SEMINAR DE PROGRAMARE LOGICA CU NOTATIA COMPUNERII DE FUNCTII IN SENSUL SAGETILOR

Claudia Mureşan

Notatie: Pentru orice functii $A \xrightarrow{f} B \xrightarrow{g} C$, se va nota compunerea in sensul sagetilor: $f; g = g \circ f : A \to C$. Aceasta notatie se extinde la cazul functiilor multisortate in modul evident.

Sa consideram urmatoarea problema: determinati modelul initial al specificatiei urmatoare, si demonstrati ca este model initial:

- signatura (Σ):
  - multimea de sorturi este formata din doua sorturi, notate Nat si $Bool$;
  - simbolurile de operatii sunt:
    - $0\ :\to\ Nat$;
    - $true,\ false\ :\to Bool$;
    - $s,\ f\ :\ Nat\ \to\ Nat$;
    - $<\ :\ Nat\ Nat\ \to\ Bool$;
- multimea Γ a $\Sigma-$ecuatiilor este formata din urmatoarele $\Sigma-$ecuatii:
  - $(\forall X : Nat)\, 0 < s(X) \doteq_{Bool} true$;
  - $(\forall X : Nat)\, s(X) < 0 \doteq_{Bool} false$;
  - $(\forall X, Y : Nat)\, s(X) < s(Y) \doteq_{Bool} X < Y$;
  - $(\forall X : Nat)\, X < X \doteq_{Bool} false$;
  - $f(0) \doteq_{Nat} s(0)$;
  - $(\forall X : Nat)\, f(X) \doteq_{Nat} s(s(X))$ if $0 < X$.

Am adoptat notatia infixata pentru $<$. Notam aceasta specificatie, cum este uzual, Γ, la fel ca pe multimea de ecuatii din cadrul ei. Prin definitie, un model al specificatiei Γ este o $\Gamma-$algebra. Prin definitie, modelul initial al specificatiei Γ este $\Gamma-$algebra initiala, adica obiectul initial in categoria $\Gamma-$algebrelor, adica acea $\Gamma-$algebra $\mathcal{I}$ cu proprietatea ca, oricare ar fi o $\Gamma-$algebra $\mathcal{A}$, exista un unic $\Gamma-$morfism de la $\mathcal{I}$ la $\mathcal{A}$. Este corecta exprimarea "o $\Gamma-$algebra initiala", dar este corecta si exprimarea "$\Gamma-$algebra initiala", pentru ca $\Gamma-$algebra initiala este unica pana la un $\Gamma-$izomorfism.

Observand fragmentul din specificatia Γ de mai sus dat de specificatia lui Lawvere (0 si s), cu siguranta banuiti ca, in modelul initial al specificatiei Γ, multimea suport de sort Nat este $\mathbb{N}$, multimea numerelor naturale. Intr-adevar, vom demonstra ca acest fapt este adevarat. Observand acest lucru, imediat se observa si faptul ca ultima ecuatie din specificatia Γ, anume acea ecuatie conditionata, poate fi inlocuita cu ecuatia neconditionata echivalenta:

$$(\forall X : Nat)\, f(s(X)) \doteq_{Nat} s(s(s(X))).$$

Aceste ecuatii sunt echivalente in specificatia Γ, adica inlocuirea uneia cu cealalta in aceasta specificatie nu duce la schimbarea modelului initial al acestei specificatii; indiferent pe care dintre aceste doua ecuatii o alegem ca ultima ecuatie a specificatiei Γ, modelul initial al specificatiei ramane acelasi. Desigur, ar fi mai simplu sa folosim ecuatia neconditionata de mai sus in locul celei conditionate, dar am ales sa scriem acea ecuatie conditionata pentru a ilustra tratarea ecuatiilor conditionate in demonstratii de tipul celei de mai jos.

De asemenea, ecuatia $(\forall X : Nat)\, X < X \doteq_{Bool} false$ din specificatia Γ putea fi inlocuita cu ecuatia $0 < 0 \doteq_{Bool} false$, care trateaza cazul complementar cazurilor tratate in primele 3 ecuatii din definitia lui $<$, si care ar putea inlocui ecuatia $(\forall X : Nat)\, X < X \doteq_{Bool} false$ fara a schimba modelul initial al specificatiei Γ. Alegerea ecuatiei $(\forall X : Nat)\, X < X \doteq_{Bool} false$ in locul celei mai simple $0 < 0 \doteq_{Bool} false$ ne ajuta in demonstratia de mai jos, permitandu-ne sa tratam cazul 3 de la finalul demonstratiei in acea maniera directa si fara a face o demonstratie prin inductie, pe care ar fi fost necesar s-o scriem daca foloseam ecuatia $0 < 0 \doteq_{Bool} false$.

In CafeObj, specificatia Γ de mai sus poate fi implementata prin urmatorul modul:

```
mod! NATF{

[Nat]

op 0 : -> Nat
ops s f : Nat -> Nat
op _<_ : Nat Nat -> Bool

vars X Y : Nat

eq 0 < s(X) = true .
eq s(X) < 0 = false .
eq s(X) < s(Y) = X < Y .
eq X < X = false .

eq f(0) = s(0) .
ceq f(X) = s(s(X)) if 0 < X .
}
```

Desigur, ultima ecuatie din modulul de mai sus putea fi scrisa ca ecuatie neconditionata sub forma:

eq f(s(X)) = s(s(s(X))) .

si modelul initial al specificatiei descrise nu s-ar schimba, precum am observat si mai sus.

De asemenea, aici poate e mai usor de observat ca puteam scrie:

eq 0 < 0 = false .

in loc de:

eq X < X = false .

si obtineam acelasi model initial, dar alegerea pe care am facut-o intre aceste doua ecuatii poate ajuta la rapiditatea rescrierii.

Ca o paranteza, trebuie sa remarcam faptul ca, intr-o implementare a unei specificatii in CafeObj, cum este cea de mai sus, ecuatiile dau un sistem de rescriere, deci se aplica numai de la stanga la dreapta, pe cand, in orice model al specificatiei Γ definite anterior, ecuatiile se aplica, desigur, in oricare dintre sensuri. De asemenea, modulul NATF scris mai sus, ca orice modul pe care il scriem in CafeObj, importa automat (in modul protecting) modulul predefinit BOOL, care contine nu numai sortul Bool si valorile de adevar true si false, ci si operatiile booleene not, and, or, xor, implies etc., cu tot cu definitiile lor prin ecuatii, spre deosebire de specificatia Γ descrisa mai sus, care nu contine pe sortul Bool decat simbolurile de operatii zeroare true si false. Intr-adevar, pentru a putea trata ecuatia conditionata pe care o contine, modulul NATF apeleaza numai la sortul boolean, Bool, si la valorile de adevar, adica operatiile zeroare true si false, de sort rezultat Bool. Restul continutului modulului predefinit BOOL este, intr-o exprimare nu foarte precisa, distinct de sistemul de rescriere dat de specificatia NATF, adica restul continutului modulului BOOL nu intervine in aceasta rescriere.

Fie algebra $\{Nat, Bool\}-$sortata $\mathcal{N} = (\mathbb{N}, \{true, false\}, 0, true, false, s, f, <)$, unde:

- multimile suport ale lui $\mathcal{N}$ sunt: $N_{Nat} = \mathbb{N}$, $N_{Bool} = \{true, false\}$;
- 0 este primul numar natural;
- $<$ este relatia de ordine stricta uzuala pe $\mathbb{N}$, definita, ca orice relatie, sub forma unei operatii de sort rezultat sortul boolean: $< : \mathbb{N} \times \mathbb{N} \to \{true, false\}$, care ia valoarea $true$ exact pe perechile de numere naturale aflate in aceasta relatie;
- operatiile unare $s, f : \mathbb{N} \to \mathbb{N}$ sunt definite prin: pentru orice $n \in \mathbb{N}$:

  (i) $s(n) = n + 1$ (s este operatia succesor pe $\mathbb{N}$),

  (ii) $f(n) = \begin{cases} 1, & \text{daca } n = 0, \\ n + 2, & \text{daca } n > 0. \end{cases}$

Vom demonstra ca $\mathcal{N}$ este model initial pentru specificatia Γ de mai sus, adica este $\Gamma-$algebra initiala, adica este obiect initial in categoria $\Gamma-$algebrelor, adica specificatia Γ este adecvata pentru $\mathcal{N}$, sau, mai complet scris, specificatia Γ este adecvata pentru a descrie pe $\mathcal{N}$ ca model initial.

Am notat operatiile din algebra $\mathcal{N}$ la fel ca simbolurile de operatie din specificatia Γ, considerand ca nu exista pericol de confuzie, si intrucat aceste notatii sunt uzuale pentru aceste operatii pe $\mathbb{N}$ si pe multimea valorilor de adevar $\{true, false\}$.

Evident, $\mathcal{N}$ satisface specificatia Γ, adica este o $\Gamma-$algebra. Intr-adevar, $\mathcal{N}$ are doua sorturi, Nat si $Bool$, si cate o operatie corespunzatoare fiecarui simbol de operatie din signatura Σ, iar faptul ca $\mathcal{N}$, cu aceste operatii, satisface ecuatiile din specificatia Γ este imediat.

Fie $\mathcal{A} = (A_{Nat}, A_{Bool}, A_0, A_{true}, A_{false}, A_s, A_f, A_<)$ un alt model pentru aceasta specificatie, adica o alta $\Gamma-$algebra.

Ramane de demonstrat ca exista un unic $\Gamma-$morfism $h : \mathcal{N} \to \mathcal{A}$. Un $\Gamma-$morfism h intre $\mathcal{N}$ si $\mathcal{A}$ este o functie $\{Nat, Bool\}-$sortata $h = (h_{Nat}, h_{Bool})$, cu $h_{Nat} : \mathbb{N} \to A_{Nat}$ si $h_{Bool} : \{true, false\} \to A_{Bool}$, care comuta cu operatiile acestor $\Gamma-$algebre. Ca o paranteza, definitia unui $\Gamma-$morfism nu depinde de multimea Γ de $\Sigma-$ecuatii; un $\Gamma-$morfism este un $\Sigma-$morfism intre doua $\Gamma-$algebre. Asadar, un $\Gamma-$morfism intre $\mathcal{N}$ si $\mathcal{A}$ nu este nimic altceva decat un $\Sigma-$morfism intre $\mathcal{N}$ si $\mathcal{A}$.

Si acum sa demonstram existenta si unicitatea $\Gamma-$morfismului $h : \mathcal{N} \to \mathcal{A}$.

Unicitatea:

Fie $h, g : \mathcal{N} \to \mathcal{A}$ doua $\Gamma-$morfisme. Din comutarea acestor morfisme cu operatiile zeroare rezulta: $h_{Nat}(0) = A_0 = g_{Nat}(0)$, $h_{Bool}(true) = A_{true} = g_{Bool}(true)$ si $h_{Bool}(false) = A_{false} = g_{Bool}(false)$. Pentru a arata egalitatea lui h cu g, ramane de demonstrat ca, pentru orice $n \in \mathbb{N}^*$, $h_{Nat}(n) = g_{Nat}(n)$.

Demonstram prin inductie matematica dupa n ca, pentru orice $n \in \mathbb{N}$, $h_{Nat}(n) = g_{Nat}(n)$.

Pasul de verificare: Conform celor de mai sus, $h_{Nat}(0) = g_{Nat}(0) = A_0$.

Pasul de inductie: Presupunem ca $h_{Nat}(n) = g_{Nat}(n)$ pentru un anumit n natural, arbitrar, fixat.

Folosind definitia operatiei s din $\mathcal{N}$, comutarea oricarui $\Sigma-$morfism cu operatiile corespunzatoare simbolului de operatie s (asigurata de definitia unui $\Sigma-$morfism) si ipoteza de inductie, rezulta: $h_{Nat}(n+1) = h_{Nat}(s(n)) = A_s(h_{Nat}(n)) = A_s(g_{Nat}(n)) = g_{Nat}(s(n)) = g_{Nat}(n+1)$.

Conform principiului inductiei matematice rezulta ca, pentru orice $n \in \mathbb{N}$, $h_{Nat}(n) = g_{Nat}(n)$, deci $h_{Nat} = g_{Nat}$.

Conform egalitatilor de mai sus ale valorilor celor doua morfisme pe operatiile zeroare de sort $Bool$, avem $h_{Bool} = g_{Bool}$.

Prin urmare, $h = g$ si deci unicitatea este demonstrata.

In continuare, pentru a nu incarca exprimarea, pentru orice simbol de operatie σ din signatura Σ si orice $\Sigma-$morfism j, vom spune: "j comuta cu σ" in loc de "j comuta cu operatiile corespunzatoare simbolului de operatie σ". De exemplu, vom inlocui exprimarea de acest gen de mai sus cu: "h comuta cu s".

Existenta:

Fie functia $\{Nat, Bool\}-$sortata $h = (h_{Nat}, h_{Bool}) : \mathcal{N} \to \mathcal{A}$, definita prin:

$$\begin{cases} h_{Bool}(true) = A_{true}; \\ h_{Bool}(false) = A_{false}; \\ h_{Nat}(0) = A_0; \\ (\forall n \in \mathbb{N}^*)\, h_{Nat}(n) = (\underbrace{A_s; \ldots; A_s}_{\text{de } n \text{ ori } A_s})(A_0). \end{cases}$$

Vom arata ca h este un $\Gamma-$morfism de la $\mathcal{N}$ la $\mathcal{A}$, adica un $\Sigma-$morfism de la $\mathcal{N}$ la $\mathcal{A}$, adica h comuta cu 0, $true$, $false$, s, $<$ si f.

De ce definim pe $h_{Nat}(n)$ astfel? Pentru ca trebuie sa avem comutarea lui h cu 0 si cu s, prin urmare trebuie sa avem: pentru orice n natural nenul, $h_{Nat}(n) = h_{Nat}((\underbrace{s; \ldots; s}_{\text{de } n \text{ ori } s})(0)) = (\underbrace{A_s; \ldots; A_s}_{\text{de } n \text{ ori } A_s})(h_{Nat}(0)) = (\underbrace{A_s; \ldots; A_s}_{\text{de } n \text{ ori } A_s})(A_0)$. Am aplicat aici faptul ca, in conformitate cu definitia operatiei s din $\mathcal{N}$, $n = 0 + \underbrace{1 + 1 + \ldots + 1}_{n \text{ de } 1} = (\underbrace{s; \ldots; s}_{\text{de } n \text{ ori } s})(0)$, apoi am aplicat de n ori comutarea unui $\Sigma-$morfism h cu s.

Conform definitiei sale, h comuta cu 0, $true$ si $false$.

Demonstram ca h comuta cu s.

Aplicand definitia lui h de mai sus, obtinem ca, pentru orice $n \in \mathbb{N}$, $h_{Nat}(s(n)) = h_{Nat}(n+1) = (\underbrace{A_s; \ldots; A_s}_{\text{de } n+1 \text{ ori } A_s})(A_0) = A_s((\underbrace{A_s; \ldots; A_s}_{\text{de } n \text{ ori } A_s})(A_0)) = A_s(h_{Nat}(n))$. Asadar, $h_{Nat}; s = A_s; h_{Nat}$, deci h comuta cu s.

Demonstram ca h comuta cu $<$. Avem de aratat ca, pentru orice $n, k \in \mathbb{N}$, $h_{Bool}(n < k) = h_{Nat}(n)\ A_<\ h_{Nat}(k)$. Fie, asadar, $n, k \in \mathbb{N}$, arbitrare, fixate. Avem de analizat cazurile: $n < k$, $k < n$ si $n = k$.

Cazul 1*:* $n < k$. In acest caz, $k - n - 1 \in \mathbb{N} = N_{Nat}$, deci este definit $h_{Nat}(k - n - 1)$. Din definitia lui h si din faptul ca $\mathcal{A}$ verifica ecuatiile din specificatia Γ rezulta ca:

$$h_{Bool}(n < k) = h_{Bool}(true) = A_{true} =$$

$$A_0\ A_<\ A_s(h_{Nat}(k - n - 1)) =$$

$$A_0\ A_<\ A_s((\underbrace{A_s; \ldots; A_s}_{\text{de } k-n-1 \text{ ori } A_s})(A_0)) =$$

$$A_0\ A_<\ (\underbrace{A_s; \ldots; A_s}_{\text{de } k-n \text{ ori } A_s})(A_0) =$$

$$A_s(A_0)\ A_<\ A_s((\underbrace{A_s; \ldots; A_s}_{\text{de } k-n \text{ ori } A_s})(A_0)) =$$

$$A_s(A_0)\ A_<\ (\underbrace{A_s;\ldots;A_s}_{\text{de } k-n+1 \text{ ori } A_s})(A_0) =$$

$$A_s(A_s(A_0))\ A_<\ A_s((\underbrace{A_s;\ldots;A_s}_{\text{de } k-n+1 \text{ ori } A_s})(A_0)) =$$

$$(A_s;A_s)(A_0)\ A_<\ (\underbrace{A_s;\ldots;A_s}_{\text{de } k-n+2 \text{ ori } A_s})(A_0) = \ldots =$$

$$(\underbrace{A_s;\ldots;A_s}_{\text{de } n \text{ ori } A_s})(A_0)\ A_<\ (\underbrace{A_s;\ldots;A_s}_{\text{de } k \text{ ori } A_s})(A_0) = h_{Nat}(n)\ A_<\ h_{Nat}(k).$$

In calculul de mai sus, am aplicat unor termeni din $\mathcal{A}$: o data ecuatia $(\forall X : Nat)\, 0 < s(X) \doteq_{Bool} true$, si de n ori ecuatia $(\forall X, Y : Nat)\, s(X) < s(Y) \doteq_{Bool} X < Y$, ambele citite de la dreapta la stanga. (A nu se trage concluzii eronate asupra sensului in care se aplica ecuatiile in CafeObj!)

Cazul 2: $k < n$. In acest caz, $n - k - 1 \in \mathbb{N} = N_{Nat}$, deci este definit $h_{Nat}(n - k - 1)$. Dupa modelul cazului 1, dar folosind aici ecuatia $(\forall X : Nat)\, s(X) < 0 \doteq_{Bool} false$ in locul ecuatiei $(\forall X : Nat)\, 0 < s(X) \doteq_{Bool} true$ aplicate in cazul 1, si aplicand ecuatia $(\forall X, Y : Nat)\, s(X) < s(Y) \doteq_{Bool} X < Y$ de k ori de aceasta data, calculam:

$$h_{Bool}(n < k) = h_{Bool}(false) = A_{false} =$$

$$A_s(h_{Nat}(n - k - 1))\ A_<\ A_0 =$$

$$A_s((\underbrace{A_s;\ldots;A_s}_{\text{de } n-k-1 \text{ ori } A_s})(A_0))\ A_<\ A_0 =$$

$$(\underbrace{A_s;\ldots;A_s}_{\text{de } n-k \text{ ori } A_s})(A_0)\ A_<\ A_0 =$$

$$A_s((\underbrace{A_s;\ldots;A_s}_{\text{de } n-k \text{ ori } A_s})(A_0))\ A_<\ A_s(A_0) =$$

$$(\underbrace{A_s;\ldots;A_s}_{\text{de } n-k+1 \text{ ori } A_s})(A_0)\ A_<\ A_s(A_0) =$$

$$A_s((\underbrace{A_s;\ldots;A_s}_{\text{de } n-k+1 \text{ ori } A_s})(A_0))\ A_<\ A_s(A_s(A_0)) =$$

$$(\underbrace{A_s;\ldots;A_s}_{\text{de } n-k+2 \text{ ori } A_s})(A_0)\ A_<\ (A_s;A_s)(A_0) = \ldots =$$

$$(\underbrace{A_s;\ldots;A_s}_{\text{de } n \text{ ori } A_s})(A_0)\ A_<\ (\underbrace{A_s;\ldots;A_s}_{\text{de } k \text{ ori } A_s})(A_0) = h_{Nat}(n)\ A_<\ h_{Nat}(k).$$

Cazul 3*:* $n = k$. Atunci: $h_{Bool}(n < k) = h_{Bool}(n < n) = h_{Bool}(false) = A_{false} = h_{Nat}(n)\ A_<\ h_{Nat}(n) = h_{Nat}(n)\ A_<\ h_{Nat}(k)$. Am folosit faptul ca $\mathcal{A}$ verifica ecuatia $(\forall X : Nat)\, X < X \doteq_{Bool} false$.

Asadar, h comuta si cu $<$.

Ramane de demonstrat ca h comuta cu f. Adica avem de demonstrat ca $h_{Nat}; f = A_f; h_{Nat}$, i. e., pentru orice $n \in \mathbb{N}$, $h_{Nat}(f(n)) = A_f(h_{Nat}(n))$.

$h_{Nat}(f(0)) = h_{Nat}(s(0)) = A_s(h_{Nat}(0)) = A_s(A_0) = A_f(A_0) = A_f(h_{Nat}(0))$, conform definitiei lui f in 0, comutarii lui h cu s, demonstrate anterior, definitiei lui h in 0 si faptului ca $\mathcal{A}$ verifica ecuatia $f(0) \doteq_{Nat} s(0)$.

Fie $n \in \mathbb{N}^*$. Vom demonstra ca $h_{Nat}(f(n)) = A_f(h_{Nat}(n))$, facand apel la ecuatia conditionata $(\forall X : Nat)\, f(X) \doteq_{Nat} s(s(X))$ if $0 < X$ din specificatia Γ.

Sa ne amintim din curs faptul ca $\mathcal{A}$ satisface ecuatia $(\forall X : Nat)\, f(X) \doteq_{Nat} s(s(X))$ if $0 < X$ daca si numai daca, pentru orice $x \in A_{Nat}$, daca $A_0\ A_<\ x = A_{true}$, atunci $A_f(x) = A_s(A_s(x))$ (observati ca acest fapt este echivalent cu definitia cu $\Sigma-$morfismul de la $\Sigma-$algebra libera generata de $\{X\}$ la $\mathcal{A}$, pentru ca un astfel de morfism duce pe 0 din $\Sigma-$algebra libera generata de $\{X\}$ in A_0 si comuta cu $<$; mai precis, conform definitiei, $\mathcal{A}$ satisface ecuatia $(\forall X : Nat)\, f(X) \doteq_{Nat} s(s(X))$ if $0 < X$ daca si numai daca, oricare ar fi $\Sigma-$morfismul $\alpha : T_\Sigma(\{X\}) \to \mathcal{A}$, daca $\alpha_{Bool}(0 < X) = A_{true}$, atunci $\alpha_{Nat}(f(X)) = \alpha_{Nat}(s(s(X)))$, adica, oricare ar fi $\Sigma-$morfismul $\alpha : T_\Sigma(\{X\}) \to \mathcal{A}$, daca $A_0\ A_<\ \alpha_{Nat}(X) = A_{true}$, atunci $A_f(\alpha_{Nat}(X)) = A_s(A_s(\alpha_{Nat}(X)))$, si acum notam $x = \alpha_{Nat}(X) \in A_{Nat}$, iar acest x poate lua orice valoare din A_{Nat}).

Avem ca $0 < n$, prin urmare, datorita definitiei lui h in 0 si $true$ si comutarii lui h cu $<$, care a fost deja demonstrata, au loc egalitatile: $A_0\ A_<\ h_{Nat}(n) = h_{Nat}(0)\ A_<\ h_{Nat}(n) = h_{Bool}(0 < n) = h_{Bool}(true) = A_{true}$, deci $h_{Nat}(n)$ satisface conditia: $A_0\ A_<\ h_{Nat}(n)$. Acum aplicam faptul ca $\mathcal{A}$ satisface ecuatia conditionata $(\forall X : Nat)\, f(X) \doteq_{Nat} s(s(X))$ if $0 < X$, si, inlocuind in aceasta ecuatie pe X cu $h_{Nat}(n)$ si aplicand succesiv de doua ori comutarea anterior demonstrata a lui h cu s, obtinem: $A_f(h_{Nat}(n)) = A_s(A_s(h_{Nat}(n)))$, asadar: $h_{Nat}(f(n)) = h_{Nat}(s(s(n))) = A_s(A_s(h_{Nat}(n))) = A_f(h_{Nat}(n))$.

Deci $h_{Nat}(f(n)) = A_f(h_{Nat}(n))$ pentru orice $n \in \mathbb{N}$, asadar $h_{Nat}; f = A_f; h_{Nat}$, adica h comuta si cu f.

Rezulta ca h este $\Sigma-$morfism intre $\Gamma-$algebrele $\mathcal{N}$ si $\mathcal{A}$, adica h este $\Gamma-$morfism intre $\mathcal{N}$ si $\mathcal{A}$.

Am demonstrat ca exista un unic $\Gamma-$morfism $h : \mathcal{N} \to \mathcal{A}$, prin urmare $\mathcal{N}$ este $\Gamma-$algebra initiala, adica obiect initial in categoria $\Gamma-$algebrelor, adica model initial pentru specificatia Γ.