstrație: Observăm că regula reflexiei extinse se poate obține din regula morfismului și eliminarea egalităților adevărate. Cum regula morfismului și eliminarea egalităților adevărate sunt corecte, rezultă că și regula reflexiei extinse este corectă. \square

Corolar 3.4.1 Regula reflexiei este corectă.

Reamintim că $S : T_\Sigma(X) \rightarrow \mathcal{B}$ este soluție pentru $(\exists X)G$ dacă și numai dacă $S(G) \subseteq \equiv_\Gamma^{\mathcal{B}}$.

Propoziție 3.4.1 Regula pararescrierii este corectă.

Demonstrație: Considerăm pararescrierea

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

unde $(\forall Y)l \stackrel{\circ}{=} r$ if $H \in \Gamma$ și $\theta : T_\Sigma(Y) \rightarrow T_\Sigma(X)$ un Σ -morfism.

Fie $S : T_\Sigma(X) \rightarrow \mathcal{B}$ o soluție pentru $(\exists X)G \cup \theta(H) \cup \{C[\theta_s(r)]\}$, prin urmare

$$S(G \cup \theta(H) \cup \{C[\theta_s(r)]\}) \subseteq \equiv_\Gamma^{\mathcal{B}}. \quad (1)$$

Trebuie să arătăm că $S : T_\Sigma(X) \rightarrow \mathcal{B}$ este soluție pentru $(\exists X)(G \cup C[\theta_s(l)])$, adică

$$S(G \cup \{C[\theta_s(l)]\}) \subseteq \equiv_\Gamma^{\mathcal{B}}.$$

Deoarece $S(G) \subseteq \equiv_\Gamma^{\mathcal{B}}$ rezultă direct din (1) mai avem de arătat că $S(C[\theta_s(l)]) \in \equiv_\Gamma^{\mathcal{B}}$.

Fie $h : \mathcal{B} \rightarrow \mathcal{M} \models \Gamma$. Din (1) folosind morfismul $S; h : T_\Sigma(X) \rightarrow \mathcal{M} \models \Gamma$ deducem

$$(S; h)(G) \cup (\theta; S; h)(H) \cup \{(S; h)(C[\theta_s(r)])\} \subseteq \Delta_M.$$

Prin urmare:

$$(\theta; S; h)(H) \subseteq \Delta_M, \quad (2)$$

$$(S; h)(C[\theta_s(r)]) \in \Delta_M. \quad (3)$$

Deoarece $\mathcal{M} \models (\forall Y)l \overset{\circ}{=}_s r$ **if** H , folosind morfismul $\theta; S; h : T_\Sigma(Y) \rightarrow \mathcal{M}$ și relația (2), deducem că

$$(\theta; S; h)_s(l) = (\theta; S; h)_s(r). \quad (4)$$

Observăm că

$$\begin{aligned} h(S(C[\theta_s(l)])) &= (S; h)(C[\theta_s(l)]) = (S; h)^z(C)[(S; h)(\theta_s(l))] = (S; h)^z(C)[(\theta_s; S; h)(l)] = \\ &= (S; h)^z(C)[(\theta_s; S; h)(r)] = (S; h)^z(C)[(S; h)(\theta_s(r))] = (S; h)(C[\theta_s(r)]). \end{aligned}$$

Din (3) deducem că $h(S(C[\theta_s(l)])) \in \Delta_M$.

Deci $S(C[\theta_s(l)]) \in \equiv_{\Gamma}^{\mathcal{B}}$. \square

Propoziție 3.4.3 *Regula paramodulației extinse se poate obține din regula morfismului și regula pararescrierii.*

Demonstrație: Fie $(\forall Y)l \overset{\circ}{=}_s r$ **if** $H \in \Gamma$, $\theta : T_\Sigma(X \cup Y) \rightarrow T_\Sigma(Z)$ astfel încât $\theta_s(l) = \theta_s(a)$, unde $a \in T_\Sigma(X)_s$. Fie C un context extins și \bullet o variabilă nouă. Aplicând regula morfismului pentru morfismul θ , obținem:

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

Menționăm următoarele egalități:

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

Acum putem aplica regula pararescrierii și obținem:

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

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

Propoziție 3.4.4 *Regula paramodulației extinse este corectă.*

Demonstrație: Din Propoziția 3.4.3 știm că orice paramodulație extinsă se poate obține din regula morfismului și regula pararescrierii. Din Propozițiile 3.4.1 și 3.4.1 știm că regulile morfismului și pararescrierii sunt corecte, de unde rezulta că și regula paramodulației extinse este corectă. \square

Corolar 3.4.2 *Regula paramodulației este corectă.*

3.5 COMPLETITUDINEA PARAMODULATIEI

3.5.1 Legături între regulile de deducție

În continuare vom prezenta alte legături dintre regulile de deducție pentru programarea logică.

În primul rând, este evident că regula reflexiei și regula paramodulației sunt cazuri particulare ale regulilor reflexiei extinse și, respectiv, paramodulației 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

$$G \cup \{l \overset{\circ}{=}_s r\} \rightarrow_m \theta(G) \cup \{\theta_s(l) \overset{\circ}{=} \theta_s(r)\},$$

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

Propoziție 3.5.1 *Pararescrierea este un caz particular de paramodulație extinsă.*

Demonstrație: Considerăm pararescrierea

$$G \cup \{C[h_s(l)]\} \rightarrow_{pr} G \cup h(H) \cup \{C[h_s(r)]\},$$

unde $(\forall Y)l \stackrel{\circ}{=} r$ **if** $H \in \Gamma$ și $h : T_\Sigma(Y) \rightarrow T_\Sigma(X)$ este un Σ -morfism (presupunem $X \cap Y = \emptyset$).

Să considerăm Σ -morfismul $\theta : T_\Sigma(X \cup Y) \rightarrow T_\Sigma(X)$ 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 $t \in T_\Sigma(X)$ și $\theta(u) = h(u)$, pentru orice $u \in T_\Sigma(Y)$.

Deoarece G este o mulțime de egalități formale din $Sen(T_\Sigma(X))$, rezultă că $\theta_s(u \dot{=} v) = (\theta_s(u) \dot{=} \theta_s(v)) = (u \dot{=} v)$, pentru orice $u \dot{=} v \in G$. În concluzie, putem scrie $\theta(G) = G$.

Similar, H este o mulțime de egalități formale din $Sen(T_\Sigma(Y))$ și astfel avem $\theta_s(u \dot{=} v) = (\theta_s(u) \dot{=} \theta_s(v)) = (h_s(u) \dot{=} h_s(v))$, pentru orice $u \dot{=} v \in H$. În concluzie, putem scrie $\theta(H) = h(H)$.

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

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

Putem aplica regula paramodulației extinse pentru $(\forall Y)l \dot{=} r$ **if** $H \in \Gamma$ și $\theta : T_\Sigma(X \cup Y) \rightarrow T_\Sigma(X)$ astfel obținând:

$$\begin{aligned} G \cup \{C[h_s(l)]\} &= G \cup \{C[a]\} \rightarrow_{pe} \theta(G \cup H \cup \{C[r]\}) = \theta(G) \cup \theta(H) \cup \theta(C[r]) = \\ &= G \cup h(H) \cup \theta^\bullet(C)[\theta_s(r)] = G \cup h(H) \cup C[h_s(r)]. \end{aligned}$$

Morfismul calculat $\theta|_X$ este identitatea lui $T_\Sigma(X)$. \square

Propoziție 3.5.2 *Dacă pentru orice clauză $(\forall Y)l \stackrel{\circ}{=} r$ **if** H din Γ , orice variabilă din Y apare în l , atunci pararescrierea este un caz particular de paramodulație în care substituția calculată este o identitate.*

Demonstrație: Păstrăm notațiile și demonstrația din propoziția precedentă. Vom proba, în plus, că θ este cel mai general unificator pentru l și a .

Fie $u : T_\Sigma(X \cup Y) \rightarrow T_\Sigma(Z)$ un unificator pentru l și 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 $u|_X$ restricția 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 $x \in X$, $u|_X(\theta(x)) = u|_X(x) = u(x)$, și 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 și a deoarece $u = \theta; u|_X$. \square

Lemă 3.5.1 *Dacă $(\forall Y)t \stackrel{\circ}{=} s$ $t \in \Gamma$, atunci $G \rightarrow_p (x \leftarrow t)(G)$, unde x este o variabilă care apare în G și nu apare în t .*

Demonstrație: Alegem o apariție a lui x în G și scriem $G = G' \cup \{C[x]\}$, unde C este un context extins. Aplicând regula paramodulației pentru $(\forall Y)t \dot{=} s$ $t \in \Gamma$, $a = x$, $\theta = \mathbf{CGU}\{a, l\} = \mathbf{CGU}\{x, t\} = x \leftarrow t$, obținem:

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

Penultima egalitate este adevărată deoarece x nu apare în t . \square

Ipoteza x apare în G nu este esențială deoarece, dacă x nu apare în G , $(x \leftarrow t)(G) = G$ și prin urmare $(x \leftarrow t)(G)$ se obține din G în 0 pași.

Lemă 3.5.2 (Lema substituției) *Dacă sunt îndeplinite următoarele condiții: G este o mulțime finită, $(\forall x)x \stackrel{\circ}{=} x \in \Gamma$ pentru orice variabilă x , $(\forall x_1 \forall x_2 \dots \forall x_n)f(x_1, x_2, \dots, x_n) \stackrel{\circ}{=} f(x_1, x_2, \dots, x_n) \in \Gamma$ pentru orice simbol de operație f , atunci regula morfismului poate fi realizată prin regula paramodulației.*

Demonstrație Primele două afirmații care urmează dovedesc că axiomele lemei substituției, mai sărace decât cele ale lemei 3.5.1, sunt suficiente pentru a demonstra concluzia lemei 3.5.1:

1. **Substituția unei variabile x cu o variabilă y poate fi realizată prin regula paramodulației în prezența axiomei $(\forall y)y \stackrel{\circ}{=} y$.**

În cazul în care x apare în G și $x \neq y$ se aplica Lema 3.5.1. În rest, evident.

2. **Arătăm că substituția unei variabile cu un termen poate fi realizată prin paramodulație în prezența axiomelor $(\forall x)x \stackrel{\circ}{=} x$ și $(\forall x_1 \forall x_2 \dots \forall x_n)f(x_1, x_2, \dots, x_n) \stackrel{\circ}{=} f(x_1, x_2, \dots, x_n)$.**

Vom demonstra acest lucru prin inducție după structura termenului t .

Primul pas al inducției este chiar (1).

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

Dacă x nu apare în G , atunci nu avem nimic de demonstrat. Presupunem că x apare în G și folosind $(\forall x_1 \forall x_2 \dots \forall x_n)f(x_1, x_2, \dots, x_n) \stackrel{\circ}{=} f(x_1, x_2, \dots, x_n) \in \Gamma$, unde variabilele x_1, \dots, x_n sunt noi, și Lema 3.5.1 deducem

$$G \rightarrow_p (x \leftarrow f(x_1, x_2, \dots, x_n))(G).$$

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

Mai observăm că

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

deoarece variabilele x_1, x_2, \dots, x_n sunt noi.

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

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

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

$$G \rightarrow_p (x_1 \leftarrow z_1)(G) \rightarrow_p \dots \rightarrow_p (x_n \leftarrow z_n)(\dots (x_1 \leftarrow z_1)(G) \dots) = G'$$

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

$$G' \xrightarrow{*}_p (z_1 \leftarrow h(x_1))(G) \xrightarrow{*}_p \dots \xrightarrow{*}_p (z_n \leftarrow h(x_n))(\dots (z_1 \leftarrow h(x_1))(G) \dots) = h(G). \square$$

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

Dacă $h : T_\Sigma(\{x, y\}) \rightarrow T_\Sigma(\{x, y, z\})$, $h(x) = z$, $h(y) = x$ și $G = \{x \stackrel{\circ}{=} y\}$, 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 deducție pentru programarea logică prin Figura 3.3.

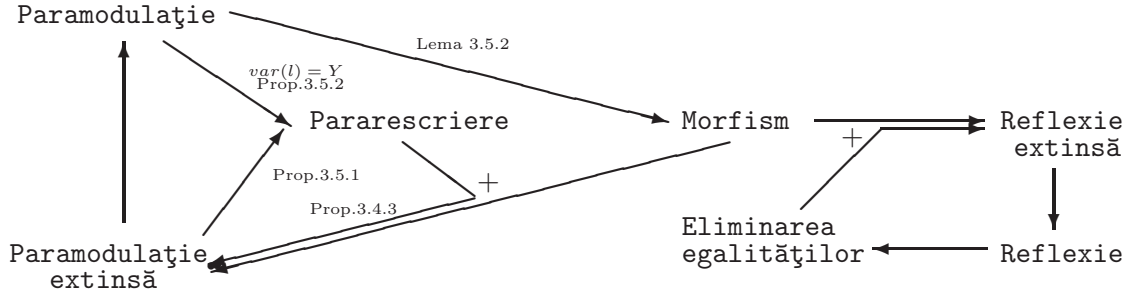


Figure 3.3: Legăturile dintre regulile de deducție

3.5.2 Amintiri despre rescriere

Reamintim câteva fapte din secțiunea 2.3.4 privind rescrierile pe care le folosim în demonstrația următoare. Lucrăm într-o Σ -algebră \mathcal{A}

O relație Q pe A se numește **închisă la contexte** dacă pentru orice context c și pentru orice pereche de elemente a, d din A , $a Q d$ implică $c[a] Q c[d]$.

Dată o relație Q pe A , relația

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

este cea mai mică relație închisă la contexte care include Q .

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

Fixăm Γ

Definim prin inducție șirul crescător de mulțimi de perechi de elemente din \mathcal{A} .

$$Q_0 = \emptyset$$

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

Prin definiție Q este reuniunea șirului crescător definit mai sus.

În cazul în care Q este definită ca mai sus, în loc de $\xrightarrow{*}_Q$ vom prefera să scriem $\xRightarrow{*}_\Gamma$.

Relația $\xRightarrow{*}_\Gamma$ este denumită Γ -rescriere sau mai scurt rescriere.

Fie \downarrow_Γ relația de întâlnire atașată lui $\xRightarrow{*}_\Gamma$. Prin definiție $a \downarrow_\Gamma b$ dacă și numai dacă există c astfel încât $a \xRightarrow{*}_\Gamma c$ și $b \xRightarrow{*}_\Gamma c$. Reamintim că întâlnirea prin rescriere \downarrow_Γ este corectă pentru \equiv_Γ , adică $\downarrow_\Gamma \subseteq \equiv_\Gamma$.

Teorema 3.5.1 *Dacă $\xRightarrow{*}_\Gamma$ este confluentă, atunci \downarrow_Γ este completă.*

3.5.3 Prolog

Reamintim că Δ înseamnă o mulțime de egalități adevărate.

Teoremă 3.5.1 *Dacă $a \downarrow_\Gamma d$, atunci $\{a \dot{=} d\} \xrightarrow{*}_{pr} \Delta$.*

Demonstrație: Presupunem $a \downarrow_\Gamma d$. Prin definiție, există v astfel încât $a \xRightarrow{*}_\Gamma v$ și $d \xRightarrow{*}_\Gamma v$. Ținând cont de definiția lui $\xRightarrow{*}_\Gamma$, putem scrie $a \xrightarrow{*}_Q v$ și $d \xrightarrow{*}_Q v$. Deoarece Q este reuniunea șirului crescător $\{Q_n\}_{n \in \mathbb{N}}$, rezultă că există un număr natural n cu proprietatea $a \xrightarrow{*}_{Q_n} v$ și $d \xrightarrow{*}_{Q_n} v$, deci $a \downarrow_{Q_n} d$.

Arătăm prin inducție după n că $\{a \dot{=} d\} \xrightarrow{*}_{pr} \Delta$.

1. Pentru $n = 0$ observăm că \rightarrow_{Q_0} este vidă și $\xrightarrow{*}_{Q_0}$ este relația de egalitate. Prin urmare $a = d$, concluzia fiind evidentă.

2. Presupunem, prin ipoteza de inducție, că pentru orice x, y dacă $x \downarrow_{Q_n} y$, atunci $\{x \stackrel{\circ}{=}_s y\} \xrightarrow{*}_{pr} \Delta$. Presupunem că $a \downarrow_{Q_{n+1}} d$.

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

În cazul contrar, presupunem, de exemplu, că $a \rightarrow_{Q_{n+1}} w$ și $w \downarrow_{Q_{n+1}} d$ cu un pas mai puțin. Din ipoteza de inducție putem scrie $\{w \stackrel{\circ}{=}_s d\} \xrightarrow{*}_{pr} \Delta$.

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

Prin urmare, deoarece $\{w \stackrel{\circ}{=}_s d\} \xrightarrow{*}_{pr} \Delta$, deducem că $\{a \stackrel{\circ}{=}_s d\} \xrightarrow{*}_{pr} h(H) \cup \Delta$. Deoarece $h(u) \downarrow_{Q_n} h(v)$, pentru orice $u \stackrel{\circ}{=} v \in H$, din ipoteza de inducție deducem $h(H) \xrightarrow{*}_{pr} \Delta$, deci $\{a \stackrel{\circ}{=}_s d\} \xrightarrow{*}_{pr} \Delta$. \square

Corolar 3.5.1 *Dacă G este o mulțime finită astfel încât $G \subseteq \downarrow_\Gamma$, atunci $G \xrightarrow{*}_{pr} \Delta$.*

3.5.4 Completitudinea

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

Presupunem că mulțimea Γ de ecuații condiționate satisface următoarele condiții: $(\forall x)x \dot{=} x \in \Gamma$, pentru orice variabilă x , $(\forall x_1 \forall x_2 \dots \forall x_n)f(x_1, x_2, \dots, x_n) \dot{=} f(x_1, x_2, \dots, x_n) \in \Gamma$, pentru orice simbol de operație f , și, pentru orice axiomă $(\forall Y)l \dot{=}_{s'} r$ **if** $H \in \Gamma$, orice variabilă din Y apare în l .

Teoremă 3.5.2 (Teorema de completitudine) *În condițiile de mai sus, dacă \downarrow_Γ este completă, atunci orice soluție poate fi obținută numai cu regula paramodulației.*

Demonstrație: Fie $\sigma : T_\Sigma(X) \rightarrow T_\Sigma(Y)$ o soluție pentru $(\exists X)G$. Prin urmare, $\sigma(G)$ este o submulțime a congruenței semantice, adică $\sigma(G) \subseteq \equiv_\Gamma$.

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

Deoarece pentru orice axiomă $(\forall Y)l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$, orice variabilă din Y apare în l , deducem, din Propoziția 3.5.2, că orice pararescriere este un caz particular de paramodulație în care substituția calculată este o identitate. În concluzie, putem scrie $\sigma(G) \xrightarrow{*}_p \Delta$ cu substituția calculată identitatea.

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

Deci $G \xrightarrow{*}_p \Delta$ cu substituția 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 $(\forall Y)l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$ o clauză. Mulțimea de variabile Y va fi disjunctă de X . Vom presupune că $T_\Sigma(X)$ și $T_\Sigma(Y)$ sunt subalgebre în $T_\Sigma(X \cup Y)$ și notăm cu

$$i_X : T_\Sigma(X) \longrightarrow T_\Sigma(X \cup Y) \text{ și } i_Y : T_\Sigma(Y) \longrightarrow T_\Sigma(X \cup Y)$$

morfismele incluziune.

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

Narrowing(Îngustare): Fie $(\forall Y)l \stackrel{\circ}{=}_s r$ **if** $H \in \Gamma$ și $\theta = \mathbf{CGU}(a, l) : T_\Sigma(X \cup Y) \longrightarrow \mathcal{B}$ unde $a \in T_\Sigma(X)$ **nu este o variabilă**. Dacă G este o mulțime de egalități formale din $T_\Sigma(X)$ și C este un context extins din $T_\Sigma(X \cup \{\bullet\})$, atunci

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

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

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

Propoziție 3.6.1 *Dacă pentru orice clauză $(\forall Y)l \stackrel{\circ}{=}_s r$ **if** H din Γ l nu este variabilă și orice variabilă din Y apare în l , atunci pararescrierea este un caz particular de îngustare în care substituția calculată este o identitate.*

Demonstrație: Este suficient să reluăm demonstrațiile propozițiilor 3.5.1 și 3.5.2 și 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 $\stackrel{*}{\Rightarrow}_\Gamma$ este relația de rescriere în A . Faptele de mai jos au fost demonstrate în secțiunea 2.4.3.

Definiția 3.6.2 Elementul $n \in A$ se numește **o formă normală** pentru dacă

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

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

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

Observația 3.6.3 *Dacă $a \stackrel{*}{\Rightarrow}_\Gamma d$, atunci $fn(a) = fn(d)$.*

Observația 3.6.4 *Axioma **FN!** implică $\stackrel{*}{\Rightarrow}_\Gamma$ este confluentă.*

Observația 3.6.5 *Funcția $fn : A \longrightarrow N$ este surjectivă și $a \downarrow_\Gamma d \Leftrightarrow fn(a) = fn(d)$.*

Pentru cazul algebrilor libere mai menționăm că orice subexpresie a unei forme normale este tot o formă normală.

3.6.3 Lema de ridicare

Definiția 3.6.6 O substituție se numește normală dacă duce orice variabilă într-un element în formă normală.

Propoziție 3.6.7 *Presupunem că mulțimile de variabile X și Y sunt disjuncte și că $T_\Sigma(X)$ și $T_\Sigma(Y)$ sunt subalgebre în $T_\Sigma(X \cup Y)$.*

Fie $l \in T_\Sigma(Y)$ astfel încât orice variabilă din Y apare în l și $a \in T_\Sigma(X)$. Fie $\rho : T_\Sigma(X \cup Y) \longrightarrow T_\Sigma(Z)$ o substituție a cărei restricție la $T_\Sigma(X)$ este normală și $\rho(a) = \rho(l)$.

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

Demonstrație: Existența lui $\mathbf{CGU}(a, l)$ rezultă din faptul că ρ este unificator pentru a și l . În plus, conform observației de la finalul secțiunii 2.6, putem să presupunem că $V \subseteq X \cup Y$ și 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 $a \in T_\Sigma(X)$ și $v \notin X$ rezultă că variabila v nu apare în a . Deoarece Y este mulțimea variabilelor care apar în l , v apare în l . Dar $\psi(v) = v$ implică apariția lui v în $\psi(l) = \psi(a)$. Deoarece variabila v nu apare în a rezultă că v a fost introdus în $\psi(a)$ prin substituția 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 formă normală, rezultă că orice subtermen al său este o formă normală. În particular $\theta(v)$ este o formă normală. \square

Dacă $\theta : X \rightarrow Z$ și $\varphi : Y \rightarrow Z$ sunt două funcții cu domeniile disjuncte notăm cu $\langle \theta, \varphi \rangle : X \cup Y \rightarrow Z$ unica funcție care pe X acționează ca θ și pe Y ca φ .

Propoziție 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 mulțime de egalități formale din $T_\Sigma(X)$.

Dacă $\theta : T_\Sigma(X) \rightarrow T_\Sigma(Z)$ este normală și

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

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

$$G \rightarrow_n R \text{ cu substituția calculată } \varphi.$$

Demonstrație: Fie $(\forall Y)l \stackrel{\circ}{=} r$ if $H \in \Gamma$ regula și $\eta : T_\Sigma(Y) \rightarrow T_\Sigma(Z)$ substituția utilizate în pararescriere. Vom presupune că variabilele din Y sunt noi, adică Y este disjunct de X .

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

$$G = G' \cup \{C[a]\}, \quad \theta(G) = \theta(G') \cup \{\theta^\bullet(C)[\eta(l)]\} \quad \text{și} \quad Q = \theta(G') \cup \eta(H) \cup \{\theta^\bullet(C)[\eta(r)]\}$$

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

Fie $\psi = \mathbf{CGU}(a, l) : T_\Sigma(X \cup Y) \rightarrow T_\Sigma(V)$ și $\theta' : T_\Sigma(V) \rightarrow T_\Sigma(Z)$ unica substituție cu proprietatea $\psi\theta' = \langle \theta, \eta \rangle$. Deoarece restricția θ a lui $\langle \theta, \eta \rangle$ la $T_\Sigma(X)$ este normală conform ipotezei, și $\langle \theta, \eta \rangle(a) = \langle \theta, \eta \rangle(l)$, aplicând propoziția 3.6.7 rezultă normalitatea lui θ' .

Notând cu $\varphi : T_\Sigma(X) \rightarrow T_\Sigma(Z)$ restricția lui ψ la $T_\Sigma(X)$ deducem că $\theta = \varphi\theta'$.

Rezultă că

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

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

$$\theta'(R) = (\psi; \theta')(G' \cup H \cup \{C[r]\}) = \langle \theta, \eta \rangle(G' \cup H \cup \{C[r]\}) = \theta(G') \cup \eta(H) \cup \{\theta^\bullet(C)[\eta(r)]\} = Q. \quad \square$$

Propoziție 3.6.9 Pentru orice $(\forall Y)l \stackrel{\circ}{=} r$ if H din Γ presupunem că orice variabilă din Y apare în l . Fie G o mulțime finită de egalități din $T_\Sigma(X)$.

Dacă $\theta : T_\Sigma(X) \rightarrow T_\Sigma(Z)$ este normală și

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

atunci

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

pentru care există o substituție normală ϵ cu proprietățile $\epsilon(G') = S$ și $\theta = \sigma\epsilon$.

$$\begin{array}{ccccc}
G & & X & \xrightarrow{\theta} & Z & & \theta(G) \\
\downarrow n & & \downarrow \varphi & & \downarrow 1_Z & & \downarrow pn \\
R & & V' & \xrightarrow{\theta'} & Z & & Q = \theta'(R) \\
\downarrow * & & \downarrow \sigma & & \downarrow 1_Z & & \downarrow * \\
G' & & V & \xrightarrow{\epsilon} & Z & & S \\
\downarrow n & & & & & & \downarrow pn
\end{array}$$

Demonstrație: Prin inducție după numărul pașilor. Vom pune în evidență prima pararescriere

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

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

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

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

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

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

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

și $(\varphi\sigma)\epsilon = \varphi\theta' = \theta$. \square

3.6.4 Epilog

Propoziție 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 substituția calculată s' pentru care există morfismul f cu proprietatea $s'; f = h$.

Demonstrație: Reamintim definiția reflexiei:

”Dacă $\theta : \mathcal{A} \longrightarrow \mathcal{B}$ este cel mai general unificator pentru l și r , atunci $G \cup \{l =_t r\} \longrightarrow_r \theta(G)$ cu morfismul calculat θ .”

Vom demonstra propoziția prin inducție după numărul elementelor mulțimii 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 și r . Atunci există un unic $v : T_\Sigma(Z) \longrightarrow T_\Sigma(Y)$ astfel încât $u; v = h$.

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

Deoarece $v(u(G')) = h(G')$ și h este unificator pentru G rezultă că v este unificator pentru $u(G')$ care are mai puține elemente decât G . Aplicând ipoteza de inducție pentru $u(G')$ deducem că $u(G') \xrightarrow{*}_r \emptyset$ cu substituția calculată w pentru care există f astfel încât $u; f = v$.

Atunci $G \xrightarrow{*}_r \emptyset$ cu substituția 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 confluența rescrierii și completitudinea relației de întâlnire prin rescriere.

Fie $s : T_{\Sigma}(X) \longrightarrow T_{\Sigma}(Z)$ o soluție pentru $(\exists X)G$. Cu ipoteza **FN!** pentru Γ soluția s se poate normaliza obținând soluția 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 inducție structurală se arată ușor 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

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

Probăm că s' este soluție pentru $(\exists X)G$ adică $s'(G) \subseteq \equiv_{\Gamma}$. Fie $u \stackrel{\circ}{=}_t v \in G$. Deoarece s este soluție pentru $(\exists X)G$ deducem $s(u) \equiv_{\Gamma} s(v)$. Deoarece $\stackrel{*}{\Rightarrow}_{\Gamma} \subseteq \equiv_{\Gamma}$ din $s(u) \stackrel{*}{\Rightarrow}_{\Gamma} s'(u)$ și $s(v) \stackrel{*}{\Rightarrow}_{\Gamma} s'(v)$ deducem $s(u) \equiv_{\Gamma} s'(u)$ și $s(v) \equiv_{\Gamma} s'(v)$. Deoarece \equiv_{Γ} este relație de echivalență obținem $s'(u) \equiv_{\Gamma} s'(v)$. Deci s' este soluție.

Propoziție 3.6.11 *Orice soluție normală se obține prin particularizarea unei soluții obținute cu narrowing și reflexie.*

Demonstrație: Fie $s' : T_{\Sigma}(X) \longrightarrow T_{\Sigma}(Z)$ o soluție normală. Utilizând completitudinea relației de întâlnire prin rescriere din $s'(G) \subseteq \equiv_{\Gamma}^{T_{\Sigma}(Z)}$ rezultă că $s'(G) \subseteq \downarrow_{\Gamma}$, prin urmare conform prologului $s'(G) \xrightarrow{*}_{pr} \Delta$. Din lema de ridicare rezultă existența substituției σ cu

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

și a substituției normale ϵ cu proprietățile $\epsilon(G') = \Delta$ și $s' = \sigma\epsilon$.

Deoarece G' este o mulțime de egalități unificabile prin ϵ , deducem conform epilogului

$$G' \xrightarrow{*}_r \emptyset \text{ cu morfismul calculat } \theta$$

și există o substituție ζ cu proprietatea $\epsilon = \theta\zeta$. Deoarece $s' = \sigma\epsilon = \sigma\theta\zeta$ deducem că orice soluție normală s' poate fi obținută prin particularizarea unei soluții $\sigma\theta$ găsite cu narrowing și reflexie.

3.7 REZOLUȚIE À LA PROLOG

Programarea logică relațională, ilustrată în viața de toate zilele de limbajul Prolog, este bazată pe rezoluție.

3.7.1 Rezoluția

Axiomele, clauze Horn, au forma $(\forall Y)\pi(v)$ **if** H unde

- 1) $\pi(v)$ este un atom, adică π este un predicat și v este un vector de termeni în concordanță cu aritatea lui π iar
- 2) H este o mulțime de atomi.

Țelul este o mulțime de atomi. Punând în evidență atomul asupra căruia va acționa rezoluția pentru o axiomă ca mai sus țelul devine $\{\pi(s)\} \cup T$. Ca mai sus presupunem că variabilele din Y sunt disjuncte de variabilele din țel.

Fie $\theta = CGU(v, s)$. Prin rezoluție, cu substituția calculată θ ajungem la țelul $\theta(H \cup T)$.

3.7.2 Rezoluție = Narrowing = Paramodulație

Trecerea de la varianta relațională la varianta ecuațională se face prin

- 1) adăugarea la semnătură a sortului b , transformarea predicatelor în simboluri de operații având rezultatul de sort b
- 2) înlocuirea fiecărui atom $\pi(v)$ cu egalitatea $\pi(v) \stackrel{\circ}{=} t$ unde t este o constantă de sort b reprezentând adevărul.

Varianta ecuațională a unei mulțimi de atomi C va fi notată cu $C^e = \{\pi(v) \stackrel{\circ}{=} t : \pi(v) \in C\}$.

Propoziție 3.7.1 *În varianta ecuațională, rezoluția se poate realiza prin narrowing și eliminarea egalităților reale.*

Demonstrație: Fie G o mulțime de atomi și $(\forall Y) \pi(v) \text{ if } H$ o clauză Horn. Considerăm $G = G' \cup \{\pi(s)\}$ și $\theta = CGU(v, s)$. Prin rezoluție obținem $\theta(G' \cup H)$.

În varianta ecuațională $G^e = G'^e \cup \{\pi(s) \stackrel{\circ}{=} t\}$ și $(\forall Y) \pi(v) \stackrel{\circ}{=} t \text{ if } H^e$.

Alegem $a = \pi(s)$, $l = \pi(v)$ și observăm că $\theta = CGU(s, v) = CGU(\pi(s), \pi(v))$.

Cum a nu este variabilă rezultă că putem aplica narrowing-ul:

$$\begin{aligned} G'^e \cup \{\pi(s) \stackrel{\circ}{=} t\} &\longrightarrow_n \theta(G'^e \cup H^e \cup \{t \stackrel{\circ}{=} t\}) = \\ &\theta((G' \cup H)^e \cup \{t \stackrel{\circ}{=} t\}) = \\ &[\theta(G' \cup H)]^e \cup \{t \stackrel{\circ}{=} t\} \end{aligned}$$

În urma eliminării egalității adevărate $t \stackrel{\circ}{=} t$ obținem varianta ecuațională a rezultatului rezoluției, $\theta(G' \cup H)$.

Corolar 3.7.1 *Fie G o mulțime de atomi. Orice soluție pentru $(\exists X)G$ obținută cu rezoluția poate fi obținută prin narrowing și eliminarea egalităților adevărate ca soluție pentru $(\exists X)G^e$*

Propoziție 3.7.2 *Fie Γ o mulțime de clauze Horn și G o mulțime de atomi. Aplicarea narrowing-ului folosind Γ^e în varianta ecuațională G^e se poate realiza prin rezoluție folosind Γ în G .*

Demonstrație: Fie $G = G' \cup \{P(s)\}$ și $(\forall Y) \pi(v) \stackrel{\circ}{=} t \text{ if } H^e$ o clauză astfel încât să se poată aplica narrowing-ul lui $P(s) \stackrel{\circ}{=} t$. Atunci $l = \pi(v)$ și 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 $P = \pi$ și $\theta = CGU(v, s)$. Prin urmare

$$G^e \longrightarrow_n \theta(G'^e \cup \{t \stackrel{\circ}{=} t\} \cup H^e) = \theta(G' \cup H)^e \cup \{t \stackrel{\circ}{=} t\}.$$

Aplicând rezoluția obținem

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

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

Corolar 3.7.2 *Fie G o mulțime de atomi. Orice soluție pentru $(\exists X)G^e$ obținută prin narrowing și eliminarea egalităților adevărate poate fi obținută cu rezoluția ca soluție pentru $(\exists X)G$.*

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

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

3.8 EXAMPLE

3.8.1 Primul

Se păstrează notațiile 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 soluție pentru următoarea interogare:

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

Rezolvare. Avem 3 variante:

- să unificăm membrul stâng din ecuația 1 cu $\text{cap}(L)$;
- să unificăm membrul stâng din ecuația 3 cu $\#(L)$;
- să unificăm membrul stâng din ecuația 4 cu $\#(L)$.

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

Identificăm cadrul de aplicare a paramodulației:

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

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

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

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

Vom unifica $\text{cap}(E \ E1 \ L2)$ cu membrul stâng al ecuației 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 ecuației 3; contextul este $z = 0$, cel mai general unificator, $L2 := \text{nil}$.

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

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

Soluția $L = 0 \ E1 \ \text{nil}$ se obține 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 soluția pentru:

1. $w \leq s\ 0$ (toate soluțiile);
2. $s\ s\ 0 \leq w$;
3. $s\ s\ 0 \leq w = \text{true}$, 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 $0 \leq s\ 0$ care este adevărat din prima clauză; soluția 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 $x \leq 0$. Cu prima clauză în care redenumim variabilele obținem $0 \leq 0$, care este adevărat cu cel mai general unificator $x := 0, x' := 0$. Soluția este $w = s\ 0$.

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

3. Programul echivalent în EQLOG se obține înlocuind $\Pi(x_1, \dots, x_n)$ cu $\Pi(x_1, \dots, x_n) = \text{true}$.

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

- contextul extins este $z = \text{true}$;
- a este $s\ s\ 0 \leq w$;
- l este $s\ x \leq s\ y$;
- cel mai general unificator pentru a și l este $w := s\ y, x := s\ 0$.

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

- contextul extins este $z = \text{true}$;
- a este $s\ 0 \leq y$;
- l este $s\ x' \leq s\ y'$;
- cel mai general unificator pentru a și l este $y := s\ y', x' := 0$.

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

- contextul extins este $z = \text{true}$;

- a este $0 \leq y'$;
- l este $0 \leq x''$;
- cel mai general unificator pentru a și l este $x'' := y'$.

Scopul devine $\{\text{true} = \text{true}, \text{true} = \text{true}, \text{true} = \text{true}\}$ și prin reflexie devine \emptyset . Soluția este $w = s \ s \ y'$.

Chapter 4

INSTITUȚII

4.1 CATEGORII

Vom folosi cuvântul **clasă** pentru a desemna colecțiile arbitrare de elemente. Orice mulțime este o clasă. Reciproca este falsă, de exemplu clasa mulțimilor nu este o mulțime.

Categoria este unul dintre cele mai abstracte concepte create de algebră. Exemplele de categorii abundă în cele mai multe dintre ramurile matematicii. Definiția categoriei este mai lungă și o vom prezenta ca pe o poveste în mai multe episoade.

Definiția 4.1.1 O **categorie** \mathcal{C} este formată din **obiecte** și **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, să-l notăm A , este numit domeniul sau sursa morfismului
- 2) Al doilea, să-l notăm B , este numit codomeniul sau cosursa morfismului.

Folosim una dintre notațiile

$$f : A \longrightarrow B \quad \text{sau} \quad A \xrightarrow{f} B.$$

Pentru orice pereche de obiecte A și B totalitatea morfismelor de sursă A și cosursă B formează o mulțime notată $\mathcal{C}(A, B)$.

Morfismele se pot compune în anumite condiții asemănătoare funcțiilor. Două morfisme f și g se pot compune numai dacă codomeniul primului coincide cu domeniul celui de al doilea. Domeniul compunerii $f;g$ coincide cu domeniul primului morfism și codomeniul compunerii coincide cu codomeniul celui de al doilea așa cum se vede mai jos

$$A \xrightarrow{\quad f \quad} B \xrightarrow{\quad g \quad} C \\ \quad \quad \quad \xrightarrow{\quad f;g \quad}$$

Compunerea este asociativă, adică oricare ar fi morfismele $f : A \longrightarrow B$, $g : B \longrightarrow C$ și $h : C \longrightarrow D$ 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

$$1_A;f = f \text{ pentru orice morfism } f \text{ de sursă } A \quad \text{și} \quad g;1_A = g \text{ pentru orice morfism } g \text{ de cosursă } A. \quad \square$$

În orice categorie morfismele identitate sunt unice.

Vom continua cu mai multe exemple.

Sunt mai multe categorii în care obiectele sunt toate mulțimile.

1. Categoria relațiilor notată **Rel** în care morfismele sunt relațiile. Compunerea relațiilor $P \in \mathbf{Rel}(A, B)$ și $Q \in \mathbf{Rel}(B, C)$, adică $P \subseteq A \times B$ și $Q \subseteq B \times C$ este prin definiție

$$P; Q = \{(a, c) \in A \times C : (\exists b)(a, b) \in P \text{ și } (b, c) \in Q\}.$$

2. Categoria funcțiilor parțiale este notată **PFn**. Prin definiție $F \in \mathbf{PFn}(A, B)$ dacă și numai dacă $F \subseteq A \times B$ și

$$(\forall a \in A)(\forall b \in B)(\forall c \in B)\{(a, b) \in F \text{ și } (a, c) \in F \text{ implică } b = c\}.$$

Observația 4.1.2 *Compunerea ca relații a două funcții parțiale este tot o funcție parțială.*

Demonstrație: Presupunem $F \in \mathbf{PFn}(A, B)$ și $G \in \mathbf{PFn}(B, C)$ și probăm că $F; G \in \mathbf{PFn}(A, C)$.

Fie $(a, c) \in F; G$ și $(a, c') \in F; G$. Din definiția compunerii există $b, b' \in B$ astfel încât $(a, b) \in F$, $(b, c) \in G$, $(a, b') \in F$ și $(b', c') \in G$. Deoarece F este funcție parțială deducem $b = b'$. Prin urmare $(b, c) \in G$ și $(b, c') \in G$ deci $c = c'$ deoarece și G este funcție parțială. În concluzie $F; G$ este funcție parțială. \square

3. Categoria mulțimilor notată **Set** are drept morfisme funcțiile, a căror compunere ca funcții coincide cu a lor compunere ca relații.

Să continuăm cu algebra.

1. Categoria grupurilor are grupurile ca obiecte și morfismele de grupuri drept morfisme.
2. Categoria grupurile abeliene sau comutative are grupurile abeliene ca obiecte și morfismele de grupuri drept morfisme.
3. Categoria inelelor are inelele ca obiecte și morfismele de inele drept morfisme.
4. Categoria Σ -algebrelor notată Alg_Σ
5. Categoria Γ -algebrelor notată Alg_Γ

Din topologie amintim categoria spațiilor topologice în care obiectele sunt spații topologice iar morfismele sunt funcțiile continue.

Încheiem cu un exemplu în care morfismele nu mai sunt funcții: categoria atașată unei mulțimi preordonate (P, \leq) . Obiectele acelei categorii sunt chiar elementele mulțimii P . Date două elemente x, y din P mulțimea morfismelor de la x la y are cel mult un element. Există un morfism de la x la y dacă și numai dacă $x \leq y$. Tranzitivitatea relației \leq ne asigură existența și unicitatea compunerii morfismelor. Reflexivitatea relației \leq ne asigură existența morfismelor identitate.

4.1.1 Morfisme distinse

Definiția 4.1.3 Un morfism $f : A \longrightarrow B$ se numește **izomorfism** dacă există un morfism $g : B \longrightarrow A$ cu proprietățile $f; g = 1_A$ și $g; f = 1_B$.

Observația 4.1.4 *Morfismul g din definiția anterioară este unic determinat.*

Demonstrație: Dacă $h : B \longrightarrow A$ are proprietățile $f; h = 1_A$ și $h; f = 1_B$, atunci

$$g = g; 1_A = g; (f; h) = (g; f); h = 1_B; h = h.$$

Datorită unicității sale morfismul g de mai sus este numit **inversul lui f** și este notat f^{-1} . \square

Propoziție 4.1.5 Dacă $f : A \longrightarrow B$ și $g : B \longrightarrow C$ sunt izomorfisme, atunci $f;g : A \longrightarrow C$ este izomorfism. În plus

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

Demonstrație: Deoarece $f : A \longrightarrow B$ este izomorfism există morfismul $f^{-1} : B \longrightarrow A$ cu proprietățile $f;f^{-1} = 1_A$ și $f^{-1};f = 1_B$. Deoarece $g : B \longrightarrow C$ este izomorfism există morfismul $g^{-1} : C \longrightarrow B$ cu proprietățile $g;g^{-1} = 1_B$ și $g^{-1};g = 1_C$.

Egalitățile

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

și

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

dovedesc că $f;g$ este izomorfism al cărui invers este $g^{-1};f^{-1}$ ceea ce probează egalitatea din enunț. \square

Definiția 4.1.6 Un morfism $f : A \longrightarrow B$ se numește **monomorfism** sau **monic** dacă oricare ar fi morfismele $g : C \longrightarrow A$ și $h : C \longrightarrow A$ egalitatea $g;f = h;f$ implică $g = h$.

Observația 4.1.7 Orice izomorfism este monomorfism.

Demonstrație: Să presupunem că $f : A \longrightarrow B$ este izomorfism, prin urmare există morfismul $f^{-1} : B \longrightarrow A$ cu proprietățile $f;f^{-1} = 1_A$ și $f^{-1};f = 1_B$.

Dacă morfismele $g : C \longrightarrow A$ și $h : C \longrightarrow A$ 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 $g = h$.

Propoziție 4.1.8 Compunerea monomorfismelor este un monomorfism.

Demonstrație: Să presupunem că morfismele $f : A \longrightarrow B$ și $g : B \longrightarrow C$ sunt monomorfisme.

Dacă morfismele $u : D \longrightarrow A$ și $v : D \longrightarrow A$ 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 și f este monomorfism.

Definiția 4.1.9 Un morfism $f : A \longrightarrow B$ se numește **epimorfism** sau **epic** dacă oricare ar fi morfismele $g : B \longrightarrow C$ și $h : B \longrightarrow C$ egalitatea $f;g = f;h$ implică $g = h$.

Observația 4.1.10 Orice izomorfism este epimorfism.

Propoziție 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.

Definiția 4.1.12 Duala \mathcal{C}^{op} are aceleași obiecte ca și \mathcal{C} . Morfismele își permută sursa și cosursa, adică prin definiție oricare ar fi obiectele A și B

$$\mathcal{C}^{op}(A, B) = \mathcal{C}(B, A).$$

Oricare ar fi morfismele $f \in \mathcal{C}^{op}(A, B)$ și $g \in \mathcal{C}^{op}(B, C)$ compunerea lor este notată $f \circ g$ și definită prin

$$f \circ g = g;f. \quad \square$$

Corectitudinea acestei definiții rezultă din următoarele:

$f : A \longrightarrow B$ și $g : B \longrightarrow C$ sunt morfismele din \mathcal{C}^{op} ;

în \mathcal{C} morfismele devin $f : B \longrightarrow A$ și $g : C \longrightarrow B$ și se pot compune obținând morfismul $g; f : C \longrightarrow A$;

care în \mathcal{C}^{op} devine un morfism de la A la C .

Se poate verifica cu ușurință că \mathcal{C}^{op} satisface condițiile de a fi o categorie.

Observația 4.1.13 *Duala categoriei \mathcal{C}^{op} este \mathcal{C} .*

Fiecare concept din teoria categoriilor are un dual. Pentru a obține conceptul dual procedăm astfel:

Scriem definiția conceptului în \mathcal{C}^{op} și apoi revenim în \mathcal{C} , obținând conceptul dual.

Dualul dualului oricărui concept este conceptul inițial.

Observația 4.1.14 *Noțiunile de monomorfism și epimorfism sunt duale una alteia.*

Demonstrație: Fie $f \in \mathcal{C}^{op}(A, B)$ un monomorfism, adică oricare ar fi morfismele $g \in \mathcal{C}^{op}(C, A)$ și $h \in \mathcal{C}^{op}(C, A)$ 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 $g \in \mathcal{C}(A, C)$ și $h \in \mathcal{C}(A, C)$ egalitatea $f; g = f; h$ implică $g = h$.

Deci dualul unui monomorfism este un epimorfism. \square

Există și concepte autoduale, adică dualul este chiar conceptul inițial. Exemple de concepte autoduale sunt morfismele identitate și izomorfismele.

Fiecare propoziție P din teoria categoriilor are o propoziție duală P^o , care se obține din P înlocuind fiecare concept cu dualul sau.

Principiul dualității. Dacă o propoziție privind teoria categoriilor este adevărată în orice categorie, atunci și duala ei este adevărată în orice categorie.

Motivație: Presupunem că propoziția P este adevărată în orice categorie și 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} . \square

Principiul dualității este util deoarece ne permite să facem mai puține demonstrații. De exemplu: propozițiile “compunerea de monomorfisme este un monomorfism” și “compunerea de epimorfisme este un epimorfism” sunt duale, prin urmare este suficient să demonstrăm numai una dintre ele deoarece a doua rezultă din prima folosind principiul dualității.

4.1.3 Subcategorii

O categorie \mathcal{D} se numește **subcategorie** a categoriei \mathcal{C} dacă

1. Orice obiect din \mathcal{D} este obiect al lui \mathcal{C} ,
2. Oricare ar fi obiectele A și 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} și
4. pentru orice obiect A din \mathcal{D} morfismul 1_A din $\mathcal{C}(A, A)$ se găsește în $\mathcal{D}(A, A)$.

PFn este subcategorie a lui **Rel**. **Set** este subcategorie a lui **PFn**. În aceste exemple obiectele subcategoriei coincid cu cele ale categoriei.

Subcategoria \mathcal{D} a categoriei \mathcal{C} se numește **plină** dacă $\mathcal{D}(A, B) = \mathcal{C}(A, B)$ oricare ar fi obiectele A și B din \mathcal{D} . O subcategorie plină este caracterizată de obiectele ei, deoarece morfismele sunt toate morfismele din categoria \mathcal{C} cu sursa și cosursa din \mathcal{D} . De exemplu subcategoria grupurilor comutative este subcategorie plină a categoriei grupurilor.

4.2 FUNCTORI

Categoriile, ca toate structurile algebrice, au și ele morfismele lor. Ele se numesc functori.

Definiția 4.2.1 Fie două categorii \mathcal{C} și \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 $f : A \longrightarrow B$ și $g : B \longrightarrow C$ din \mathcal{C} ,
 - (b) $F(1_A) = 1_{F(A)}$ pentru orice obiect A din \mathcal{C} . \square

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} și,
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.

Definiția 4.2.2 Compunerea $F; G : \mathcal{A} \longrightarrow \mathcal{C}$ a functorilor $F : \mathcal{A} \longrightarrow \mathcal{B}$ și $G : \mathcal{B} \longrightarrow \mathcal{C}$ 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} . \square

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 $f : A \longrightarrow B$ și $g : B \longrightarrow C$ 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 $F : \mathcal{A} \longrightarrow \mathcal{B}$, $G : \mathcal{B} \longrightarrow \mathcal{C}$ și $H : \mathcal{C} \longrightarrow \mathcal{D}$. 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 $1_{\mathcal{A}}; F = F$ și $F; 1_{\mathcal{B}} = F$. Prin urmare categoriile formează o categorie “mare” deoarece totalitatea functorilor dintre două categorii fixate s-ar putea să nu fie o mulțime. Vom nota această categorie mare cu \mathcal{Cat} .

Propoziție 4.2.3 Dacă $F : \mathcal{C} \longrightarrow \mathcal{D}$ este un functor și $f : A \longrightarrow B$ un izomorfism, atunci $F(f)$ este un izomorfism.

Demonstrație: Deoarece $f : A \longrightarrow B$ un izomorfism există morfismul $f^{-1} : B \longrightarrow A$ cu proprietățile $f; f^{-1} = 1_A$ și $f^{-1}; f = 1_B$. Aplicând functorul F obținem

$$F(f; f^{-1}) = F(1_A) \text{ și } F(f^{-1}; f) = F(1_B),$$

prin urmare

$$F(f); F(f^{-1}) = 1_{F(A)} \text{ și } F(f^{-1}); F(f) = 1_{F(B)},$$

deci $F(f)$ este izomorfism având inversul $F(f^{-1})$ adică

$$F(f)^{-1} = F(f^{-1}). \quad \square$$

Fie \mathcal{S} și \mathcal{C} două categorii și $M : \mathcal{S} \longrightarrow \mathcal{C}^{op}$ un functor. Dacă $f : A \longrightarrow B$ și $g : B \longrightarrow C$ 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 numește contravariant.

4.3 CONEXIUNI GALOIS

O conexiune Galois

$$(P, \leq) \xrightleftharpoons[*]{*} (Q, \leq)$$

este formată din două mulțimi parțial ordonate (P, \leq) și (Q, \leq) și două funcții $*$ (deși sunt notate la fel deosebirea se va face din context) cu următoarele proprietăți:

1. $(\forall p, p_1 \in P) p \leq p_1 \implies p_1^* \leq p^*$,
2. $(\forall q, q_1 \in Q) q \leq q_1 \implies q_1^* \leq q^*$,
3. $(\forall p \in P) p \leq p^{**}$,
4. $(\forall q \in Q) q \leq q^{**}$.

Primele două cerințe ne spun că funcțiile $*$ sunt descrescătoare.

Probăm unele consecințe ale axiomelor de mai sus

1. $*** = *$.

Din (4) rezultă $p^* \leq p^{***}$. Din (3), $p \leq p^{**}$, și cu (1) obținem $p^{***} \leq p^*$. 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) și (2) avem $q^* \leq p^*$ și $p^{**} \leq q^{**}$ deci $p^\bullet \leq q^\bullet$.

Exercițiu Înlocuiți una dintre relațiile de ordine cu duala ei și arătați că noțiunea de conexiune Galois este echivalentă cu o pereche de aplicații $F : Q \longrightarrow P$ și $G : P \longrightarrow Q$ 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 și cu cel de functori adjuncți între categoriile atașate unor mulțimi parțial ordonate.

4.3.1 Conexiunea Galois atașată unei relații

Fie A și B două mulțimi și $R \subseteq A \times B$ o relație. Relației R îi atașăm conexiunea Galois

$$(\mathcal{P}(A), \subseteq) \xrightleftharpoons[*]{*} (\mathcal{P}(B), \subseteq)$$

unde $(\mathcal{P}(X), \subseteq)$ reprezintă mulțimea părților mulțimii X ($X \in \{A, B\}$) ordonată (parțial) cu incluziunea iar funcțiile $*$ sunt definite în modul următor:

$$\begin{aligned} C^* &= \{b \in B \mid \forall c \in C, cRb\} \text{ pentru orice } C \subseteq A, \\ D^* &= \{a \in A \mid \forall d \in D, aRd\} \text{ pentru orice } D \subseteq B. \end{aligned}$$

Se verifică ușor cele patru proprietăți din definiția conexiunii Galois. Probăm numai prima și a treia condiție

Fie $C \subseteq C_1 \subseteq A$. Probăm $C_1^* \subseteq C^*$.

Fie $b \in C_1^*$. Prin definiție 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 definiția lui C^* deducem că pentru orice $b \in C^*$ are loc cRb . Deoarece c este în relația R cu orice element din C^* deducem că $c \in C^{**}$.

Propoziție 4.3.1 *Conexiunea Galois atașată unei relații satisface*

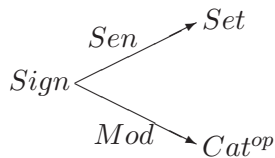
1. $(\bigcup_{i \in I} C_i)^* = \bigcap_{i \in I} C_i^*$ unde $C_i \subseteq A$ pentru orice $i \in I$,
2. $(\bigcup_{i \in I} D_i)^* = \bigcap_{i \in I} D_i^*$ unde $D_i \subseteq B$ pentru orice $i \in I$.

Demonstrație: Probăm numai prima egalitate. Pentru orice $b \in B$ remarcăm echivalența următoarelor observații

$$\begin{aligned} &b \in (\bigcup_{i \in I} C_i)^*, \\ &bRc \text{ pentru orice } c \in \bigcup_{i \in I} C_i, \\ &bRc \text{ pentru orice } i \in I \text{ și } c \in C_i, \\ &bRc \text{ pentru orice } c \in C_i \text{ și } i \in I, \\ &b \in C_i^* \text{ pentru orice } i \in I \text{ și} \\ &b \in \bigcap_{i \in I} C_i^*. \end{aligned}$$

4.4 INSTITUȚII

Se numește *instituție* (notată cu \mathcal{I}) următoarea construcție



unde $Sign$ este o categorie, Set este categoria mulțimilor, Cat^{op} este duala categoriei categoriilor iar Sen și Mod sunt doi functori. Obiectele categoriei $Sign$ se numesc *signaturi*. $Sign$ este numită categoria signaturilor. De asemenea, se mai dă câte o **relație 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 **condiție de satisfacere**:

$$\begin{aligned} \forall \varphi : \Sigma \longrightarrow \Sigma', \forall M' \in |Mod(\Sigma')|, \forall e \in Sen(\Sigma), \\ M' \models_{\Sigma'} Sen(\varphi)(e) \iff Mod(\varphi)(M') \models_{\Sigma} e. \quad \square \end{aligned}$$

Se numește Σ -model orice obiect $M \in |Mod(\Sigma)|$. Se numește Σ -propoziție orice element $e \in Sen(\Sigma)$.

Pentru orice semnătură Σ , relației \models_{Σ} îi putem asocia o conexiune Galois ca mai înainte

$$(\mathcal{P}(|Mod(\Sigma)|), \subseteq) \xrightleftharpoons[*]{*} (\mathcal{P}(Sen(\Sigma)), \subseteq).$$

Definiția 4.4.1 Spunem că Σ -modelul M satisface mulțimea de Σ -propoziții $E \subseteq Sen(\Sigma)$, fapt notat prin $M \models_{\Sigma} E$ dacă $M \models_{\Sigma} e$ pentru orice $e \in E$.

Observația 4.4.2 $M \models_{\Sigma} E$ dacă și numai dacă $M \in E^*$.

Observăm că E^* este mulțimea modelelor lui E .

Observația 4.4.3 Dacă $M \models_{\Sigma} E$ atunci $M \models_{\Sigma} E^{\bullet}$.

Demonstrație: Într-adevăr, dacă $M \models_{\Sigma} E$ atunci $M \in E^* = E^{***} = E^{\bullet\bullet}$ adică $M \models_{\Sigma} E^{\bullet}$. \square

Are loc și **condiția de satisfacere extinsă**:

Pentru orice morfism de semnături $\varphi : \Sigma \longrightarrow \Sigma'$, pentru orice $M' \in |Mod(\Sigma')|$ și pentru orice $E \subseteq Sen(\Sigma)$

$$M' \models_{\Sigma'} Sen(\varphi)(E) \iff Mod(\varphi)(M') \models_{\Sigma} E.$$

Membrul stâng este echivalent cu

$M' \models_{\Sigma'} Sen(\varphi)(e)$ pentru orice $e \in E$ care conform condiției de satisfacere este echivalent cu $Mod(\varphi)(M') \models_{\Sigma} e$ pentru orice $e \in E$, adică cu membrul drept.

4.4.1 Consecință semantică

Definiția 4.4.4 Spunem că mulțimea de Σ -propoziții E implică Σ -propoziția $e \in Sen(\Sigma)$, sau că e este o consecință semantică a lui E și notăm $E \models_{\Sigma} e$ dacă

$$(\forall M \in |Mod(\Sigma)|) M \models_{\Sigma} E \implies M \models_{\Sigma} e. \quad \square$$

Observația 4.4.5 $E \models_{\Sigma} e$ dacă și numai dacă $e \in E^{\bullet}$.

Demonstrație: Din definiție $E \models_{\Sigma} e$ este echivalent cu $(\forall M) M \in E^* \implies M \models_{\Sigma} e$, adică cu $e \in E^{**}$, deci cu $e \in E^{\bullet}$.

Mulțimea E^{\bullet} se numește **mulțimea consecințelor semantice** ale mulțimii E .

Definiția 4.4.6 Oricare ar fi mulțimile de Σ -propoziții E și E_1 spunem că E_1 este o consecință semantică a lui E și notăm $E \models_{\Sigma} E_1$ dacă

$$(\forall M \in |Mod(\Sigma)|) M \models_{\Sigma} E \implies M \models_{\Sigma} E_1. \quad \square$$

Observația 4.4.7 $E \models_{\Sigma} E_1$ dacă și numai dacă $E^* \subseteq E_1^*$, dacă și numai dacă $E_1 \subseteq E^{\bullet}$

Observația 4.4.8 Pentru orice morfism de semnături $\varphi : \Sigma \longrightarrow \Sigma'$ și pentru orice $E \subseteq \text{Sen}(\Sigma)$ avem

$$\text{Sen}(\varphi)(E)^* = \text{Mod}(\varphi)^{-1}(E^*).$$

Demonstrație: Din condiția de satisfacere extinsă, pentru orice $M' \in |\text{Mod}(\Sigma')|$,

$$M' \models_{\Sigma'} \text{Sen}(\varphi)(E) \iff \text{Mod}(\varphi)(M') \models_{\Sigma} E$$

de unde $M' \in \text{Sen}(\varphi)(E)^* \iff \text{Mod}(\varphi)(M') \in E^*$ deci

$$M' \in \text{Sen}(\varphi)(E)^* \iff M' \in \text{Mod}(\varphi)^{-1}(E^*)$$

ceea ce demonstrează egalitatea din enunț. \square

Observația 4.4.9 (Lema închiderii) Pentru orice $\varphi : \Sigma \longrightarrow \Sigma'$ morfism de semnături și orice $E \subseteq \text{Sen}(\Sigma)$ avem

$$\text{Sen}(\varphi)(E^\bullet) \subseteq \text{Sen}(\varphi)(E)^\bullet.$$

Demonstrație: Plecând de la

$$\text{Sen}(\varphi)(E^\bullet) \subseteq \text{Sen}(\varphi)(E^\bullet)^\bullet = \text{Sen}(\varphi)(E^\bullet)^{**}$$

și folosind observația anterioară

$$\text{Sen}(\varphi)(E^\bullet)^{**} = (\text{Mod}(\varphi)^{-1}(E^{\bullet*}))^* = (\text{Mod}(\varphi)^{-1}(E^*))^* = (\text{Sen}(\varphi)(E))^{**} = (\text{Sen}(\varphi)(E))^\bullet$$

obținem concluzia. \square

Observația 4.4.10 Fie $\varphi : \Sigma \longrightarrow \Sigma'$ un morfism de semnături $e \in \text{Sen}(\Sigma)$ și $E \subseteq \text{Sen}(\Sigma)$.

1. $E \models_{\Sigma} e \Rightarrow \text{Sen}(\varphi)(E) \models_{\Sigma'} \text{Sen}(\varphi)(e)$,
2. Dacă $\text{Mod}(\varphi) : \text{Mod}(\Sigma') \longrightarrow \text{Mod}(\Sigma)$ este surjectiv pe obiecte, atunci $E \models_{\Sigma} e \Leftrightarrow \text{Sen}(\varphi)(E) \models_{\Sigma'} \text{Sen}(\varphi)(e)$.

Demonstrație:

1. Semnificația lemei închiderii este aceea că pentru orice $e \in E^\bullet$, $\text{Sen}(\varphi)(e) \in \text{Sen}(\varphi)(E)^\bullet$ adică dacă $E \models_{\Sigma} e$ atunci $\text{Sen}(\varphi)(E) \models_{\Sigma'} \text{Sen}(\varphi)(e)$ ceea ce, în cuvinte, înseamnă că dacă aplicăm unei deductibilități semantice un morfism de semnături obținem tot o deductibilitate semantică.

2. Pentru demonstrarea incluziunii contrare folosim ipoteza $\text{Sen}(\varphi)(E) \models_{\Sigma'} \text{Sen}(\varphi)(e)$.

Fie $M \models_{\Sigma} E$. Deoarece $\text{Mod}(\varphi) : \text{Mod}(\Sigma') \longrightarrow \text{Mod}(\Sigma)$ este surjectiv pe obiecte există M' astfel încât $M = \text{Mod}(\varphi)(M')$, prin urmare $\text{Mod}(\varphi)(M') \models_{\Sigma} E$. Din condiția de satisfacere deducem $M' \models_{\Sigma'} \text{Sen}(\varphi)(E)$, prin urmare din ipoteză $M' \models_{\Sigma'} \text{Sen}(\varphi)(e)$. Folosind din nou condiția de satisfacere $\text{Mod}(\varphi)(M') \models_{\Sigma} e$, deci $M \models_{\Sigma} e$. \square

Propoziție 4.4.11 Dacă $\varphi : \Sigma \longrightarrow \Sigma'$ este morfism de semnături iar $\text{Mod}(\varphi) : \text{Mod}(\Sigma') \longrightarrow \text{Mod}(\Sigma)$ este surjectiv pe obiecte, atunci pentru orice $E \subseteq \text{Sen}(\Sigma)$ avem $\text{Mod}(\varphi)(\text{Sen}(\varphi)(E)^*) = E^*$.

Demonstrație: Facem observația că pentru o funcție surjectivă $f : A \longrightarrow B$, pentru orice $C \subseteq B$ are loc egalitatea $f(f^{-1}(C)) = C$. Cum din lema 4.4.8 rezultă egalitatea

$$\text{Sen}(\varphi)(E)^* = \text{Mod}(\varphi)^{-1}(E^*)$$

obținem, ținând seamă de observația anterioară, $\text{Mod}(\varphi)(\text{Sen}(\varphi)(E)^*) = \text{Mod}(\varphi)(\text{Mod}(\varphi)^{-1}(E^*)) = E^*$.

4.4.2 Exemple

Un exemplu din algebră

Vom descrie succint un exemplu simplu și neartificial de instituție. În acest exemplu X este o mulțime 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 coeficienți în R și variabile din X , adică $Sen(R) = R[X]$.

În plus orice morfism de inele $g : R \rightarrow R'$ se prelungește în mod natural la un morfism de inele

$Sen(g) : R[X] \rightarrow R'[X]$ 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 $a : R \rightarrow I$ este un morfism de inele.

Pentru orice morfism de inele $\varphi : R' \rightarrow R$ functorul

$$Mod(\varphi) : Mod(R) \rightarrow Mod(R')$$

este definit prin:

$Mod(\varphi)(a, I) = (\varphi; a, I)$ pentru orice R -algebră (a, I) și

$Mod(\varphi)(m) = m$ pentru orice morfism m de R -algebre.

Relația de satisfacere așa cum este definită mai jos reflectă ideea că într-o R -algebră polinomul are rădăcini.

Pentru orice R -algebră (a, I) și orice funcție $f : X \rightarrow I$ notăm cu $\langle a, f \rangle$ unicul morfism de inele a cărui restricție la R este a și a cărui restricție la X este f . Pentru orice R -algebră (a, I) și orice polinom $p \in R[X]$, relația de satisfacere este definită prin:

$$(a, I) \models_R p \iff (\exists s : X \rightarrow I) \langle a, s \rangle (p) = 0.$$

Pentru orice morfism de inele comutative $\varphi : R' \rightarrow R$, pentru orice R -algebră (a, I) și pentru orice funcție $s : X \rightarrow I$ observăm că

$$Sen(\varphi); \langle a, s \rangle = \langle \varphi; a, s \rangle.$$

Verificăm condiția de satisfacere. Fie $\varphi : R' \rightarrow R$ un morfism de inele comutative, (a, I) o R -algebră și $p \in R'[X]$. Observăm că afirmația

$$(a, I) \models_R Sen(\varphi)(p)$$

este succesiv echivalentă cu

$$\begin{aligned} (\exists s : X \rightarrow I) \langle a, s \rangle (Sen(\varphi)(p)) &= 0, \\ (\exists s : X \rightarrow I) \langle \varphi; a, s \rangle (p) &= 0, \\ (\varphi; a, I) &\models_{R'} p, \\ Mod(\varphi)(a, I) &\models_{R'} p \end{aligned}$$

deci condiția de satisfacere este verificată.

Instituția calculului propozițional

Fie X o mulțime de variabile propoziționale și Σ o signatură algebrică cu un singur sort, cu simbolurile de funcții

$$\neg : bool \rightarrow bool \text{ și } \wedge : bool \, bool \rightarrow bool.$$

Arătăm că Mod este functor.

Pentru compunere considerăm $X \xrightarrow{f} Y \xrightarrow{g} Z$ două funcții și 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 șirul de egalități

$$\begin{aligned} Mod(f; g)(h) &= Sen(f; g); h \\ &= Sen(f); Sen(g); h \\ &= Sen(f); Mod(g)(h) \\ &= Mod(f)(Mod(g)(h)) \\ &= (Mod(g); Mod(f))(h) \end{aligned}$$

Deci $Mod(f; g) = Mod(g); Mod(f)$.

Pentru identități 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 $Mod(1_X) = 1_{Alg_{\Sigma}(\mathcal{T}_{\Sigma}(X), \mathbf{B}_2)}$.

Relația de satisfacere

Fie X o mulțime de variabile, $e \in Sen(X)$ și $h \in Mod(X) = Alg_{\Sigma}(\mathcal{T}_{\Sigma}(X), \mathbf{B}_2)$. Intuitiv, morfismul $h : \mathcal{T}_{\Sigma}(X) \rightarrow \mathbf{B}_2$ care dă valori de adevăr variabilelor din X , satisface expresia e a calculului propozițional dacă rezultatul evaluării expresiei e pentru valorile de adevăr ale variabilelor din X este 1, adică adevărul. Definim relația de satisfacere \models_X prin

$$h \models_X e \Leftrightarrow h(e) = 1$$

Verificăm condiția de satisfacere. Fie $f : X \rightarrow Y$ o funcție, $e \in Sen(X)$, $h' \in Mod(Y)$. 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

Propoziție 4.5.1 Fie $\varphi : \Sigma \rightarrow \Sigma_1$ un morfism de semnături, $E \subseteq Sen(\Sigma)$ și $E_1 \subseteq Sen(\Sigma_1)$. Următoarele afirmații sunt echivalente:

1. $Sen(\varphi)(E) \subseteq E_1^{\bullet}$,
2. $E_1 \models_{\Sigma} Sen(\varphi)(E)$.
3. pentru orice $M_1 \in |Mod(\Sigma_1)|$, $M_1 \models_{\Sigma_1} E_1 \implies Mod(\varphi)(M_1) \models_{\Sigma} E$,

Demonstrație: Prima afirmație este echivalentă cu

$$Sen(\varphi)(E)^{\bullet} \subseteq E_1^{\bullet}$$

care conform proprietăților conexiunii Galois este echivalentă cu

$$Sen(\varphi)(E)^* \supseteq E_1^*$$

adică cu a doua afirmație și în conformitate cu observația 4.4.8, este echivalent cu

$$Mod(\varphi)^{-1}(E^*) \supseteq E_1^*.$$

Ultima afirmație 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 afirmație. \square

Definiția 4.5.2 Se numește **prezentare** o pereche (Σ, E) unde $\Sigma \in |Sign|$, și $E \subseteq Sen(\Sigma)$.

Un morfism de semnături $\varphi : \Sigma \longrightarrow \Sigma'$ se numește **morfism de prezentări** $\varphi : (\Sigma, E) \longrightarrow (\Sigma', E')$ dacă satisface condițiile echivalente ale propoziției precedente.

Observația 4.5.3 Dacă $\varphi : (\Sigma, E) \longrightarrow (\Sigma', E')$ este morfism de prezentări, atunci $\varphi : (\Sigma, E^\bullet) \longrightarrow (\Sigma', E')$ este tot morfism de prezentări.

Demonstrație: Din ipoteză rezultă că $Sen(\varphi)(E) \subseteq E'^\bullet$ de unde $Sen(\varphi)(E)^\bullet \subseteq E'^\bullet$ și, folosind lema închiderii, obținem $Sen(\varphi)(E^\bullet) \subseteq Sen(\varphi)(E)^\bullet \subseteq E'^\bullet$ ceea ce demonstrează afirmația. \square

Propoziția 4.5.4 Compunerea ca morfisme de semnături a două morfisme de prezentări este tot un morfism de prezentări.

Demonstrație: 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 lemei anterioare. \square

Observație. Din propoziția anterioară obținem că mulțimea prezentărilor unei instituții \mathcal{I} împreună cu morfismele de prezentări formează o categorie, pe care o vom nota cu $Pres(\mathcal{I})$.

Folosind propoziția 4.5.1 putem defini corespondentul lui Mod pentru categoria prezentărilor și anume

$$Mod^p : Pres(\mathcal{I}) \longrightarrow Cat^{op}.$$

Prin definiție $Mod^p(\Sigma, E)$ reprezintă subcategoria plină dată de toate modelele ce satisfac E . Pentru orice morfism de prezentări $\varphi : (\Sigma, E) \longrightarrow (\Sigma', E')$ considerăm diagrama

$$\begin{array}{ccc} Mod(\Sigma') & \xrightarrow{Mod(\varphi)} & Mod(\Sigma) \\ \uparrow & & \uparrow \\ Mod^p(\Sigma', E') & \xrightarrow{Mod^p(\varphi)} & Mod^p(\Sigma, E) \end{array}$$

unde prin definiție $Mod^p(\varphi)$ este restricția și co-restricția lui $Mod(\varphi)$.

4.5.1 Instituția prezentărilor

Definim instituția prezentărilor \mathcal{I}^p unei instituții \mathcal{I} . Categoria prezentărilor lui \mathcal{I} devine categoria semnăturilor 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) și $V(\phi) = \phi$ pentru orice morfism de prezentări ϕ . Prin definiție

$$Sen^p = V; Sen.$$

Cum functorul “modele” Mod^p a fost definit mai sus este suficient să mai definim relația de satisfacere și să demonstrăm condiția de satisfacere. Prin definiție

$$M \models_{(\Sigma, E)} e \iff M \models_{\Sigma} e$$

pentru orice (Σ, E) -model M și orice $e \in \text{Sen}(\Sigma)$.

Demonstrăm condiției de satisfacere. Fie $\varphi : (\Sigma, E) \longrightarrow (\Sigma', E')$ un morfism de prezentări, M' un (Σ', E') model și $e \in \text{Sen}(\Sigma)$.

$$\text{Mod}^p(\varphi)(M') \models_{(\Sigma, E)} e$$

adică $\text{Mod}(\varphi)(M') \models_{\Sigma} e$ este echivalentă conform condiției de satisfacere din \mathcal{I} cu $M' \models_{\Sigma} \text{Sen}(\varphi)(e)$, deci echivalentă cu

$$M' \models_{(\Sigma, E)} \text{Sen}^p(\varphi)(e).$$

4.6 TEORII

Se numește *teorie* o prezentare (Σ, E) cu $E = E^\bullet$. Deoarece \bullet este operator de închidere rezultă că teoriile formează pentru fiecare semnătură câte o familie Moore, prin urmare.

Propoziție 4.6.1 *Orice intersecție de teorii este o teorie.*

Propoziție 4.6.2 *Dacă $\varphi : \Sigma \longrightarrow \Sigma'$ este morfism de semnături iar (Σ', E') este o teorie atunci și $(\Sigma, \text{Sen}(\varphi)^{-1}(E'))$ este teorie.*

Demonstrație: *Demonstrația următoare utilizează următoarele proprietăți ale unei funcții $f : A \longrightarrow B$*

1. $f(f^{-1}(D)) \subseteq D$ pentru orice $D \subseteq B$,
2. Dacă $C \subseteq A$ și $D \subseteq B$, $f(C) \subseteq D$ implică $C \subseteq f^{-1}(D)$.

Din lema închiderii

$$\text{Sen}(\varphi)(\text{Sen}(\varphi)^{-1}(E')^\bullet) \subseteq \text{Sen}(\varphi)(\text{Sen}(\varphi)^{-1}(E'))^\bullet.$$

Folosind prima proprietate de mai sus și faptul că închiderea este crescătoare

$$\text{Sen}(\varphi)(\text{Sen}(\varphi)^{-1}(E'))^\bullet \subseteq E'^\bullet = E'.$$

Prin tranzitivitatea incluziunii

$$\text{Sen}(\varphi)(\text{Sen}(\varphi)^{-1}(E'))^\bullet \subseteq E',$$

deci $\text{Sen}(\varphi)^{-1}(E')^\bullet \subseteq \text{Sen}(\varphi)^{-1}(E')$, conform proprietății a doua de mai sus.

4.6.1 Instituția teoriilor

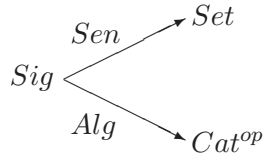
Un morfism de teorii este un morfism de prezentări între două teorii. Se obține astfel o nouă categorie, categoria teoriilor, notată $\text{Th}(\mathcal{I})$. Prin definiție, $\text{Th}(\mathcal{I})$ este o subcategorie plină a lui $\text{Pres}(\mathcal{I})$.

Putem considera restricția lui Mod^p la categoria teoriilor pe care o vom nota cu Mod^t .

$$\begin{array}{ccccc} \text{Th}(\mathcal{I}) & \hookrightarrow & \text{Pres}(\mathcal{I}) & \xrightarrow{\text{Mod}^p} & \text{Cat}^{op} \\ & & \searrow \text{Mod}^t & & \uparrow \end{array}$$

4.7 INSTITUȚIA LOGICII ECUAȚIONALE MULTISORTATE

Instituția logicii ecuaționale multisortate este următoarea construcție



unde categoria Sig a signaturilor algebrice, functorii Alg și Sen și relația de satisfacere \models_Σ vor fi definite în secțiunile următoare.

4.7.1 Categoria signaturilor algebrice Sig

Definiția 4.7.1 Fie S o mulțime și $\Sigma = \{\Sigma_{w,s}\}_{w \in S^*, s \in S}$ o familie de mulțimi.

Perechea $(S, \{\Sigma_{w,s}\}_{w \in S^*, s \in S})$ se numește **signatură algebrică**.

În cele ce urmează vom folosi și notațiile $(S, \Sigma_{w,s})$ sau (S, Σ) .

Vom nota cu Sig categoria signaturilor algebrice construită astfel :

Obiectele categoriei Sig sunt signaturile algebrice.

Dacă $(S, \Sigma_{w,s})$ și $(S', \Sigma'_{w',s'})$ sunt două signaturi algebrice, **morfismele** categoriei Sig între obiectele $(S, \Sigma_{w,s})$ și $(S', \Sigma'_{w',s'})$ sunt de forma

$$\varphi = (f, \{g_{w,s}\}_{w \in S^*, s \in S}) : (S, \Sigma_{w,s}) \longrightarrow (S', \Sigma'_{w',s'}),$$

unde $f : S \longrightarrow S'$ și $g_{w,s} : \Sigma_{w,s} \longrightarrow \Sigma'_{f^*(w), f(s)}$ pentru orice $w \in S^*, s \in S$ sunt funcții. (Am notat cu $f^* : S^* \longrightarrow S'^*$ funcția definită prin $f^*(s_1 s_2 \dots s_n) = f(s_1) f(s_2) \dots f(s_n)$, pentru orice $n \in \mathbb{N}$, și $s_i \in S$ pentru orice $1 \leq i \leq n$).

În cele ce urmează vom utiliza și notația $\varphi = (f, g)$.

Operația de compunere a morfismelor se definește astfel :

Fie $\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$ și $\varphi' = (f', g') : (S', \Sigma') \longrightarrow (S'', \Sigma'')$ două morfisme. Atunci $\varphi\varphi' : (S, \Sigma) \longrightarrow (S'', \Sigma'')$ este morfismul

$$\varphi\varphi' = (ff', \{g_{w,s}g'_{f^*(w), f(s)}\}_{w \in S^*, s \in S}).$$

Propoziție 4.7.2 Sig este o categorie.

Demonstrație: 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, operația de compunere a morfismelor este asociativă. Într-adevăr, fie

$\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$, $\varphi' = (f', g') : (S', \Sigma') \longrightarrow (S'', \Sigma'')$ și $\varphi'' = (f'', g'') : (S'', \Sigma'') \longrightarrow (S''', \Sigma''')$ trei morfisme de signaturi algebrice. Atunci

$$\begin{aligned} (\varphi\varphi')\varphi'' &= ((ff')f'', (g_{w,s}g'_{f^*(w), f(s)})g''_{(ff')^*(w), (ff')(s)}) \\ &= (f(f'f''), g_{w,s}(g'_{f^*(w), f(s)}g''_{f'^*(f^*(w)), f'(f(s))})) \\ &= (f, g)(f'f'', g'_{w',s'}g''_{f'^*(w'), f'(s')}) \\ &= \varphi(\varphi'\varphi'') \end{aligned}$$

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

$$A : S \longrightarrow Set \text{ și } A^{w,s} : \Sigma_{w,s} \longrightarrow (A_w \rightarrow A_s) \text{ pentru orice } w \in S^*, s \in S \text{ sunt funcții.}$$

Vom nota $A_s = A(s)$, pentru orice $s \in S$. Reamintim că $A_{s_1 s_2 \dots s_n} = A_{s_1} \times A_{s_2} \times \dots \times A_{s_n}$. Dacă $\sigma \in \Sigma_{w,s}$ în loc de $A^{w,s}(\sigma)$ mai scriem $A_{\sigma}^{w,s}$ 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

$$Alg(f, g)(A, \{A^{w',s'}\}_{w' \in S'^*, s' \in S'}) = (fA, \{g_{w,s} A^{f^*(w), f(s)}\}_{w \in S^*, s \in S}) \in |Alg_{\Sigma}|$$

Din $f : S \longrightarrow S'$ și $A : S' \longrightarrow Set$ rezultă că $fA : S \longrightarrow Set$. Conform notației precedente, $A_{f(s)} = (fA)_s$ 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 $(A, \{A^{w',s'}\}_{w' \in S'^*, s' \in S'})$.

Observație: Dacă $w = s_1 \dots s_n$, atunci

$$(fA)_w = (fA)_{s_1} \times \dots \times (fA)_{s_n} = A_{f(s_1)} \times \dots \times A_{f(s_n)} = A_{f(s_1) \dots f(s_n)} = A_{f^*(w)}.$$

Deoarece $g_{w,s} : \Sigma_{w,s} \longrightarrow \Sigma'_{f^*(w), f(s)}$ și $A^{f^*(w), f(s)} : \Sigma'_{f^*(w), f(s)} \longrightarrow (A_{f^*(w)} \rightarrow A_{f(s)})$ avem $g_{w,s} A^{f^*(w), f(s)} : \Sigma_{w,s} \longrightarrow ((fA)_w \rightarrow (fA)_s)$, pentru orice $w \in S^*$ și orice $s \in S$.

b) Pe morfisme : Fie $h : (A, A^{w',s'}) \longrightarrow (B, B^{w',s'})$ un morfism de Σ' -algebre. Definim $Alg(f, g)(h) : (fA, \{g_{w,s} A^{f^*(w), f(s)}\}) \longrightarrow (fB, \{g_{w,s} B^{f^*(w), f(s)}\})$ prin

$$Alg(f, g)(h) = fh,$$

unde $(fh)(s) = h_{f(s)} : A_{f(s)} \longrightarrow B_{f(s)}$ pentru orice $s \in S$.

Arătăm că definiția este corectă, adică familia de funcții fh este un morfism de Σ -algebre. Aceasta revine la a arăta că pentru orice $w \in S^*$, orice $s \in S$ și orice $\sigma \in \Sigma_{w,s}$ diagrama următoare este comutativă

$$\begin{array}{ccc} (fA)_w & \xrightarrow{(g_{w,s} A^{f^*(w), f(s)})(\sigma)} & (fA)_s \\ \downarrow (fh)_w & & \downarrow (fh)_s \\ (fB)_w & \xrightarrow{(g_{w,s} B^{f^*(w), f(s)})(\sigma)} & (fB)_s \end{array}$$

Comutativitatea diagramei precedente este echivalentă cu comutativitatea diagramei

$$\begin{array}{ccc}
A_{f^*(w)} & \xrightarrow{A^{f^*(w),f(s)}(g_{w,s}(\sigma))} & A_{f(s)} \\
\downarrow h_{f^*(w)} & & \downarrow h_{f(s)} \\
B_{f^*(w)} & \xrightarrow{B^{f^*(w),f(s)}(g_{w,s}(\sigma))} & B_{f(s)}
\end{array}$$

iar aceasta rezultă din faptul că h este morfism de Σ' -algebre.

Propoziție 4.7.3 $Alg(f,g)$ este functor.

Demonstrație: 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 $h : \mathcal{A} \rightarrow \mathcal{B}$ și $h' : \mathcal{B} \rightarrow \mathcal{C}$ două morfisme de Σ' -algebre. Prin urmare pentru orice $s \in S$

$$\begin{aligned}
(Alg(f,g)(h); Alg(f,g)(h'))_s &= Alg(f,g)(h)_s; Alg(f,g)(h')_s \\
&= (fh)_s; (fh')_s = h_{f(s)}; h'_{f(s)} \\
&= (hh')_{f(s)} = (f(hh'))_s = (Alg(f,g)(hh'))_s,
\end{aligned}$$

deci $Alg(f,g)(h); Alg(f,g)(h') = Alg(f,g)(hh')$. Așadar $Alg(f,g)$ este functor.

Propoziție 4.7.4 Alg este functor.

Demonstrație: Pentru orice semnătură algebrică (S, Σ) avem evident $Alg(1_{(S, \Sigma)}) = 1_{Alg(S)}$.

Fie $\varphi = (f, g) : (S, \Sigma) \rightarrow (S', \Sigma')$ și $\varphi' = (f', g') : (S', \Sigma') \rightarrow (S'', \Sigma'')$ două morfisme de semnături algebrice. Vom arăta că functorii $Alg(\varphi')Alg(\varphi)$ și $Alg(\varphi\varphi')$ coincid, demonstrând că acționează identic pe obiecte și pe morfisme.

Fie $\mathcal{A} = (A, \{A^{w,s}\}_{w \in S'^*, s \in S''})$ o Σ'' -algebră. Atunci

$$\begin{aligned}
(Alg(\varphi')Alg(\varphi))(\mathcal{A}) &= Alg(\varphi)(Alg(\varphi')(\mathcal{A})) = Alg(\varphi)(f'A, \{g'_{w,s}A^{f'^*(w),f'(s)}\}_{w \in S'^*, s \in S'}) = \\
&= (f(f'A), \{g_{w,s}(g'_{f^*(w),f(s)}A^{f'^*(f^*(w)),f'(f(s))})\}_{w \in S^*, s \in S}) \\
&= ((ff')A, \{(g_{w,s}g'_{f^*(w),f(s)})A^{(ff')^*(w),(ff')(s)}\}_{w \in S^*, s \in S}) \\
&= Alg(ff', \{g_{w,s}g'_{f^*(w),f(s)}\})(\mathcal{A}) = Alg(\varphi\varphi')(\mathcal{A}).
\end{aligned}$$

Pentru orice morfism de Σ'' -algebre $h : \mathcal{A} \rightarrow \mathcal{B}$

$$(Alg(\varphi')Alg(\varphi))(h) = Alg(f,g)(Alg(f',g')(h)) = Alg(f,g)(f'h) = f(f'h) = (ff')h = Alg(\varphi\varphi')(h).$$

Deci $Alg(\varphi)Alg(\varphi') = Alg(\varphi\varphi')$, oricare ar fi morfismele de semnături algebrice φ și φ' . Așadar Alg este functor.

4.7.3 Functorul Sen

Un comentariu este necesar pentru a motiva prezentarea de mai jos.

Propozițiile logicii ecuaționale sunt și propoziții ale calculului cu predicare cu egalitate. Reciproca nu este adevărată. În multe variante ale calculului cu predicate apar variabile libere și variabile legate, fapt

ce îngreunează, cel puțin din punct de vedere tehnic, atât expunerea cât și înțelegerea. Vom folosi din nou artificii de transformare a variabilelor în operații constante. După cum am mai făcut-o în capitolele precedente, această transformare a variabilelor în constante se realizează prin schimbarea semnăturii. Acest fapt explică utilizarea unei algebre inițiale în locul unei algebre libere pentru modelarea “expresiilor” din calculul cu predicate.

Pentru orice morfism de semnături algebrice $\varphi : (S, \Sigma) \longrightarrow (S', \Sigma')$, notăm cu $\varphi^\# : T_\Sigma \longrightarrow \text{Alg}(\varphi)(T_{\Sigma'})$ unicul morfism de Σ -algebre.

Definim functorul $\text{Sen} : \text{Sig} \longrightarrow \text{Set}$ astfel :

1. Pe obiecte : Pentru orice semnătură algebrică Σ

$$\text{Sen}(\Sigma) = T_\Sigma \times T_\Sigma = \{l \stackrel{\circ}{=}_s r \mid s \in S, \quad l, r \in (T_\Sigma)_s\}_{s \in S}.$$

2. Pe morfisme : Pentru orice morfism de semnături algebrice $\varphi : (S, \Sigma) \longrightarrow (S', \Sigma')$ funcția $\text{Sen}(\varphi) : \text{Sen}(\Sigma) \longrightarrow \text{Sen}(\Sigma')$ este definită prin

$$\text{Sen}(\varphi)(l \stackrel{\circ}{=}_s r) = \varphi_s^\#(l) \stackrel{\circ}{=}_{f(s)} \varphi_s^\#(r).$$

Propoziție 4.7.5 *Sen este functor.*

Demonstrație:

Fie $\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$ și $\psi = (f', g') : (S', \Sigma') \longrightarrow (S'', \Sigma'')$ două morfisme de semnături algebrice. Deoarece $\text{Alg}(\varphi)(\text{Alg}(\psi)(T_{\Sigma''})) = (\text{Alg}(\psi)\text{Alg}(\varphi))(T_{\Sigma''}) = \text{Alg}(\varphi\psi)(T_{\Sigma''})$ și $\psi^\# : T_{\Sigma'} \rightarrow \text{Alg}(\psi)(T_{\Sigma''})$, deducem

$$\text{Alg}(\varphi)(\psi^\#) : \text{Alg}(\varphi)(T_{\Sigma'}) \longrightarrow \text{Alg}(\varphi\psi)(T_{\Sigma''}),$$

prin urmare există două morfisme

$$T_\Sigma \xrightarrow[\quad (\varphi\psi)^\# \quad]{\varphi^\#; \text{Alg}(\varphi)(\psi^\#)} \text{Alg}(\varphi\psi)(T_{\Sigma''})$$

care deoarece T_Σ este Σ -algebră inițială sunt egale. Așadar avem

$$\begin{aligned} (\text{Sen}(\varphi); \text{Sen}(\psi))(l \stackrel{\circ}{=}_s r) &= \text{Sen}(\psi)(\varphi_s^\#(l) \stackrel{\circ}{=}_{f(s)} \varphi_s^\#(r)) \\ &= (\psi_{f(s)}^\#(\varphi_s^\#(l)) \stackrel{\circ}{=}_{f'(f(s))} \psi_{f(s)}^\#(\varphi_s^\#(r))) \\ &= (\varphi_s^\#; \text{Alg}(\varphi)(\psi^\#)_s)(l) \stackrel{\circ}{=}_{(ff')(s)} (\varphi_s^\#; \text{Alg}(\varphi)(\psi^\#)_s)(r) \\ &= (\varphi\psi)_s^\#(l) \stackrel{\circ}{=}_{(ff')(s)} (\varphi\psi)_s^\#(r) \\ &= \text{Sen}(\varphi\psi)(l \stackrel{\circ}{=}_s r), \end{aligned}$$

deci Sen este functor.

4.7.4 Incluziuni de semnături

Definiția 4.7.6 O incluziune de semnături este un morfism de semnături

$$\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$$

care verifică condițiile suplimentare

1. S este o submulțime a lui S' și f este funcția incluziune a lui S în S'
2. Pentru orice $w \in S^*$ și $s \in S$, $\Sigma_{w,s}$ este o submulțime a lui $\Sigma'_{f^*(w),f(s)}$ și funcția $g_{w,s}$ este incluziunea dintre cele două mulțimi.

Propoziție 4.7.7 *Dacă φ este o incluziune de semnături, atunci $\text{Alg}(\varphi)$ este un functor uituc.*

Cuantificarea $.s$ are rolul de a indica sortul.

Fie (S, Σ) o semnătură algebrică. Vom nota cu L_Σ o anumită Σ -algebră inițială. Suportul ei este definit drept cea mai mică familie $\{T_s\}_{s \in S}$ de mulțimi cu proprietatea

$$\sigma \in \Sigma_{s_1 s_2 \dots s_n, s}, x_i \in T_{s_i} \implies \sigma(x_1, x_2, \dots, x_n).s \in T_s.$$

Folosind construcția de mai sus observăm că pentru orice incluziune de semnături $i : (S, \Sigma) \hookrightarrow (S', \Sigma')$ obținem $(L_\Sigma)_s \subset (L_{\Sigma'})_s$ pentru orice $s \in S$. Mai mult L_Σ este subalgebră a lui $\text{Alg}(i)(L_{\Sigma'})$, morfismul de incluziune fiind chiar $i^\#$.

Propoziție 4.7.8 *Dacă i este o incluziune de semnături, atunci $\text{Sen}(i)$ este o incluziune de mulțimi.*

4.7.5 Relația și condiția de satisfacere

Vom defini în continuare relația de satisfacere.

Definiția 4.7.9 Fie \mathcal{A} o Σ -algebră. Spunem că \mathcal{A} satisface Σ -ecuația $l \stackrel{\circ}{=}_s r$ dacă și numai dacă $u_s(l) = u_s(r)$ pentru unicul Σ -morfism $u : T_\Sigma \longrightarrow \mathcal{A}$.

Notații 4.7.10 *Dacă \mathcal{A} satisface $l \stackrel{\circ}{=}_s r$, vom nota $\mathcal{A} \models_\Sigma l \stackrel{\circ}{=}_s r$.*

Cu această definiție are loc

Propoziție 4.7.11 *Fie $\varphi = (f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$ un morfism de semnături algebrice, $l \stackrel{\circ}{=}_s r \in \text{Sen}(\Sigma)$ și \mathcal{B} o Σ' -algebră. Atunci este îndeplinită condiția de satisfacere*

$$\text{Alg}(\varphi)(\mathcal{B}) \models_\Sigma l \stackrel{\circ}{=}_s r \iff \mathcal{B} \models_{\Sigma'} \text{Sen}(\varphi)(l \stackrel{\circ}{=}_s r).$$

Demonstrație: Notând cu $t : T_{\Sigma'} \longrightarrow \mathcal{B}$ unicul Σ' -morfism deducem că

$$\varphi^\#; \text{Alg}(\varphi)(t) : T_\Sigma \longrightarrow \text{Alg}(\varphi)(\mathcal{B})$$

este unicul Σ -morfism. Din definiția relației de satisfacere deducem următoarele echivalențe:

$$\begin{aligned} \text{Alg}(\varphi)(\mathcal{B}) \models_\Sigma l \stackrel{\circ}{=}_s r &\Leftrightarrow \\ (\varphi^\#; \text{Alg}(\varphi)(t))_s(l) &= (\varphi^\#; \text{Alg}(\varphi)(t))_s(r) \Leftrightarrow \\ (\varphi_s^\#; \text{Alg}(\varphi)(t)_s(l)) &= (\varphi_s^\#; \text{Alg}(\varphi)(t)_s(r)) \Leftrightarrow \\ t_{f(s)}(\varphi_s^\#(l)) &= t_{f(s)}(\varphi_s^\#(r)) \Leftrightarrow \\ \mathcal{B} \models_{\Sigma'} \varphi_s^\#(l) &\stackrel{\circ}{=}_{f(s)} \varphi_s^\#(r) \Leftrightarrow \\ \mathcal{B} \models_{\Sigma'} \text{Sen}(\varphi)(l \stackrel{\circ}{=}_s r). \end{aligned}$$

Cu aceasta condiția de satisfacere este demonstrată.

Chapter 5

MODULARIZARE

5.1 LIMITE și COLIMITE ÎNTR-O CATEGORIE

Limitele și colimitele sunt noțiuni duale. Din punct de vedere tehnic nu contează pe care o tratăm. Vom prefera colimitele deoarece au mai multe aplicații în informatică.

Definiția 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} și 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 și (γ_i, A) . \square

Compunerea unui cocon cu un morfism este tot un cocon.

Observația 5.1.2 Dacă $F(i) \xrightarrow{\gamma_i} A$ este un cocon pentru $F : I \longrightarrow \mathcal{C}$ și dacă $f : A \longrightarrow B$ este un morfism, atunci $F(i) \xrightarrow{\gamma_i; f} B$ este un cocon pentru F .

Demonstrație: 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 într-un cocon.

Observația 5.1.3 Dacă $F(i) \xrightarrow{\gamma_i} A$ este un cocon pentru $F : I \longrightarrow \mathcal{C}$ și dacă $G : \mathcal{C} \longrightarrow \mathcal{D}$ este un functor, atunci $G(F(i)) \xrightarrow{G(\gamma_i)} G(A)$ este un cocon pentru $F; G$.

Demonstrație: 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

Definiția 5.1.4 Un cocon $F(i) \xrightarrow{\gamma_i} A$ pentru functorul $F : I \longrightarrow \mathcal{C}$ se numește **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 notația

$$\operatorname{colim} F = F(i) \xrightarrow{\gamma_i} A. \quad \square$$

Definiția 5.1.5 O categorie se numește mică dacă obiectele sale formează o mulțime.

Un caz particular de colimite sunt **sumele directe**, numite și coproduse. Ele se obțin pentru o categorie mică I în care toate morfismele sunt identități, adică o mulțime. 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 mulțimea I . Observăm că un cocon pentru F este pur și simplu o familie de morfisme $\alpha_i : F(i) \longrightarrow A$, deoarece condiția impusă mai sus pentru morfismele din I este automat îndeplinită pentru morfismele identitate.

Definiția 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 .

Menționăm că în categoria atașată unei mulțimi parțial ordonate sumele directe, dacă există, coincid cu supremurile.

5.1.1 Colimite de mulțimi

Propoziție 5.1.7 În cazul particular al teoriei multimilor, suma directă a unei familii $\{F_i\}_{i \in I}$ de mulțimi este reuniunea lor disjunctă $A = \bigcup_{i \in I} F_i \times \{i\}$ împreună cu familia de funcții $\{\alpha_i : F_i \longrightarrow A\}_{i \in I}$ definită pentru orice obiect i din I și orice $a \in F_i$ prin $\alpha_i(a) = (a, i)$.

Demonstrație: Fie $\{\beta_i : F_i \longrightarrow B\}_{i \in I}$ o familie de funcții. Definim funcția $\beta : A \longrightarrow B$ prin $\beta(a, i) = \beta_i(a)$ pentru orice $i \in I$ și $a \in F_i$. Observăm că pentru orice $i \in I$ și $a \in F_i$

$$(\alpha_i; \beta)(a) = \beta(\alpha_i(a)) = \beta(a, i) = \beta_i(a).$$

Deci pentru orice $i \in I$ rezultă că $\alpha_i; \beta = \beta_i$.

Pentru a proba unicitatea presupunem că funcția $\gamma : A \longrightarrow B$ verifică egalitățile $\alpha_i; \gamma = \beta_i$ pentru orice $i \in I$. Observăm pentru orice $i \in I$ și orice $a \in F_i$ că

$$\gamma(a, i) = \gamma(\alpha_i(a)) = (\alpha_i; \gamma)(a) = \beta_i(a) = \beta(a, i)$$

deci $\gamma = \beta$. \square

Definiția 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.

Propoziție 5.1.9 *Categoria mulțimilor are colimite.*

Demonstrație: Fie I o categorie mică și $F : I \longrightarrow \text{Set}$ un functor. Vom construi o colimită pentru F .

Fie (α_i, A) suma directă a familiei de mulțimi $\{F(i)\}_{i \in |I|}$ construită ca mai sus. Definim în mulțimea A relația

$$R = \{((a, i), (F(\alpha)(a), j)) : i, j \in |I|, a \in F(i), \alpha \in I(i, j)\}.$$

Fie \sim relația de echivalență generată de R , adică cea mai mică relație de echivalență din A care include R . Fie A/\sim cântul lui A prin echivalența \sim și $s : A \longrightarrow A/\sim$ surjecția canonică care duce fiecare element din A în clasa sa de echivalență. 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 relația R este inclusă în echivalența nucleară a lui f . Fie $\alpha \in I(a, b)$ un morfism din I și $a \in F(i)$. Observăm că

$$\begin{aligned} f(F(\alpha)(a), j) &= f(\alpha_j(F(\alpha)(a))) = (\alpha_j; f)(F(\alpha)(a)) = \beta_j(F(\alpha)(a)) = (F(\alpha); \beta_j)(a) = (\text{deoarece } \{\beta_i\} \text{ este cocon}) \\ &= \beta_i(a) = (\alpha_i; f)(a) = f(\alpha_i(a)) = f(a, i). \end{aligned}$$

Rezultă că \sim este inclusă în echivalența nucleară a lui f , prin urmare există o unică funcție $g : A/\sim \longrightarrow B$ 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 funcție 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 surjectivității lui s că $h = g$. \square

5.1.2 Despre unicitatea colimitelor

Pentru un functor dat $F : I \longrightarrow \mathcal{C}$ colimitele lui sunt unice abstractie de un izomorfism. Acest fapt este ilustrat de urmatoarele două propoziții.

Propoziție 5.1.10 *Dacă (α_i, A) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ și $j : A \longrightarrow B$ un izomorfism, atunci $(\alpha_i; j, B)$ este o colimită a lui F .*

Demonstrație: Observăm că $(\alpha_i; j, B)$ este cocon deoarece se obține 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 $g = j^{-1}; f$.

Propoziție 5.1.11 *Două colimite ale aceluiași functor sunt izomorfe.*

Demonstrație: Fie (α_i, A) și (β_i, B) două colimite ale functorului $F : I \longrightarrow \mathcal{C}$.

Deoarece (α_i, A) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ și (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 $F : I \longrightarrow \mathcal{C}$ și (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 $\alpha_i; (f; g) = (\alpha_i; f); g = \beta_i; g = \alpha_i$ pentru orice obiect i din I , ținând cont că (α_i, A) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ deducem $f; g = 1_A$.

Deoarece $\beta_i; (g; f) = (\beta_i; g); f = \alpha_i; f = \beta_i$ pentru orice obiect i din I , ținând cont că (β_i, B) este o colimită a functorului $F : I \longrightarrow \mathcal{C}$ deducem $g; f = 1_B$.

Egalitățile $f; g = 1_A$ și $g; f = 1_B$ arată că f și g sunt izomorfisme inverse unul altuia.

5.1.3 Crearea colimitelor

Definiția 5.1.12 Fie $F : I \longrightarrow \mathcal{C}$ un functor și (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 conservă $G : \mathcal{C} \longrightarrow \mathcal{D}$ colitele dacă el conservă cu orice colimită a oricărui functor $F : I \longrightarrow \mathcal{C}$.

Observația 5.1.13 *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 .*

Demonstrație: Considerăm o colimită a functorului F

$$\text{Colim } F = (\gamma_i, C).$$

pe care functorul G o duce într-o colimită a functorului FG

$$\text{Colim } FG = (G(\gamma_i), G(C)).$$

Fie (γ'_i, C') o altă colimită a functorului F .

Cele două colimite ale lui F fiind izomorfe rezultă existența 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 și $\text{Colim } FG = (G(\gamma_i), G(C))$ rezultă că $(G(\gamma'_i), G(C'))$ este o colimită a functorului FG .

Definiția 5.1.14 Functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ **crează slab(crează) colimitele** dacă pentru orice functor $F : I \longrightarrow \mathcal{C}$ și pentru orice colimită (δ_i, D) a functorului FG există(și e unică) o colimită (γ_i, C) a functorului F cu proprietățile $G(C) = D$ și $G(\gamma_i) = \delta_i$ pentru orice obiect i din I .

Propoziție 5.1.15 *Dacă categoria \mathcal{D} are colimite și functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ crează slab colimitele, atunci categoria \mathcal{C} are colimite și functorul G comută cu colimitele.*

Demonstrație: Fie $F : I \longrightarrow \mathcal{C}$ un functor și o colimită a functorului FG

$$\text{Colim } FG = (\delta_i, D).$$

Deoarece functorul $G : \mathcal{C} \longrightarrow \mathcal{D}$ crează slab colimitele există o colimită (γ_i, C) a functorului F cu proprietățile $G(C) = D$ și $G(\gamma_i) = \delta_i$ pentru orice obiect i din I . Am probat existența unei colimite a lui F cu care G comută. Este acum suficient să aplicăm observația precedentă.

5.2 COLIMITE DE SIGNATURI ALGEBRICE

Functorul uituc $U : \text{Sig} \longrightarrow \text{Set}$ este definit astfel:

- pe obiecte: pentru orice semnătură algebrică $(S, \Sigma) \in |\text{Sig}|$

$$U(S, \Sigma) = S;$$

- pe morfisme: pentru orice morfism de semnături algebrice $(f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$

$$U(f, g) = f.$$

Teorema 5.2.1 *Functorul uituc $U : \text{Sig} \longrightarrow \text{Set}$ crează slab (dar nu crează) colimitele.*

Demonstrație: Fie I o categorie mică și $F : I \longrightarrow \text{Sig}$ un functor. Folosim următoarele notații:

$$F(i) = (S_i, \Sigma^i) \quad \text{pentru orice obiect } i \in |I|,$$

$$F(\alpha) = (f_\alpha, g^\alpha) : (S_i, \Sigma^i) \longrightarrow (S_j, \Sigma^j) \quad \text{pentru orice morfism } \alpha \in I(i, j).$$

Dacă $\alpha : i \longrightarrow j$ și $\beta : j \longrightarrow k$ sunt morfisme din I din $F(\alpha; \beta) = F(\alpha); F(\beta)$, adică

$$(f_{\alpha; \beta}, g_{w^i, s^i}^{\alpha; \beta}) = (f_\alpha; f_\beta, g_{w^i, s^i}^\alpha; g_{f_\alpha^*(w^i), f_\alpha(s^i)}^\beta)$$

deducem

$$f_{\alpha; \beta} = f_\alpha; f_\beta \quad \text{și} \quad g_{w^i, s^i}^{\alpha; \beta} = g_{w^i, s^i}^\alpha; g_{f_\alpha^*(w^i), f_\alpha(s^i)}^\beta. \quad (5.1)$$

Fie

$$\text{colim } FU = (f_i, S) = S_i \xrightarrow{f_i} S \quad (5.2)$$

Observăm că pentru orice morfism $\alpha : i \longrightarrow j$ din I

$$f_i = f_\alpha; f_j.$$

Pentru orice $w \in S^*$ **și** $s \in S$ vom constui mulțimea $\Sigma_{w, s}$. Definim categoria mică $\mathcal{C}_{w, s}$ astfel:
pe obiecte:

$$|\mathcal{C}_{w, s}| = \{ (i, w^i, s^i) \mid i \in |I|, w^i \in S_i^*, s^i \in S_i, \text{ cu } f_i^*(w^i) = w, f_i(s^i) = s \},$$

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 $\alpha : (i, w^i, s^i) \longrightarrow (j, w^j, s^j)$ și $\beta : (j, w^j, s^j) \longrightarrow (k, w^k, s^k)$ observăm că

$(f_{\alpha;\beta})(s^i) = f_\beta(f_\alpha(s^i)) = f_\beta(s^j) = s^k$ și că $(f_{\alpha;\beta}^*)(w^i) = f_\beta^*(f_\alpha^*(w^i)) = f_\beta^*(w^j) = w^k$,
ceea ce arată că

$\alpha \in \mathcal{C}_{w,s}((i, w^i, s^i), (j, w^j, s^j))$ și $\beta \in \mathcal{C}_{w,s}((j, w^j, s^j), (k, w^k, s^k))$ implică $\alpha; \beta \in \mathcal{C}_{w,s}((i, w^i, s^i), (k, w^k, s^k))$.

Definim functorul:

$$G_{w,s} : \mathcal{C}_{w,s} \longrightarrow \text{Set}$$

pe obiecte:

$$G_{w,s}(i, w^i, s^i) = \Sigma_{w^i, s^i}^i;$$

pe morfisme: pentru orice $\alpha \in \mathcal{C}_{w,s}((i, w^i, s^i), (j, w^j, s^j))$

$$G_{w,s}(\alpha) = g_{w^i, s^i}^\alpha : \Sigma_{w^i, s^i}^i \longrightarrow \Sigma_{w^j, s^j}^j.$$

Pentru morfismele $\alpha : (i, w^i, s^i) \longrightarrow (j, w^j, s^j)$ și $\beta : (j, w^j, s^j) \longrightarrow (k, w^k, s^k)$ observăm că

$$G_{w,s}(\alpha); G_{w,s}(\beta) = g_{w^i, s^i}^\alpha; g_{w^j, s^j}^\beta = g_{w^i, s^i}^\alpha; g_{f_\alpha^*(w^i), f_\alpha(s^i)}^\beta = g_{w^i, s^i}^{\alpha; \beta} = G_{w,s}(\alpha; \beta)$$

Fie

$$\text{colim } G_{w,s} = (g_{w^i, s^i}^i, \Sigma_{w,s}) \quad (5.3)$$

Pentru orice $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$, deoarece $f_i^*(w^i) = w$ și $f_i^*(s^i) = s$ observăm că

$$g_{w^i, s^i}^i : \Sigma_{w^i, s^i}^i \longrightarrow \Sigma_{w,s}.$$

Pentru orice $i \in |I|$, $w^i \in S_i^*$ și $s^i \in S_i$, pentru $w = f_i^*(w^i)$ și $s = f_i^*(s^i)$ constatăm existența morfismului

$$g_{w^i, s^i}^i : \Sigma_{w^i, s^i}^i \longrightarrow \Sigma_{f_i^*(w^i), f_i(s^i)}.$$

Vom dovedi

$$\text{colim } F = ((f_i, g^i), (S, \Sigma)).$$

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 mai trebuie probat pentru $w^i \in S_i^*$ și $s^i \in S_i$ că

$$g_{w^i, s^i}^\alpha g_{f_\alpha^*(w^i), f_\alpha(s^i)}^j = g_{w^i, s^i}^i$$

Definind $w = f_i^*(w^i)$ și $s = f_i(s^i)$ observăm că $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$. Deoarece $w = f_j^*(f_\alpha^*(w^i))$ și $s = f_j(f_\alpha(s^i))$, luând $w^j = f_\alpha^*(w^i)$ și $s^j = f_\alpha(s^i)$ deoarece $w = f_j^*(w^j)$ și $s = f_j(s^j)$ observăm că $(j, w^j, s^j) \in |\mathcal{C}_{w,s}|$ și că $\alpha : (i, w^i, s^i) \longrightarrow (j, w^j, s^j)$ este morfism în $\mathcal{C}_{w,s}$. Proprietatea de con inductiv a lui $(g_{w^i, s^i}^i, \Sigma_{w,s})$ aplicată lui α implică comutativitatea următoarei diagrame.

$$\begin{array}{ccc}
G_{w,s}(i, w^i, s^i) = \Sigma_{w^i, s^i}^i & \xrightarrow{G_{w,s}(\alpha) = g_{w^i, s^i}^\alpha} & \Sigma_{w^j, s^j}^j = G_{w,s}(j, w^j, s^j) \\
& \searrow g_{w^i, s^i}^i & \swarrow g_{w^j, s^j}^j \\
& \Sigma_{w,s} &
\end{array}$$

Comutativitatea diagramei este chiar egalitatea de mai sus.

Fie $((u_i, v^i), (S', \Sigma'))$ alt con inductiv de bază F .

Deoarece (u_i, S') este con inductiv de bază FU există un unic morfism $f : S \longrightarrow S'$ cu proprietatea $f_i; f = u_i$ pentru orice $i \in |I|$.

$$\begin{array}{ccc}
S_i & \xrightarrow{f_i} & S \\
& \searrow u_i & \swarrow f \\
& S' &
\end{array}$$

Pentru orice $i \in |I|$, orice $w^i \in S_i^*$ și orice $s^i \in S$ considerăm morfismul

$$v_{w^i, s^i}^i : \Sigma_{w^i, s^i}^i \longrightarrow \Sigma_{u_i^*(w^i), u_i(s^i)}^i$$

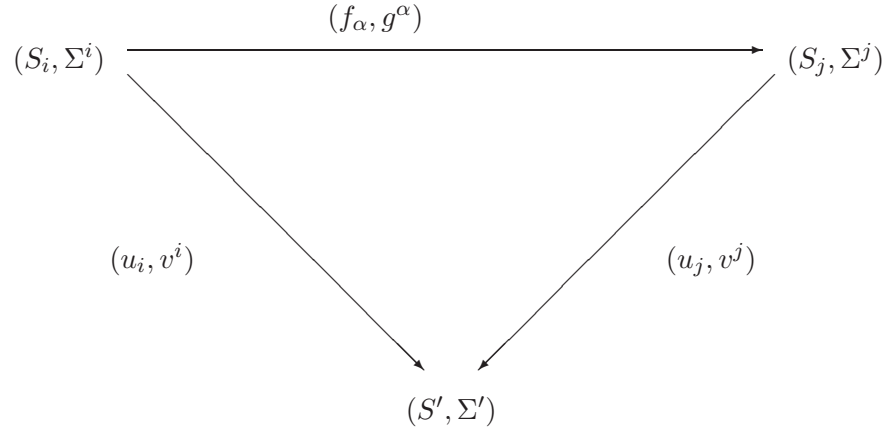
Dacă $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$ utilizând $f^*(w) = f^*(f_i^*(w^i)) = u_i^*(w^i)$ și $f(s) = u_i(s^i)$ observăm că

$$v_{w^i, s^i}^i : G_{w,s}(i, w^i, s^i) \longrightarrow \Sigma_{f^*(w), f(s)}^i.$$

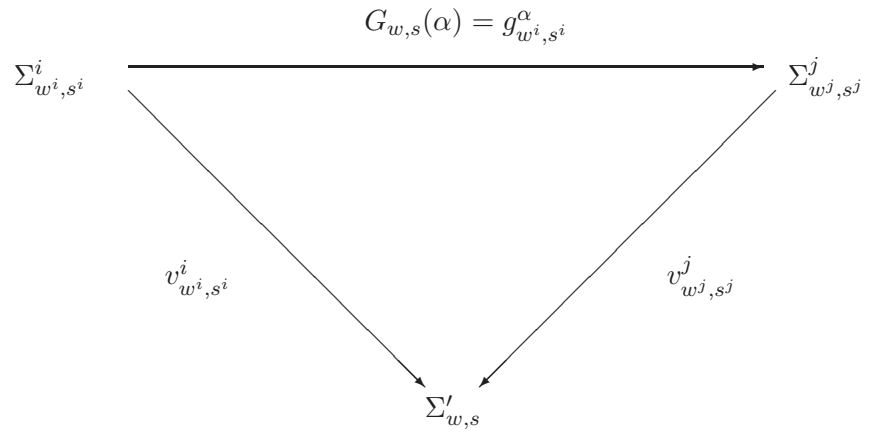
Pentru orice $w \in S^*$ și $s \in S$ probăm că

$$(v_{w^i, s^i}^i, \Sigma_{f^*(w), f(s)}^i)$$

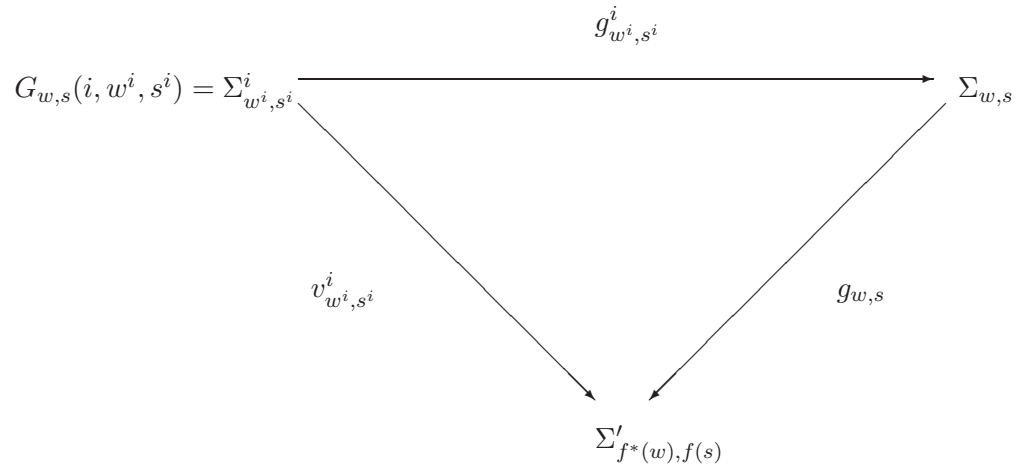
este cocon pentru $G_{w,s}$. Intr-adevăr pentru orice morfism α de la (i, w^i, s^i) la (j, w^j, s^j) din $\mathcal{C}_{w,s}$ din proprietatea de cocon a lui $((u_i, v^i), (S', \Sigma'))$ aplicată morfismului $\alpha : i \longrightarrow j$ obținem comutativitatea diagramei



prin urmare folosind componentele de pe a doua poziție deducem $g_{w^i, s^i}^\alpha v_{w^j, s^j}^j = v_{w^i, s^i}^i$, deci următoarea diagramă este comutativă



Prin urmare pentru orice $w \in S^*$ și $s \in S$ există un unic morfism $g_{w, s} : \Sigma_{w, s} \longrightarrow \Sigma'_{f(w), f(s)}$ cu proprietatea $g_{w^i, s^i}^i; g_{w, s} = v_{w^i, s^i}^i$ pentru orice $(i, w^i, s^i) \in |\mathcal{C}w, s|$.



Deoarece $g_{w^i, s^i}^i; g_{f_i^*(w^i), f_i(s^i)} = v_{w^i, s^i}^i$ deducem că $(f_i, g^i); (f, g) = (u_i, v^i)$.

$$\begin{array}{ccc}
 (S_i, \Sigma^i) & \xrightarrow{(f_i, g^i)} & (S, \Sigma) \\
 & \searrow (u_i, v^i) \quad \swarrow (f, g) & \\
 & (S', \Sigma') &
 \end{array}$$

Unicitatea se arată ușor. \square

Folosind propoziția 5.1.15 și existența colimitelor din **Set** deducem

Corolar 5.2.2 *Categoria signaturilor algebrice are colimite și functorul uituc comută cu colimite.*

5.3 FUNCTORUL Alg CONSERVĂ COLIMITELE

Reamintim definiția functorului

$$\text{Alg} : \text{Sig} \longrightarrow \text{Cat}^{op}$$

după cum urmează:

1) pe obiecte: pentru orice signatură algebrică (S, Σ)

$$\text{Alg}(S, \Sigma) = \text{Alg}_\Sigma,$$

unde Alg_Σ 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

$$A : S \longrightarrow \text{Set} \quad \text{și} \quad A_{w,s} : \Sigma_{w,s} \longrightarrow (A_w \rightarrow A_s) \text{ pentru orice } w \in S^*, s \in S.$$

2) pe morfisme: pentru orice morfism de semnături $(f, g) : (S, \Sigma) \longrightarrow (S', \Sigma')$

$$\text{Alg}(f, g) : \text{Alg}_{\Sigma'} \longrightarrow \text{Alg}_\Sigma,$$

este functorul definit astfel:

α) pe obiecte: pentru orice Σ' -algebra $(A, \{A_{w,s}\}_{w \in S'^*, s \in S'})$

$$\text{Alg}(f, g)(A, \{A_{w,s}\}_{w \in S'^*, s \in S'}) = (f; A, \{g_{w,s}; A_{f^*(w), f(s)}\}_{w \in S^*, s \in S}) \in |\text{Alg}_\Sigma|. \quad (5.4)$$

Din $f : S \longrightarrow S'$ și $A : S' \longrightarrow \text{Set}$ rezultă că

$$fA : S \longrightarrow \text{Set};$$

Pentru orice $w \in S^*$ și $s \in S$ din $g_{w,s} : \Sigma_{w,s} \longrightarrow \Sigma'_{f^*(w), f(s)}$ și $A_{f^*(w), f(s)} : \Sigma'_{f^*(w), f(s)} \longrightarrow (A_{f^*(w)} \longrightarrow A_{f(s)})$; 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

$$\text{Alg}(f, g)(h) = f; h. \quad (5.5)$$

5.3.1 Alg conservă colimitele

Fie I o categorie mică și $F : I \longrightarrow \text{Sig}$ un functor. Folosim aceleași notații și construim ca în secțiunea anterioară

$$\text{colim } F = ((f_i, g^i), (S, \Sigma)).$$

Vom arăta în continuare că functorul Alg comută cu colimitele, adică $\lim(F \text{Alg})$ este

$$\text{Alg}(f_i, g^i) : \text{Alg}_\Sigma \longrightarrow \text{Alg}_{\Sigma^i}.$$

Este un con(proiectiv) fiind obținut prin aplicarea functorului Alg unui cocon.

Fie \mathcal{C} o categorie și pentru orice $i \in |I|$, fie $G_i : \mathcal{C} \rightarrow \text{Alg}_{\Sigma^i}$ 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 și $G_i(c)^o$ sunt operațiile algebrei.

Presupunem că (\mathcal{C}, G_i) este con(proiectiv) pentru $F \text{Alg}$, adică pentru orice morfism $\alpha \in I(i, j)$, următoarea diagramă este comutativă:

$$\begin{array}{ccc} & \mathcal{C} & \\ G_i \swarrow & & \searrow G_j \\ \text{Alg}_{\Sigma^i} & \xleftarrow{\text{Alg}(f_\alpha, g^\alpha)} & \text{Alg}_{\Sigma^j} \end{array}$$

prin urmare pentru orice morfism $\alpha \in I(i, j)$

a) pentru orice obiect c din \mathcal{C}

$$(G_i(c)^s, G_i(c)^o) = \text{Alg}(f_\alpha, g^\alpha)((G_j(c)^s, G_j(c)^o) = (f_\alpha; G_j(c)^s, \{g_{w,s}^\alpha; G_j(c)^o_{f_\alpha^*(w), f_\alpha(s)}\}_{w \in S_i^*, s \in S_i})$$

de unde pentru fiecare componentă obținem

$$f_\alpha G_j(c)^s = G_i(c)^s \quad (5.6)$$

și pentru orice $w^i \in S_i^*$ și $s^i \in S_i$

$$G_i(c)^o_{w^i, s^i} = g_{w^i, s^i}^\alpha; G_j(c)^o_{f_\alpha^*(w^i), f_\alpha(s^i)}, \quad (5.7)$$

b) pentru orice morfism $u \in \mathcal{C}(c, c')$

$$\text{Alg}(f_\alpha, g^\alpha)(G_j(u)) = f_\alpha G_j(u) = G_i(u) \quad (5.8)$$

1. Existența

Vom construi un functor $G : \mathcal{C} \longrightarrow \text{Alg}_\Sigma$ astfel încât următoarea diagramă să fie comutativă pentru orice obiect i din I :

$$\begin{array}{ccc}
 Alg_{\Sigma} & \xrightarrow{Alg(f_i, g^i)} & Alg_{\Sigma^i} \\
 & \searrow G \quad \nearrow G_i & \\
 & \mathcal{C} &
 \end{array}$$

Construcția lui G pe obiecte

Din relația (5.6) rezultă că familia de funcții $G_i(c)^s : S_i \longrightarrow Set$ formează un con inductiv pentru FU prin urmare există o unică funcție $G(c)^s : S \longrightarrow Set$ astfel încât pentru orice obiect i din $|I|$

$$f_i; G(c)^s = G_i(c)^s \quad (5.9)$$

$$\begin{array}{ccc}
 S_i & \xrightarrow{f_i} & S \\
 & \searrow G_i(c)^s \quad \nearrow G(c)^s & \\
 & Set &
 \end{array}$$

$G(c)^s$ astfel determinat va constitui suportul Σ -algebrei $G(c)$.

Din (5.9) deducem că pentru orice $s_i \in S_i$

$$G(c)_{f_i(s_i)}^s = G_i(c)_{s_i}^s \quad (5.10)$$

și analog pentru orice cuvînt $w^i \in S_i^*$

$$G(c)_{f_i^*(w^i)}^s = G_i(c)_{w^i}^s. \quad (5.11)$$

Operațiile Σ -algebrei $G(c)$ se construiesc astfel:

Fie $w \in S^*$ și $s \in S$. Familia de funcții $G_i(c)_{w^i, s^i}^o : \Sigma_{w^i, s^i}^i \longrightarrow (G_i(c)_{w^i}^s \longrightarrow G_i(c)_{s^i}^s)$ indexată de $(i, w^i, s^i) \in |\mathcal{C}_{w, s}|$, formează, așa cum ne arată relațiile (5.7), un cocon pentru $G_{w, s}$. Este util să remarcăm aici că toate codomeniile funcțiilor din familia de mai sus coincid deoarece conform egalităților (5.10) și (5.11)

$$G_i(c)_{w^i}^s \longrightarrow G_i(c)_{s^i}^s \quad \text{coincide cu} \quad G(c)_w^s \longrightarrow G(c)_s^s.$$

Prin urmare există o unică funcție

$$G(c)_{w, s}^o : \Sigma_{w, s} \longrightarrow (G(c)_w^s \longrightarrow G(c)_s^s)$$

astfel încât pentru orice $(i, w^i, s^i) \in |\mathcal{C}_{w, s}|$

$$g_{w^i, s^i}^i G(c)_{w, s}^o = G_i(c)_{w^i, s^i}^o \quad (5.12)$$

adică următoarea diagramă este comutativă:

$$\begin{array}{ccc}
G_{w,s}(i, w^i, s^i) = \Sigma_{w^i, s^i}^i & \xrightarrow{g_{w^i, s^i}^i} & \Sigma_{w,s} \\
& \searrow G_i(c)_{w^i, s^i}^o & \swarrow G(c)_{w,s}^o \\
& (G(c)_w^s \longrightarrow G(c)_s^s) &
\end{array}$$

Prin urmare pentru orice obiect $i \in |I|$ pentru orice $w^i \in S_i^*$ și orice $s^i \in S_i$

$$g_{w^i, s^i}^i; G(c)_{f_i^*(w^i), f_i(s^i)}^o = G_i(c)_{w^i, s^i}^o. \quad (5.13)$$

Pentru orice obiect $c \in |\mathcal{C}|$ prin definiție

$$G(c) = (G(c)^s, G(c)^o)$$

cu $G(c)^s$ și $G(c)^o$ definiți mai sus.

Pentru orice $c \in |\mathcal{C}|$ observăm că

$$G_i(c) = (G_i(c)^s, G_i(c)^o) \stackrel{(5.9)}{=} (f_i; G(c)^s, G_i(c)^o) \stackrel{(5.13)}{=} (f_i; G(c)^s, \{g_{w^i, s^i}^i; G(c)_{f_i^*(w^i), f_i(s^i)}^o\}_{w^i, s^i}),$$

adică pentru orice $c \in |\mathcal{C}|$ și orice $i \in |I|$

$$Alg(f_i, g^i)(G(c)) = G_i(c). \quad (5.14)$$

Construcția lui \mathbf{G} pe morfisme:

Fie $u : c \longrightarrow c'$ un morfism din \mathcal{C} .

Familia funcțiilor indexată de obiectele lui I

$$G_i(u) : S_i \longrightarrow \bigcup_{i \in |I|} \bigcup_{s^i \in S_i} (G_i(c)_{s^i}^s \longrightarrow G_i(c')_{s^i}^s)$$

formează în conformitate cu (5.8) un cocon pentru functorul FU , prin urmare există și este unică o funcție

$$G(u) : S \longrightarrow \bigcup (G_i(c)_{s^i}^s \longrightarrow G_i(c')_{s^i}^s)$$

astfel încât pentru orice $i \in |I|$

$$f_i; G(u) = G_i(u) \quad (5.15)$$

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, există $i \in |I|$ și $s_i \in S_i$, cu $f_i(s_i) = s$. Rezultă că

$$G(u)_s = G(u)_{f_i(s_i)} \stackrel{(5.15)}{=} G_i(u)_{s_i} : G_i(c)_{s_i}^s \longrightarrow G_i(c')_{s_i}^s.$$

Dar

$$G_i(c)_{s_i}^s \stackrel{(5.9)}{=} (f_i G(c)^s)_{s_i} = G(c)_{f_i(s_i)}^s = G(c)_s^s$$

și analog

$$G_i(c')_{s_i}^s = G(c')_s^s$$

deci pentru orice $s \in S$

$$G(u)_s : G(c)_s^s \longrightarrow G(c')_s^s.$$

G(u) este morfism de Σ -algebre

Arătăm că $\{G(u)_s\}_{s \in S}$ formează un morfism de la $G(c)$ în $G(c')$, pentru orice $u \in \mathcal{C}(c, c')$.

Fie $w \in S^*$, $s \in S$ și $\sigma \in \Sigma_{w,s}$. Deoarece familia

$$\{g_{w^i, s^i}^i\}_{(i, w^i, s^i) \in |\mathcal{C}_{w,s}|}$$

este epimorfă rezultă că există $(i, w^i, s^i) \in |\mathcal{C}_{w,s}|$ și $\sigma^i \in \Sigma_{w^i, s^i}^i$ astfel încât

$$\sigma = g_{w^i, s^i}^i(\sigma^i) \quad (5.16)$$

Deoarece $G_i(u) : G_i(c) \longrightarrow G_i(c')$ este morfism de Σ^i -algebre, diagrama de mai jos este comutativă:

$$\begin{array}{ccc} G_i(c)_{w^i}^s & \xrightarrow{G_i(c)_{w^i, s^i}^o(\sigma^i)} & G_i(c)_{s^i}^s \\ \downarrow G_i(u)_{w^i} & & \downarrow G_i(u)_{s^i} \\ G_i(c')_{w^i}^s & \xrightarrow{G_i(c')_{w^i, s^i}^o(\sigma^i)} & G_i(c')_{s^i}^s \end{array}$$

Dar $f_i(s^i) = s$ și egalitățile (5.10, 5.15) implică $G_i(c)_{s^i}^s = G(c)_s^s$, $G_i(c')_{s^i}^s = G(c')_s^s$ și $G_i(u)_{s^i} = G(u)_s$.

Analog $f_i^*(w^i) = w$ și egalitățile (5.11, 5.15) implică $G_i(c)_{w^i}^s = G(c)_w^s$, $G_i(c')_{w^i}^s = G(c')_w^s$ și $G_i(u)_{w^i} = G(u)_w$.

Din $f_i(s^i) = s$, $f_i^*(w^i) = w$ și 5.12 deducem că

$$G_i(c)_{w^i, s^i}^o(\sigma^i) = G(c)_{w, s}^o(\sigma) \quad \text{și} \quad G_i(c')_{w^i, s^i}^o(\sigma^i) = G(c')_{w, s}^o(\sigma)$$

prin urmare următoarea diagramă este comutativă:

$$\begin{array}{ccc} G(c)_w^s & \xrightarrow{G(c)_{w, s}^o(\sigma)} & G(c)_s^s \\ \downarrow G(u)_w & & \downarrow G(u)_s \\ G(c')_w^s & \xrightarrow{G(c')_{w, s}^o(\sigma)} & G(c')_s^s \end{array}$$

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 există $i \in |I|$ și $s_i \in S_i$ astfel încât $f_i(s_i) = s$. Din

$$\begin{aligned} G(1_c)_s &= G(1_c)_{f_i(s_i)} = (f_i; G(1_c))_{s_i} \stackrel{5.13}{=} (G_i(1_c))_{s_i} \stackrel{G_i \text{ functor}}{=} (1_{G_i(c)})_{s_i} \stackrel{5.9}{=} \\ 1_{(f_i G(c)^s)_{s_i}} &= 1_{G(c)_{f_i(s_i)}^s} = 1_{G(c)_s^s} = (1_{G(c)})_s \end{aligned}$$

$$G(1_c) = 1_{G(c)}.$$

Fie $u : c \longrightarrow c'$, $v : c' \longrightarrow c''$. Fie $s \in S$. Observăm că există $i \in |I|$ și există $s_i \in S_i$ astfel încât $f_i(s_i) = s$. Din

$$\begin{aligned} G(u)_s; G(v)_s &= G(u)_{f_i(s_i)}; G(v)_{f_i(s_i)} \stackrel{5.15}{=} G_i(u)_{s_i}; G_i(v)_{s_i} = (G_i(u); G_i(v))_{s_i} = G_i(u; v)_{s_i} \stackrel{5.15}{=} (f_i; G(u; v))_{s_i} = \\ &= G(u; v)_{f_i(s_i)} = G(u; v)_s. \end{aligned}$$

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 (5.14) o vom arăta numai pentru morfisme. Pentru $u : c \longrightarrow c'$ observăm că egalitatea (5.15) 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 și pe morfisme. Pentru $c \in |\mathcal{C}|$ notăm $H(c) = (H(c)^s, H(c)^o)$.

a) pe obiecte:

pentru $i \in |I|$ și $c \in |\mathcal{C}|$, observăm că

$$G_i(c) = Alg(f_i, g^i)(H(c)) = (f_i; H(c)^s, \{g_{w^i, s^i}^i; H(c)_{f_i^*(w^i), f_i(s^i)}^o\})$$

prin urmare $G_i(c)^s = f_i H(c)^s$ pentru orice $i \in |I|$, deci în conformitate cu unicitatea din (5.9) deducem $G(c)^s = H(c)^s$, adică algebrele $G(c)$ și $H(c)$ au aceleași suporturi.

Pentru $i \in |I|$, $c \in |\mathcal{C}|$, $w^i \in S_i^*$ și $s^i \in S_i$ din egalitatea de mai sus rezultă că

$$G_i(c)_{w^i, s^i}^o = g_{w^i, s^i}^i H(c)_{f_i^*(w^i), f_i(s^i)}^o.$$

Fie $w \in S^*$ și $s \in S$. Pentru orice $(i, w^i, s^i) \in |\mathcal{C}_{w, s}|$ deducem

$$G_i(c)_{w^i, s^i}^o = g_{w^i, s^i}^i H(c)_{w, s}^o,$$

deci, din unicitatea lui (5.12) rezultă că

$$G(c)_{w, s}^o = H(c)_{w, s}^o$$

adică algebrele $G(c)$ și $H(c)$ au aceleași operații.

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 lui (5.15), deducem $G(u) = H(u)$.

Deci functorii G și H coincid și pe morfisme. În concluzie $G = H$.

5.4 COLIMITE ÎN INSTITUȚII

Fie \mathcal{I} o instituție, $Pres(\mathcal{I})$ categoria prezentărilor lui \mathcal{I} . Functorul uituc

$$V : Pres(\mathcal{I}) \longrightarrow Sign$$

este definit prin

1. $V(\Sigma, E) = \Sigma$ pentru orice prezentare (Σ, E) .
2. $V(\phi) = \phi$ pentru orice morfism de prezentări ϕ .

Propoziție 5.4.1 Fie $\{(\Sigma_i, E_i)\}_{i \in I}$ o familie de prezentări și $\phi_i : \Sigma_i \longrightarrow \Sigma$ o familie de morfisme de semnături. 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) și pentru orice morfism de semnături $\psi : \Sigma \longrightarrow \Sigma'$ 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.

Demonstrație: 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. \square

Propoziție 5.4.2 Functorul uituc $V : Pres(\mathcal{I}) \longrightarrow Sign$ crează slab colimitele.

Demonstrație: 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 $F; V$ și $V(F(i)) \xrightarrow{\phi_i} \Sigma$ o colimită a functorului $F; V : I \longrightarrow Sign$ există un unic morfism de semnături $\psi : \Sigma \longrightarrow \Sigma'$ cu proprietate $\phi_i \psi = \rho_i$ pentru orice obiect i din I . Deoarece $\phi_i \psi = \rho_i$ sunt morfisme de prezentări, din propoziția precedentă rezultă că ψ este morfism de prezentări. \square

Corolar 5.4.3 *Într-o instituție în care categoria signaturilor are colimite, categoria prezentărilor are colimite și functorul V comută cu ele.*

Fie $Th(\mathcal{I})$ categoria teoriilor instituției \mathcal{I} . Functorul uituc

$$U : Th(\mathcal{I}) \longrightarrow Sign$$

este prin definiție restricția lui V la categoria teoriilor lui \mathcal{I} .

Propoziție 5.4.4 *Fie $\{(\Sigma_i, E_i)\}_{i \in I}$ o familie de teorii și $\phi_i : \Sigma_i \longrightarrow \Sigma$ o familie de morfisme de semnături. 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) și pentru orice morfism de semnături $\psi : \Sigma \longrightarrow \Sigma'$ 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.

Demonstrație: Deoarece

$$\phi_i \psi : (\Sigma_i, E_i) \longrightarrow (\Sigma', E)$$

sunt morfisme de teorii, $Sen(\phi_i \psi)(E_i) \subseteq E$ pentru orice $i \in I$. Prin urmare

$$[\bigcup_{i \in I} Sen(\phi_i \psi)(E_i)]^\bullet \subseteq E$$

deoarece E este închisă. Deoarece 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. \square

Propoziție 5.4.5 *Functorul uituc $U : Th(\mathcal{I}) \longrightarrow Sign$ crează slab colimitele.*

Demonstrație: 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 teorii.

Deoarece $V(F(i)) \xrightarrow{\phi_i} \Sigma$ este o colimită a functorului $F; V : I \longrightarrow \text{Sign}$ deducem că

$$F(i) \xrightarrow{\phi_i} (\Sigma, [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i)]^\bullet)$$

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 $F; U$ și $U(F(i)) \xrightarrow{\phi_i} \Sigma$ o colimită a functorului $F; U : I \longrightarrow \text{Sign}$ există un unic morfism de semnături $\psi : \Sigma \longrightarrow \Sigma'$ cu proprietate $\phi_i; \psi = \rho_i$ pentru orice obiect i din I . Deoarece $\phi_i; \psi = \rho_i$ sunt morfisme de teorii, din propoziția precedentă rezultă că ψ este morfism de teorii. \square

Corolar 5.4.6 *Într-o instituție în care categoria semnăturilor are colimite, categoria teoriilor are colimite și functorul U comută cu ele.*

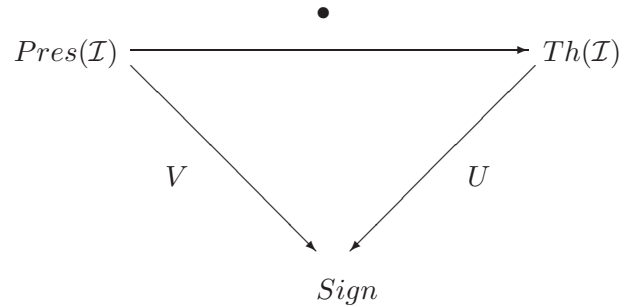
Din teoremele de mai sus rezultă că într-o instituție în care categoria semnăturilor are colimite, atât categoria prezentărilor ei cât și categoria teoriilor ei au colimite. Mai mult propozițiile de mai sus dau și o construcție a acestora.

5.4.1 Functorul \bullet

Observația 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.

Demonstrație: Din lema închiderii deducem $\text{Sen}(\psi)(E^\bullet) \subseteq \text{Sen}(\psi)(E)^\bullet$. Deoarece ψ este morfism de prezentări $\text{Sen}(\psi)(E) \subseteq E_1^\bullet$ prin urmare $\text{Sen}(\psi)(E)^\bullet \subseteq E_1^\bullet$ deci $\text{Sen}(\psi)(E^\bullet) \subseteq E_1^\bullet$, adică $\psi : (\Sigma, E^\bullet) \longrightarrow (\Sigma_1, E_1^\bullet)$ este morfism de teorii.



Functorul $\bullet : \text{Pres}(\mathcal{I}) \longrightarrow \text{Th}(\mathcal{I})$ 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 $\bullet; U = V$.

Observația 5.4.8 *Dacă $\phi_i : \Sigma_i \longrightarrow \Sigma$ este o familie de morfisme de semnături și $E_i \subseteq \text{Sen}(\Sigma_i)$ pentru orice indice $i \in I$, atunci*

$$[\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i)]^\bullet = [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i^\bullet)]^\bullet$$

Demonstrație: *Demonstrația prin dublă incluziune începe prin a observa că incluziunea*

$$[\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i)]^\bullet \subseteq [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i^\bullet)]^\bullet$$

rezultă ușor plecând de la $E_i \subseteq E_i^\bullet$ și observând că toate construcțiile folosite sunt crescătoare.

Reciproc, utilizând lema incluziunii

$$\text{Sen}(\phi_i)(E_i^\bullet) \subseteq [\text{Sen}(\phi_i)(E_i)]^\bullet$$

deducem succesiv

$$\begin{aligned} \text{Sen}(\phi_i)(E_i^\bullet) &\subseteq [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i)]^\bullet, \\ \bigcup_{i \in I} \text{Sen}(\phi_i)(E_i^\bullet) &\subseteq [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i)]^\bullet \text{ și} \\ [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i^\bullet)]^\bullet &\subseteq [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i)]^\bullet. \quad \square \end{aligned}$$

Propoziție 5.4.9 În orice instituție functorul \bullet conservă colimitele.

Demonstrație: Fie $F : I \longrightarrow \text{Pres}(\mathcal{I})$ 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 observația 5.1.13 și 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 propoziția 5.4.2 în colimita functorului $F; \bullet$ construită conform propoziției 5.4.5.

Fie $V(F(i)) \xrightarrow{\phi_i} \Sigma$ o colimită a functorului $F; V : I \longrightarrow \text{Sign}$. Conform construcției din propoziția 5.4.2

$$\phi_i : (\Sigma^i, E_i) \longrightarrow (\Sigma, \bigcup_{i \in I} \text{Sen}(\phi_i)(E_i)) \quad (5.17)$$

este o colimită a functorului F și deoarece $(F; \bullet)(i) = (\Sigma^i, E_i^\bullet)$ pentru orice obiect i din I , conform construcției din propoziția 5.4.5

$$\phi_i : (\Sigma^i, E_i) \longrightarrow (\Sigma, [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i^\bullet)]^\bullet) \quad (5.18)$$

este o colimită a functorului $F; \bullet$.

Egalitatea demonstrată mai sus

$$[\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i)]^\bullet = [\bigcup_{i \in I} \text{Sen}(\phi_i)(E_i^\bullet)]^\bullet$$

arată că aplicând functorul \bullet colimitei 5.17 a functorului F se obține colimita 5.18 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 convinși că lucrurile sunt corecte este necesar să arătăm următoarele: colimita prezentărilor reprezintă chiar teoria colimită a teoriilor reprezentate de prezentările care au intrat în calcul. Ultima propoziție ne asigură de acest fapt.

5.5 CATEGORII INCLUSIVE

Conceptul de incluziune abstractă a fost introdus de [?] în [?], unde definiția categoriei inclusive cere mai multe condiții. Cu actuala definiție sub numele de categorie slab inclusivă a fost introdus de [?] în [?]. Definiția utilizată aici coincide cu cea din cartea lui Răzvan Diaconescu.

Definiția 5.5.1 O categorie \mathcal{I} este numită **categorie de incluziuni** dacă pentru orice pereche de obiecte A și B

- 1) $\mathcal{I}(A, B)$ are cel mult un element și
- 2) $\mathcal{I}(A, B) \neq \emptyset$ și $\mathcal{I}(B, A) \neq \emptyset$ 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 parțial ordonată organizată ca o categorie.

Pentru incluziuni $i \in \mathcal{I}(A, B)$ mai utilizăm și notația $i : A \hookrightarrow B$.

Definiția 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 aceleași obiecte ca și \mathcal{C} .

Definiția 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 aceleași obiecte ca și \mathcal{C} și
3. Orice morfism f din \mathcal{C} poate fi factorizat unic $f = e_f; i_f$ unde $e_f \in \mathcal{E}$ și $i_f \in \mathcal{I}$.

O categorie împreună cu un sistem de incluziuni se numește **categorie inclusivă**. \square

Fie $f : A \longrightarrow B$ un morfism dintr-o categorie inclusivă. Imaginea lui f notată cu $f(A)$ este prin definiție codomeniul morfismului $e_f : A \longrightarrow f(A)$ același cu domeniul morfismului $i_f : f(A) \hookrightarrow B$.

Continuăm cu mai multe exemple.

Set

Considerăm categoria mulțimilor **Set** cu sistemul de incluziuni (**Incluziuni, Surjecții**). Prin incluziune de mulțimi se înțelege unica funcție de la o submulțime la o supramulțime a sa care duce orice element în el însuși. Subcategoria incluziunilor este o latice unde *supremum* e reuniunea și *infimum* este intersecția.

Este binecunoscut că orice funcție $f : A \longrightarrow B$ se poate scrie ca o compunere $f = e_f; i_f$ unde $i_f : f(A) \hookrightarrow B$ este incluziune și surjecția $e_f : A \longrightarrow f(A)$ 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 surjecție $e : A \longrightarrow C$ și o incluziune $i : C \hookrightarrow B$. Evident $C \subseteq B$. Observăm că pentru orice $a \in A$, $e(a) = i(e(a)) = f(a) = e_f(a)$, prin urmare $f(A) \subseteq C$. Probăm incluziunea contrară. Fie $c \in C$. Deoarece e este surjectivă există $a \in A$ astfel încât $c = e(a)$. Deducem $c = f(a)$, deci $c \in f(A)$. Deoarece $f(A) = C$ deducem ușor că $e = e_f$ și $i = i_f$.

Sig

Definiția 5.5.4 Fie două semnături $(S, \Sigma), (S', \Sigma') \in |\mathbf{Sig}|$. Definim incluziunea $(S, \Sigma) \subseteq (S', \Sigma')$ dacă și numai dacă $S \subseteq S'$ și $\Sigma_{w,s} \subseteq \Sigma'_{w,s}$ pentru orice $w \in S^*, s \in S$.

Cu relația astfel definită semnăturile algebrice formează o clasă parțial ordonată.

Incluziunile lui **Sig** sunt definite prin morfismele care au toate componentele incluziuni de mulțimi. Notăm cu \mathcal{I} categoria incluziunilor de semnături. Ea este o subcategorie de incluziuni a lui **Sig**.

Evident că pentru orice pereche de semnături $(S, \Sigma), (S', \Sigma')$ și pentru orice pereche de incluziuni $(f_1, g_1), (f_2, g_2)$ din $\mathcal{I}((S, \Sigma), (S', \Sigma'))$ deoarece funcțiile $f_1, f_2 : S \hookrightarrow S'$, sunt incluziuni rezultă că $f_1 = f_2$. De asemenea, pentru orice $w \in S^*$ și $s \in S$ funcțiile $g_{1w,s}, g_{2w,s} : \Sigma_{w,s} \hookrightarrow \Sigma'_{f^*(w), f(s)}$, unde $f = f_1$, sunt incluziuni. Din faptul că $g_{1w,s}$ și $g_{2w,s}$ sunt incluziuni rezultă $g_{1w,s} = g_{2w,s}$. Deci $(f_1, g_1) = (f_2, g_2)$.

Mai observăm că $(S, \Sigma) \subseteq (S', \Sigma')$ și $(S', \Sigma') \subseteq (S, \Sigma)$ implică $(S, \Sigma) = (S', \Sigma')$. Prin urmare, \mathcal{I} este subcategorie de incluziuni a lui **Sig**.

Definiția 5.5.5 Un morfism de semnături $(f, g) : (S', \Sigma') \longrightarrow (S, \Sigma)$ se numește *surjecție abstractă de semnături* dacă f este o funcție surjectivă și pentru orice $w \in S^*, s \in S$ și $\sigma \in \Sigma_{w,s}$ există $w' \in S'^*, s' \in S'$ și $\sigma' \in \Sigma'_{w',s'}$ cu proprietățile $f^*(w') = w$, $f(s') = s$ și $g_{w',s'}(\sigma') = \sigma$. Notăm cu \mathcal{E} categoria surjecțiilor abstracte de semnături.

Observația 5.5.6 \mathcal{E} este subcategorie a lui **Sig** care are aceleași obiecte ca și **Sig**.

Demonstrație: Fie $(f', g') : (S'', \Sigma'') \longrightarrow (S', \Sigma')$ și $(f, g) : (S', \Sigma') \longrightarrow (S, \Sigma)$ două surjecții abstracte de semnături. Vom arăta că și compunerea lor este o surjecție abstractă de semnături.

Deoarece f' și f sunt surjecții rezultă că $f'; f$ este surjecție.

Fie $w \in S^*, s \in S$ și $\sigma \in \Sigma_{w,s}$. Deoarece $(f, g) : (S', \Sigma') \longrightarrow (S, \Sigma)$ este surjecție abstractă de semnături există $w' \in S'^*, s' \in S'$ și $\sigma' \in \Sigma'_{w',s'}$ cu proprietățile $f^*(w') = w$, $f(s') = s$ și $g_{w',s'}(\sigma') = \sigma$. Deoarece $(f', g') : (S'', \Sigma'') \longrightarrow (S', \Sigma')$ este surjecție abstractă de semnături există $w'' \in S''^*, s'' \in S''$ și $\sigma'' \in \Sigma''_{w'',s''}$ cu proprietățile $f'^*(w'') = w'$, $f'(s'') = s'$ și $g'_{w'',s''}(\sigma'') = \sigma'$. Rezultă că $(f'; f)^*(w'') = f^*(f'^*(w'')) = f^*(w') = w$, $(f'; f)(s'') = f(f'(s'')) = f(s') = s$ și $(g'; g)_{w'',s''}(\sigma'') = (g'_{w'',s''}; g_{f^*(w''), f'(s'')})(\sigma'') = g_{w',s'}(g'_{w'',s''}(\sigma'')) = g_{w',s'}(\sigma') = \sigma$.

Deci $(f', g'); (f, g)$ este surjecție abstractă de semnături.

Propoziția 5.5.7 În categoria semnăturilor algebrice **Sig** $\langle \mathcal{I}, \mathcal{E} \rangle$ este sistem inclusiv.

Demonstrație: Demonstrăm că orice morfism de semnături (f, g) poate fi scris unic ca o compunere între o surjecție abstractă de semnături (e_f, e_g) din \mathcal{E} și o incluziune (i_f, i_g) din \mathcal{I} . Căutăm o descompunere de tipul:

$$\begin{array}{ccc}
 (S, \Sigma) & \xrightarrow{(f, g)} & (S', \Sigma') \\
 \searrow (e_f, e_g) & & \nearrow (i_f, i_g) \\
 & (f(S), \{\Sigma_{w,s}^0\}_{w \in f(S)^*, s \in f(S)}) &
 \end{array}$$

Demonstrăm existența descompunerii.

În categoria **Set**, pentru orice funcție f există o unică descompunere cu e_f surjecție și i_f incluziune astfel încât $f = e_f; i_f$ de forma :

$$\begin{array}{ccc} S & \xrightarrow{f} & S' \\ & \searrow e_f & \nearrow i_f \\ & f(S) & \end{array}$$

Știm că $e_f(x) = f(x)$ pentru orice $x \in S$.

Pentru orice $w' \in f^*(S)$ și $s' \in f(S)$ definim

$$\Sigma_{w',s'}^0 = \bigcup \{g_{w,s}(\Sigma_{w,s}) \mid w \in (f^*)^{-1}(w') \text{ și } s \in f^{-1}(s')\} \subseteq \Sigma_{w',s'}^0.$$

Pentru orice $w \in S^*, s \in S$ definim

$$e_{g_{w,s}} : \Sigma_{w,s} \longrightarrow \Sigma_{e_f(w),e_f(s)}^0$$

astfel $e_{g_{w,s}}(\sigma) = g_{w,s}(\sigma) \in \Sigma_{f^*(w),f(s)}^0$, pentru fiecare $\sigma \in \Sigma_{w,s}$.

Observăm că $(e_f, e_g) \in \mathcal{E}$, $(i_f, i_g) \in \mathcal{I}$ și $(f, g) = (e_f, e_g); (i_f, i_g)$.

Arătăm unicitatea acestei descompuneri.

Presupunem că există $(f', g') \in \mathcal{E}$ și o incluziune (i'_f, i'_g) de semnături în \mathcal{I} astfel încât diagrama următoare să fie comutativă:

$$\begin{array}{ccc} (S, \Sigma) & \xrightarrow{(f,g)} & (S', \Sigma') \\ & \searrow (f',g') & \nearrow (i'_f,i'_g) \\ & (S'', \Sigma'') & \end{array}$$

Dar, în categoria mulțimilor **Set**, e_f și f' sunt ambele surjecții. Cum o astfel de descompunere $f = f'; i'_f = e_f; i_f$ este unică în **Set** înseamnă că $f' = e_f$, $i'_f = i_f$ și $S'' = f(S)$. Pentru orice $w \in S^*, s \in S$ și $\sigma \in \Sigma_{w,s}$ observăm că

$$g'_{w,s}(\sigma) = (i'_g)_{f'^*(w),f'(s)}(g'_{w,s}(\sigma)) = (g'_{w,s}; (i'_g)_{f'^*(w),f'(s)})(\sigma) = g_{w,s}(\sigma).$$

Deoarece (f', g') este surjecție abstractă pentru orice $w' \in f^*(S)$ și $s' \in f(S)$

$$\Sigma_{w',s'}'' = \bigcup \{g'_{w,s}(\Sigma_{w,s}) \mid s \in f^{-1}(s'), w \in (f^*)^{-1}(w')\} =$$

$$\bigcup \{g_{w,s}(\Sigma_{w,s}) \mid s \in f^{-1}(s'), w \in (f^*)^{-1}(w')\} = \Sigma_{w',s'}^0.$$

Deci $g' = e_g$ și $\Sigma'' = \Sigma^0$.

Deci **Sig** cu sistemul $\langle \mathcal{I}, \mathcal{E} \rangle$ este categorie inclusivă. \square

Teoriile unei instituții

Fie o instituție \mathcal{I} . Fie $U : Th(\mathcal{I}) \longrightarrow Sign$ functorul uituc definit prin $U(\Sigma, E) = \Sigma$ și $U(\varphi) = \varphi$.

Reamintim că o prezentare este o pereche (S, Σ) unde $\Sigma \in |Sign|$ și $E \subseteq Sen(\Sigma)$ și o teorie este o prezentare (S, Σ) cu $E^\bullet = E$.

Teorema 5.5.8 *Dacă $Sign$ este o categorie inclusivă, atunci $Th(\mathcal{I})$ este o categorie inclusivă.*

Demonstrație: Fie (I, \mathcal{E}) un sistem de incluziuni în categoria signaturilor $Sign$. Vom defini $(\mathcal{I}_T, \mathcal{E}_T)$, un sistem de incluziuni în categoria teoriilor după cum urmează.

Fie $\psi : (\Sigma, E) \longrightarrow (\Sigma', E')$ un morfism de teorii.

- 1) $\psi \in \mathcal{I}_T$ dacă și numai dacă $\psi \in I$
- 2) $\psi \in \mathcal{E}_T$ dacă și numai dacă $\psi \in \mathcal{E}$ și $E' = Sen(\psi)(E)^\bullet$.

Probăm că \mathcal{I}_T este o categorie de incluziuni.

- 1) Deoarece

$$\mathcal{I}_T((\Sigma, E)(\Sigma', E')) \subseteq I(\Sigma, \Sigma')$$

și $I(\Sigma, \Sigma')$ are cel mult un element deducem că $\mathcal{I}_T((\Sigma, E)(\Sigma', E'))$ are cel mult un element.

- 2) Fie $\psi \in \mathcal{I}((\Sigma, E)(\Sigma', E'))$ și $\varphi \in \mathcal{I}((\Sigma, E)(\Sigma', E'))$. Deducem $I(\Sigma, \Sigma') \neq \emptyset$ și $I(\Sigma, \Sigma') \neq \emptyset$, deci $\Sigma = \Sigma'$. Din $\psi \in \mathcal{I}(\Sigma, \Sigma)$ și $\varphi \in \mathcal{I}(\Sigma, \Sigma)$ deducem $\psi = \varphi = 1_\Sigma$. Deoarece $1_\Sigma : (\Sigma, E) \longrightarrow (\Sigma, E')$ și $1_\Sigma : (\Sigma, E') \longrightarrow (\Sigma, E)$ 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}_T rezultă că \mathcal{I}_T are aceleași obiecte ca și $Th(\mathcal{I})$ deci \mathcal{I}_T este o subcategorie de incluziuni a lui $Th(\mathcal{I})$.

Deoarece pentru orice teorie (Σ, E) , $1_\Sigma : (\Sigma, E) \longrightarrow (\Sigma, E)$ este morfism de teorii din \mathcal{E}_T rezultă că \mathcal{E}_T are aceleași obiecte ca și $Th(\mathcal{I})$.

Probăm existența descompunerii. Fie $\psi : (\Sigma, E) \longrightarrow (\Sigma', E')$ un morfism de teorii. Descompunem ψ în categoria inclusivă a signaturilor

$$\begin{array}{ccc} \Sigma & \xrightarrow{\psi} & \Sigma' \\ & \searrow e_\psi & \nearrow i_\psi \\ & \psi(\Sigma) & \end{array}$$

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}_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 $\mathcal{I}_{\mathcal{T}}$.

$$\begin{array}{ccc} (\Sigma, E) & \xrightarrow{\psi} & (\Sigma', E') \\ & \searrow e_{\psi} \quad \nearrow i_{\psi} & \\ & (\psi(\Sigma), (Sen(e_{\psi})(E))^{\bullet}) & \end{array}$$

Probăm unicitatea descompunerii. Fie $\psi = e; i$ o altă descompunere unde e este din $\mathcal{E}_{\mathcal{T}}$ și i este din $\mathcal{I}_{\mathcal{T}}$.

$$\begin{array}{ccc} (\Sigma, E) & \xrightarrow{\psi} & (\Sigma', E') \\ & \searrow e \quad \nearrow i & \\ & (\Sigma_1, (Sen(e)(E))^{\bullet}) & \end{array}$$

Deoarece descompunerea $\psi = e; i$ are loc și în categoria inclusivă a signaturilor deducem $\Sigma_1 = \psi(\Sigma)$, $e = e_{\psi}$ și $i = i_{\psi}$ deci descompunerea este unică.

În concluzie $Th(\mathcal{I})$ este o categorie inclusivă. \square

Corolar 5.5.9 *Teoriile instituției logicii ecuaționale formează o categorie inclusivă.*

5.5.1 Categorii inclusive cu sume

Definiția 5.5.10 Spunem că o categorie inclusivă \mathcal{C} are sume dacă subcategoria incluziunilor ei \mathcal{I} are supremuri finite.

Reamintim că într-o categorie de incluziuni supremurile coincid cu sumele directe. Continuăm cu mai multe exemple de categorii inclusive cu sumă.

Set

Am văzut că în categoria mulțimilor sumele directe sunt reuniunile disjuncte.

În categoria incluziunilor de mulțimi sumele directe sunt reuniunile. De fapt având de a face cu o clasă parțial ordonată, sumele directe coincid cu supremurile care în cazul mulțimilor coincid cu reuniunile.

Sig

Propoziție 5.5.11 *Categoria inclusivă a signaturilor algebrice are sume.*

Demonstrație:

Vom demonstra chiar mai mult, existența supremurilor arbitrare.

Considerăm K o mulțime arbitrară de indecși. Pentru $k \in K$ fie (S_k, Σ^k) o semnătură algebrică. Semnătura (S, Σ) este definită prin $S = \bigcup \{S_k | k \in K\}$ și pentru orice $w \in S^*$ și $s \in S$ prin

$$\Sigma_{w,s} = \bigcup \{\Sigma_{w,s}^k | w \in S_k^*, s \in S_k\}.$$

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 $w \in S^*$, $s \in S$ și $\sigma \in \Sigma_{w,s}$. Prin urmare există $k \in K$, $w \in S_k^*$ și $s \in S_k$ cu proprietatea $\sigma \in \Sigma_{w,s}^k$. 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) . \square

Teoriile unei instituții

Teorema 5.5.12 *Dacă Sign este o categorie inclusivă cu sume, atunci $\text{Th}(\mathcal{I})$ este o categorie inclusivă cu sume.*

Demonstrație: Păstrăm notațiile din propoziția precedentă.

Fie (Σ_k, E_k) o familie finită de teorii. Fie Σ supremul (suma directă) familiei $\{\Sigma_k\}_k$ și $i_k : \Sigma_k \hookrightarrow \Sigma$ incluziuni. Vom proba că

$$(\Sigma, (\bigcup_k \text{Sen}(i_k)(E_k))^\bullet)$$

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 și $j_k \in \mathcal{I}_{\mathcal{T}}((\Sigma_k, E_k), (\Sigma_1, E_1))$. Deoarece Σ este supremul familiei de semnături $\{\Sigma_k\}_k$, există o unică incluziune de semnături $j : \Sigma \hookrightarrow \Sigma_1$ cu proprietatea $i_k; j = j_k$ pentru orice k .

Deoarece familia

$$i_k : (\Sigma_k, E_k) \longrightarrow (\Sigma, (\bigcup_k \text{Sen}(i_k)(E_k))^\bullet)$$

are proprietatea din propoziția 5.4.4 și $i_k; j$ sunt morfisme de teorii rezultă că

$$j : (\Sigma, (\bigcup_k \text{Sen}(i_k)(E_k))^\bullet) \longrightarrow (\Sigma_1, E_1)$$

este morfism de teorii. \square

Deoarece categoria signaturilor algebrice Sig este inclusivă cu sume rezultă corolarul următor.

Corolar 5.5.13 *Categoria teoriilor din instituția logicii ecuaționale este inclusivă cu sume.*

Din punct de vedere practic, deoarece teoriile sunt implementate prin prezentări ale lor, este utilă următoarea propoziție.

Propoziție 5.5.14 *Fie (Σ_k, E_k) o familie finită de prezentări și $i_k : \Sigma_k \longrightarrow \Sigma$ supremul în categoria incluziunilor de semnături. Supremul familiei de teorii (Σ_k, E_k^\bullet) este prezentat de*

$$(\Sigma, \bigcup_k \text{Sen}(i_k)(E_k)).$$

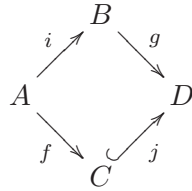
Demonstrație: Este suficient să invocăm observația 5.4.8

$$(\bigcup_k \text{Sen}(i_k)(E_k))^\bullet = (\bigcup_k \text{Sen}(i_k)(E_k^\bullet))^\bullet.$$

5.5.2 Sume fibratate care conservă incluziunile

Definiția 5.5.15 Într-o categorie inclusivă, spunem că sumele fibratate conservă incluziunile dacă:

pentru orice incluziune $i : A \longrightarrow B$ și 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 $g : B \longrightarrow D$ și o incluziune $j : C \longrightarrow D$.

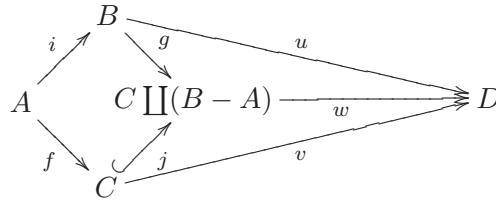


Continuăm cu exemple de categorii inclusive în care sumele fibratate conservă incluziunile.

Set

Propoziție 5.5.16 În categoria mulțimilor sumele fibratate conservă incluziunile.

Demonstrație: Fie $f : A \longrightarrow C$ o funcție și $A \subseteq B$. Notăm cu $i : A \hookrightarrow B$ incluziunea lui A în B .



Definim pe g 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 $u : B \longrightarrow D$ și $v : C \longrightarrow D$ 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 $y; w = u$. Deoarece pentru orice $x \in C$, $w(j(x)) = w(x) = v(x)$, deducem $j; v = w$.

Unicitatea lui w rezultă din construcție. \square

Sig

Propoziție 5.5.17 *În categoria signaturilor algebrice sumele fibrat conservă incluziunile.*

Demonstrație: Fie $i : (S, \Sigma) \longrightarrow (S', \Sigma')$ o incluziune de signaturi și $(f, g) : (S, \Sigma) \longrightarrow (S'', \Sigma'')$ un morfism de signaturi.

Plecând de la incluziunea de mulțimi $S \hookrightarrow S'$ și funcția $f : S \longrightarrow S''$ facem în categoria mulțimilor, după modelul de mai sus, o sumă fibrată care conservă incluziunea.

Definim f' astfel $f'(s) = \begin{cases} f(s) & s \in S \\ s & \text{altfel} \end{cases}$. Observăm că $f'|_S = f; l$.

Variante de definire Σ^0 :

1) Dacă $w^0 \in S''^*$ și $s^0 \in S''$ atunci

$$\Sigma_{w^0, s^0}^0 = \Sigma_{w^0, s^0}'' \cup \{(w', s', \sigma') | f'(w') = w^0, f'(s') = s^0, \sigma' \in \Sigma_{w', s'}' - \Sigma_{w', s'}\}.$$

2) În caz contrar

$$\Sigma_{w^0, s^0}^0 = \{(w', s', \sigma') | f'(w') = w^0, f'(s') = s^0, \sigma' \in \Sigma_{w', s'}' - \Sigma_{w', s'}\}.$$

Vom defini g' 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 $g'_{w', s'} : \Sigma_{w', s'}' \longrightarrow \Sigma_{f'(w'), f'(s')}^0$. Comutativitatea diagramei $i; (f', g') = (f, g); l$ este evidentă.

$$\begin{array}{ccccc} & (S', \Sigma') & & & \\ & \nearrow i & \searrow (f', g') & \searrow (f_1, g^1) & \\ (S, \Sigma) & & (S'' \amalg (S' - S), \Sigma^0) & \xrightarrow{\exists!(u, v)} & (S^1, \Sigma^1) \\ & \searrow (f, g) & \nearrow l & \searrow (f_2, g^2) & \\ & (S'', \Sigma'') & & & \end{array}$$

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 $f_1|_S = f; f_2$ și pentru orice $w \in S^*$, $s \in S$ și $\sigma \in \Sigma_{w, s}$

$$g_{w, s}^1(\sigma) = g_{f^*(w), f(s)}^2(g_{w, s}(\sigma)).$$

Construim morfismul de signaturi $(u, v) : (S'' \amalg (S' - S), \Sigma^0) \longrightarrow (S^1, \Sigma^1)$. Din proprietatea sumei fibrat pentru mulțimi știm că există o unică funcție u care face diagramele comutative pe mulțimi, adică $f'; u = f_1$ și $u|_{S''} = f_2$.

Definim pe $v_{w^0, s^0} : \Sigma_{w^0, s^0}^0 \longrightarrow \Sigma_{u(w^0), u(s^0)}^1$ astfel

1) Dacă $w^0 \in S''^*$, $s^0 \in S''$, $\sigma \in \Sigma_{w^0, s^0}''$ atunci $v_{w^0, s^0}(\sigma) = g_{w^0, s^0}^2(\sigma)$.

2) Dacă $\sigma \in \Sigma_{w', s'}' - \Sigma$ și $f'^*(w') = w^0$, $f'(s') = s^0$ atunci $v_{w^0, s^0}(w', s', \sigma') = g_{w', s'}^1(\sigma')$.

Din primul caz este evident că $v|_{\Sigma''} = g^2$, 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 $w' \in S'^*$ și $s \in S'$ că

$$g'_{w', s'}; v_{f^*(w'), f(s)} = g_{w', s'}^1.$$

Fie $\sigma \in \Sigma'_{w',s'}$. Conform definiției lui $g'_{w',s'}$ considerăm două cazuri

- a) În cazul în care $\sigma \in \Sigma'_{w',s'} - \Sigma$ avem $v_{f'(w'),f'(s')}(g'_{w',s'}(\sigma)) = v_{f'(w),f'(s')}(w',s',\sigma) = g_{w',s'}^1(\sigma)$.
- b) Dacă $\sigma \in \Sigma_{w,s}$, $w \in S^*$, $s \in S$ avem $v_{f'(w),f'(s)}(g'_{w,s}(\sigma)) = v_{f(w),f(s)}(g_{w,s}(\sigma)) = g_{f(w),f(s)}^2(g_{w,s}(\sigma)) = g_{w,s}^1(\sigma)$.

Unicitatea rezultă ușor. \square

Teoriile unei instituții

Propoziție 5.5.18 *Fie \mathcal{I} o instituție în care $Sign$ este categorie inclusivă. Dacă în categoria $Sign$ sumele fibrare conservă incluziunile, atunci în categoria $Th(\mathcal{I})$ sumele fibrare conservă incluziunile.*

Demonstrație: În $Th(\mathcal{I})$ presupunem că incluziunea $i : (\Sigma, E) \longrightarrow (\Sigma_1, E_1)$ și morfismul $\phi : (\Sigma, E) \longrightarrow (\Sigma_2, E_2)$ au o sumă fibrată. Functorul $U : Th(\mathcal{I}) \longrightarrow Sign$ conservă, conform propoziției 5.4.6 colimitele, prin urmare în $Sign$ incluziunea $i : \Sigma \longrightarrow \Sigma_1$ și morfismul $\phi : \Sigma \longrightarrow \Sigma_2$ au o sumă fibrată. Deoarece în categoria signaturilor sumele fibrare conservă incluziunile deducem existența unei sume fibrare

$$\begin{array}{ccc} & \Sigma_1 & \\ i \nearrow & & \searrow \rho \\ \Sigma & & \Sigma' \\ \phi \searrow & \hookrightarrow & \nearrow j \\ & \Sigma_2 & \end{array}$$

în care j este o incluziune de semnături.

Deoarece functorul $U : Th(\mathcal{I}) \longrightarrow Sign$ crează slab colimitele, există suma fibrată

$$\begin{array}{ccc} & (\Sigma_1, E_1) & \\ i \nearrow & & \searrow \rho \\ (\Sigma, E) & & (\Sigma', (Sen(\rho)(E_1) \cup Sen(j)(E_2))^{\bullet}) \\ \phi \searrow & \hookrightarrow & \nearrow j \\ & (\Sigma_2, E_2) & \end{array}$$

în care j este o incluziune de teorii. \square

Corolar 5.5.19 *În categoria teoriilor din instituția logicii ecuaționale sumele fibrare conservă incluziunile.*

5.6 IMPORTURI

Pentru o prezentare cele mai utilizate semantici sunt semantica inițială și semantica slabă. Conform semanticii slabe sunt considerate toate modele prezentării. Conform semanticii inițiale este considerat doar modelul inițial. Prin urmare în conformitate cu semantica adoptată o prezentare poate da mai multe specificații, adică o prezentare și o clasă de modele ale acesteia. Practica programării cere și alte metode de definire a specificațiilor. Printre acestea metoda importurilor.

Pentru definirea semanticii inițiale este necesar conceptul de săgeată universală.

Definiția 5.6.1 Fie $F : \mathcal{C} \longrightarrow \mathcal{D}$ un functor, C un obiect din categoria \mathcal{C} și D un obiect din categoria \mathcal{D} .

Un morfism $u : D \longrightarrow F(C)$ se numește **săgeată 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** și **using**.

Fie o specificație (Σ, E, \mathcal{M}) importată de prezentarea (Σ', E') . Acceptăm automat că $\mathcal{M} \subseteq \text{Mod}^p(\Sigma, E)$ și că prezentarea importată este inclusă în specificația care importă:

$$\alpha : (\Sigma, E) \hookrightarrow (\Sigma', E').$$

Prin urmare $\text{Mod}^p(\alpha) : \text{Mod}^p(\Sigma', E') \longrightarrow \text{Mod}^p(\Sigma, E)$ este un functor uituc, adică uită o parte dintre suporturi și o parte dintre operații.

Vom defini specificația $(\Sigma', E', \mathcal{M}')$ definind de fapt clasa de modele \mathcal{M}' în funcție de tipul importului și de semantica acceptată pentru prezentarea importatoare.

5.6.1 Semantică slabă

Prin definiție în funcție de tipul importului:

a) **Protecting**

$$\mathcal{M}' = \{M' \in \text{Mod}^p(\Sigma', E') : \text{Mod}^p(\alpha)(M') \in \mathcal{M}\}$$

b) **Extending**

$$\mathcal{M}' = \{M' \in \text{Mod}^p(\Sigma', E') : (\exists M \in \mathcal{M}) \text{ și un morfism injectiv } j : M \longrightarrow \text{Mod}^p(\alpha)(M')\}$$

c) **Using**

$$\mathcal{M}' = \{M' \in \text{Mod}^p(\Sigma', E') : (\exists M \in \mathcal{M}) \text{ și un morfism } j : M \longrightarrow \text{Mod}^p(\alpha)(M')\}$$

5.6.2 Semantica inițială

a) **Protecting**

$$\mathcal{M}' = \{M' \in \text{Mod}^p(\Sigma', E') : \text{Mod}^p(\alpha)(M') \in \mathcal{M} \text{ și } 1_{\text{Mod}^p(\alpha)(M')} \text{ este săgeată universală}\}$$

b) **Extending**

$$\mathcal{M}' = \{M' \in \text{Mod}^p(\Sigma', E') : (\exists M \in \mathcal{M}) \text{ și un morfism injectiv și universal } j : M \longrightarrow \text{Mod}^p(\alpha)(M')\}$$

c) **Using**

$$\mathcal{M}' = \{M' \in \text{Mod}^p(\Sigma', E') : (\exists M \in \mathcal{M}) \text{ și un morfism universal } j : M \longrightarrow \text{Mod}^p(\alpha)(M')\}$$

Dacă o prezentare importă mai multe submodule se utilizează conjuncția condițiilor 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 prezentări și săgețile sunt incluziuni de prezentări. Dacă un modul este importat pe două cai 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 folosește pentru același scop, modularizarea, așa numita metodă a programării parametrizate.

Așa numitele argumente, formale respective actuale, din programarea clasică sunt înlocuite de așa numiții parametri, formali respectivi actuali. Parametri formali sunt prin definiție teorii, iar parametri actuali sunt prin definiție morfisme de prezentări.

Vom ilustra la nivel teoretic tehnica parametrizării programelor.

Fie $M[T1, T2, T3]$ un modul parametrizat formal de trei teorii $T1$, $T2$ și $T3$. Numărul trei este utilizat pentru acest exemplu deoarece în general orice număr finit este permis. Nu uitați că practic, în programe, teoriile sunt reprezentate de prezentări ale lor.

Fie $T1 + T2 + T3$ suma directă în categoria incluziunilor de teorii. Notăm cu $i1 : T1 \hookrightarrow T1 + T2 + T3$, $i2 : T2 \hookrightarrow T1 + T2 + T3$, $i3 : T3 \hookrightarrow T1 + T2 + T3$ și $i : T1 + T2 + T3 \hookrightarrow M[T1, T2, T3]$ incluziunile care apar.

Fie $M[m1, m2, m3]$ modulul obținut prin înlocuirea parametrilor formali cu parametri actuali $m1$, $m2$ și $m3$. Menționăm că $m1 : T1 \rightarrow M1$, $m2 : T2 \rightarrow M2$ și $m3 : T3 \rightarrow M3$ sunt morfisme de prezentări. Vom explica în continuare cine este $M[m1, m2, m3]$.

$$\begin{array}{ccc}
 M[T1, T2, T3] & \xrightarrow{\quad} & M[m1, m2, m3] \\
 \uparrow i & & \uparrow \\
 T1 + T2 + T3 & \xrightarrow{m1+m2+m3} & M1 + M2 + M3 \\
 \uparrow i_k & & \uparrow j_k \\
 T_k & \xrightarrow{m_k} & M_k
 \end{array}$$

Fie $M1 + M2 + M3$ suma directă în categoria incluziunilor. Notăm cu $j1 : M1 \hookrightarrow M1 + M2 + M3$, $j2 : M2 \hookrightarrow M1 + M2 + M3$ și $j3 : M3 \hookrightarrow M1 + M2 + M3$ incluziunile care sunt morfisme structurale ale sumei directe.

Fie $m1 + m2 + m3 : T1 + T2 + T3 \rightarrow M1 + M2 + M3$ unicul morfism de teorii cu proprietate

$$ik; (m1 + m2 + m3) = mk; jk$$

pentru orice $k \in \{1, 2, 3\}$.

La ultimul pas al construcției utilizând morfismul $m1 + m2 + m3 : T1 + T2 + T3 \rightarrow M1 + M2 + M3$ se face o sumă fibrată care conservă incluziunea $i : T1 + T2 + T3 \hookrightarrow M[T1, T2, T3]$. Se obține astfel incluziunea $M1 + M2 + M3 \hookrightarrow M[m1, m2, m3]$. Această ultimă incluziune împreună cu incluziunile $jk : Mk \hookrightarrow M1 + M2 + M3$ dau structura programului $M[m1, m2, m3]$.

[1]

[1, comentariu]

[1, ???, 4]

[???

Bibliography

- [1] text
- [??] text43
- [2] F. Baader, T. Nipkow. Term Rewriting and All That. Cambridge University Press, 1998
- [3] 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
- [4] V. E. Căzănescu, G. Roşu. Weak invlusion systems. *Mathematical Structure in Computer Science*, 7(2):195-206, 1997
- [5] R. Diaconescu. *Institution-independent Model Theory* Birkhäuser 2008
- [6] 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
- [7] R. Diaconescu, J. Goguen, P. Stefaneas. Logical support for modularisation. In Gerard Huet şi Gordon Plotkin, editori, *Logical Environments*, pag 83-130. Cambridge, 1993.
- [8] H. Ehrig, B. Mahr. Fundamentals of Algebraic Specification 1: Equations and Initial Semantics. EATCS Monographs on Theoretical Computer Science, vol 6. Springer-Verlag, 1985
- [9] H. Ehrig, B. Mahr. Fundamentals of Algebraic Specification 2: Module Specifications and Constraints. EATCS Monographs on Theoretical Computer Science, vol 21. Springer-Verlag, 1990
- [10] J. Goguen, Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95-146, 1992
- [11] J. Goguen, J. Thatcher, E. Wagner, J. Wright. Initial algebra semantics and continuous algebras, *Journal of the ACM*, 24:68-95, 1977
- [12] 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
- [13] 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
- [14] 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
- [15] F. W. Lawvere. Functorial semantics of algebraic theories. *Proceedings, National Academy of Sciences*, 50:869-873, 1993. Summary of Ph.D. Thesis, Columbia University

- [16] S. MacLane. *Categories for the working mathematician*. Graduate Texts in Mathematics 5, 2nd edition, Springer-Verlag, 1998.
- [17] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer Verlag, 2002
- [18] Terese, *Term Rewriting Systems*, Cambridge University Press, 2003