Semàntica externa: una variant interessant de la semàntica de comportament

Vicent-Ramon Palasí Lallana

Report LSI–96–33–R

Semàntica externa: una variant interessant de la semàntica de comportament.

Vicent-Ramon Palasí Lallana
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
e-mail: vicent@goliat.upc.es

Abstract

Partint de la semàntica de comportament, es proposa una nova semàntica, anomenada externa. L'objectiu que es pretèn assolir amb aquesta semàntica és resoldre el problema de la correcció del software respecte d'una especificació algebraica.

1 Introducció

D'entre totes les teories que pretenen donar semàntica a les especificacions algebraiques, la semàntica inicial ([GTW75], [GTW78], [TWW79a]) és una de les més senzilles i intuïtives i també una de les més esteses. Tot i això, ja des del bell començament, es va palesar que aquesta semàntica era massa restrictiva per a modelar certs tipus abstractes de dades com poden ser piles, llistes, conjunts, etc.

Com a conseqüència, va aparèixer la semàntica de comportament ([SaW83], [SaT85], [Rei81], [Rei84] però aquí partirem de l'enfoc de [Niv87]), que intenta capturar el concepte intuïtiu d'"abstracció respecte a la implementació". Aquesta semàntica es basa en dividir les propietats que compleix una especificació algebraica en observables i no observables. Dues especificacions es consideren equivalents si les diferències entre elles són no observables.

Tot i que hi han hagut diverses formulacions de la "semàntica de comportament" (per un survey veure [BBK94] o [Kna93]), cap d'elles és del tot adequada per tractar el problema de la demostració de correctesa d'algorismes respecte una especificació algebraïca. Ho veurem amb un exemple.

Suposem que tenim un llenguatge de programació imperatiu en el qual l'operació de multiplicació no és predefinida. Suposem també que programem una funció per computar-la. El resultat podria ser de la forma següent:

```
function *(a,b:nat) ret c: nat
    c:=0;
    while b > 0 do
      c:=c+a;
      b:=b-1
    endwhile
endfunction
```

Si ara volem tractar el problema de la correctesa de l'algorisme, haurem de fer una especificació d'aquest. Com que ens trobem al camp de l'especificació algebraica, un resultat possible seria (utilitzem semàntica inicial):

```
spec MULT1 is
    sorts
      nat
    signature
      *: nat nat ⟶ nat
    equations
      ∀ a,b:nat
      *(zero,b)=zero
      *(suc(a),b)=+(*(a,b),b)
endspec
```

Se suposa que les operacions suc, zero *i* + *(amb notació prefixa) ja han estat definides. Les seves signatures i equacions s'ometen per raons de brevetat.*

Considerarem que la funció de nom "*" és correcta si és equivalent a l'especificació MULT1 segons alguna noció raonable d'equivalència.

Per trobar aquesta noció d'equivalència, l'opció més lògica és convertir el programa en una especificació algebraica (d'aquesta manera hem de comparar dues especificacions que sempre és més fàcil que comparar un programa i una especificació). Per fer això, podem aplicar la semàntica algebraica sobre la funció "*". El resultat serà de l'estil

```
spec MULT2 is
    sorts
      nat
    signature
      *: nat nat ⟶ nat
    equations
      ∀ a,b:nat
      *(a,b)=eval_function("function *(a,b:nat) ret c: nat ... endfunction",a,b)
endspec
```

L'operació eval_function *és part de la semàntica algebraica del llenguatge. Se suposa que totes les operacions de la semàntica algebraica ja han estat definides (amb les seves signatures i equacions) però s'ometen per raons de brevetat. Per un exemple exhaustiu de definició de semàntica algebraica d'un llenguatge veure* [GoP81] *o* [Wan80].

Ara podrem dir que la funció "*" és correcta respecte l'especificació MULT1 si les especificacions MULT1 i MULT2 (que és la semàntica algebraica de la funció "*") són "equivalents". És fàcil veure algunes propietats que necessàriament ha de complir aquesta "equivalència".

1. Dues especificacions poden ser equivalents encara que tinguin signatures diferents. (Així, a l'exemple, MULT2 té com a mínim una operació (eval_function) que MULT1 no té).

2. En tot cas, podem dividir les operacions d'una especificació en dues classes: aquelles que ens interessa especificar (com "*" a l'exemple) i aquelles que només incloem perquè són necessaries per definir les primeres (com "eval_function"). Les anomenarem, respectivament, observables i ocultes.

3. Per tant, dues especificacions seran equivalents si les operacions observables es "comporten" de la mateixa manera (no ens importa quines operacions ocultes hi hagi ni quin sigui el seu comportament). Notem que, per això, cal que les signatures de les dues especificacions, tot i poder ser diferents, tinguin les mateixes operacions observables.

4. Finalment, és ben sabut que un terme representa intuïtivament una "computació" del sistema de software especificat. Els termes que tinguin alguna operació oculta representen estats de computació interns que no són visibles en el comportament extern del sistema. Per tant, només els termes que tenen totes les operacions observables haurien de ser considerats per definir l'equivalència.

Val a dir que la necessitat d'operacions ocultes no es deu únicament als arguments exposats fins ara. Si volem especificar tots els tipus de dades que ens interessen (els tipus de dades semicomputables), calen operacions ocultes. En canvi, la necessitat o no de gèneres ocults és un important problema obert (per aquests temes, veure [BeT87]).

Malgrat totes les formulacions que hi han de la semàntica de comportament, no hi ha cap que compleixi totes les propietats esmentades abans (veure [BBK94]).

L'objectiu d'aquest article és presentar una variant de la semàntica de comportament que compleixi totes aquestes propietats i que sigui, per tant, adequada per a la demostració de programes. (De fet, aquest article forma part de la meva futura tesi, que tracta a la verificació automàtica d'algorismes).

A aquesta semàntica l'anomenarem "semàntica externa" per distingir-la de les semàntiques de comportament convencionals. Aquestes darreres es poden definir de dues maneres: a partir del concepte d'equivalència de comportament o bé mitjançant la relaxació de la relació de satisfacció. En canvi, la semàntica externa només es pot definir de la primera manera (la raó d'això es veurà a l'apartat 6).

L'estructura d'aquest article és la següent. Primer, donem un conjunt de conceptes bàsics d'especificació algebraïca per fixar la notació (apartat 2). Després definim la semàntica externa i finalment esbossem algunes conclusions i línies de treball futur.

2 Conceptes bàsics

Comencem per recordar alguns conceptes bàsics de la teoria de l'especificació algebraica.

Definició 1. Un S-conjunt C és una família de conjunts indexada per S, $C = \{C_s\}_{s \in S}$.

Definició 2. Una signatura simple Σ és una tupla $\Sigma = (S, F)$ on S és un conjunt els elements del qual s'anomenen gèneres i F és un $S^* \times S$-conjunt $F = \{F_{w,s}\}_{(w,s) \in S^* \times S}$.

Si $\sigma \in F_{w,s}$, on $w = w1 \times ... \times wn$ amb $w1, ..., wn, s \in S$, diem que σ és un símbol de funció de dominis w1,...,wn i gènere s. Ho notem així : $\sigma \in F_{w1..wn,s}$ o també $\sigma : w1 \times ... \times wn \longrightarrow s$.

Notem $sorts(\Sigma)$ a S i $opns(\Sigma)$ a F. A les variables del gènere s, les notem vars(s).

Definició 3. Siguin dues signatures simples $\Sigma_1 = (S_1, F_1)$ i $\Sigma_2 = (S_2, F_2)$. Diem que $\Sigma_1 \subseteq \Sigma_2$ si $S_1 \subseteq S_2$ i $F_1 \subseteq F_2$.

Definició 4. Sigui Σ=(S,F) una signatura simple i X un conjunt de variables. Els conjunts $T_{\Sigma_s}(X)$ es defineixen de la següent manera:

- Si $x \in X$ i $x \in vars(s)$, llavors $x \in T_{\Sigma_s}(X)$.
- Si $\sigma \in F_{\lambda,s}$, llavors $\sigma \in T_{\Sigma_s}(X)$.
- Si $\sigma \in F_{w_1...w_n,s}$, $t_1 \in T_{\Sigma_{w_1}}, ..., t_n \in T_{\Sigma_{w_n}}$, llavors $\sigma(t_1, ..., t_n) \in T_{\Sigma_s}$.

Als elements de $T_{\Sigma_s}(X)$ els anomenem termes de gènere s.

Definició 5. Sigui Σ una signatura simple i X un conjunt de variables. Definim $T_\Sigma(X) = \{T_{\Sigma_s}(X)\}_{s \in S}$ i $T_\Sigma = T_\Sigma(\varnothing)$. Als elements de $T_\Sigma(X)$ els anomenem *termes* i als de T_Σ *termes base*.

Definició 6. Sigui Σ=(S,F) una signatura simple. Una Σ-àlgebra A és un parell (A_S, A_F) on $A_S = \{A_s\}_{s \in S}$ i $A_F = \{\sigma_A\}_{\sigma \in F}$ de forma que:

- si $\sigma \in F_{\lambda,s}$, llavors $\sigma_A \in A_s$.
- si $\sigma \in F_{w_1..w_n,s}$, llavors $\sigma_A : A_{s_1}, ..., A_{s_n} \longrightarrow A_s$.

A σ_A se li diu interpretació del símbol de funció σ en A.

Definició 7. Sigui Σ una signatura simple. Sigui t un terme sobre una Σ-àlgebra A. Definim $\varepsilon_A(t)$ de la següent manera:

- Si $\sigma \in F_{\lambda,s}$ on $s \in S$, llavors $\varepsilon_A(\sigma) = \sigma_A$.
- Si $\sigma \in F_{w_1...w_n,s}$, $t_1 \in T_{\Sigma_{w_1}}, ..., t_n \in T_{\Sigma_{w_n}}$, llavors $\varepsilon_A(\sigma(t_1, ..., t_n)) = \sigma_A(\varepsilon_A(t_1), ..., \varepsilon_A(t_n))$.

A $\varepsilon_A(t)$ se li anomena avaluació de t en A.

Definició 8. Sigui Σ una signatura simple. Sigui X un conjunt de variables. Anomenem Σ-equació d'aritat n (o bé Σ-equació amb n condicions) a una (2*n+3)-tupla $(X, c_1, d_1, ..., c_n, d_n, t_1,$ on $t_1, t_2 \in T_{\Sigma_s}(X)$; $c_1, d_1 \in T_{\Sigma_{s_1}}(X)$; ...; $c_n, d_n \in T_{\Sigma^{\circ}_{s_n}}(X)$, amb $s, s_1, ..., s_n \in sorts(\Sigma)$.

Notarem $c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$ (o, de vegades, $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$) a l'equació $e = (X, c_1, d_1, ..., c_n, d_n, t_1, t_2)$. Al conjunt de variables d'una equació e, el notarem com a $vars(e)$.

Una Σ-equació d'aritat 0 s'anomena *equació simple o incondicional.*

Definició 9. Sigui A una Σ-àlgebra. Sigui X un conjunt de variables. Anomenem assignació de valors a una aplicació $v : X \longrightarrow A$.

Definició 10. Donada una assignació de valors v, i donat un terme $t \in T_\Sigma(X)$ definim $v^*(t)$ de la següent manera:

- Si $t \in X$, llavors $v^*(t) = v(t)$.
- Si t és de la forma $\sigma(t_1, ..., t_n)$, amb n $\geq$ 0, llavors $v^*(t) = \sigma(v^*(t_1), ..., v^*(t_n))$.

Definició 11. Direm que una Σ-àlgebra A satisfà una equació $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$ si $\forall v : vars(e) \longrightarrow A$ es compleix que: $\varepsilon_A(v^*(c_1)) = \varepsilon_A(v^*(d_1)) \wedge ... \wedge \varepsilon_A(v^*(c_n)) = \varepsilon_A(v^*(d_n))$ implica $\varepsilon_A(v^*(t_1)) = \varepsilon_A(v^*(t_2))$. Ho notarem com $A \models e$

Definició 12. Sigui E un conjunt d'equacions. Diem que $A \models E$ si $\forall e \in E$ es compleix que $A \models e$.

Definició 13. Anomenem especificació simple a la tupla $SPEC = (\Sigma, E)$ on Σ és una signatura simple i E un conjunt d'equacions. Anomenarem $eqns(SPEC)$ a E.

Definició 14. Sigui SPEC una especificació simple. Definim $Alg[SPEC] = \{A \mid A \models eqns(SPEC)\}$

3 Signatures esteses

Comencem definint el tipus de signatura escaient per a la teoria que es vol exposar. Com es va justificar a la introducció, la signatura que busquem ha de distingir entre gèneres i operacions que són observables i els que són ocults. Una forma de fer-ho és la següent:

Definició 15. Anomenarem "signatura estesa"[1] a una tupla $(\Sigma_{Obs}, \Sigma_{All})$ tal que $\Sigma_{Obs} = (S_{Obs}, F_{Obs})$, $\Sigma_{All} = (S, F)$ són signatures simples i, a més, $\Sigma_{Obs} \subseteq \Sigma_{All}$.

A Σ_{Obs} l'anomenarem signatura observable i, per tant, els elements de S_{Obs} i de F_{Obs} es diran gèneres observables i símbols de funció observables. A Σ_{All} li direm signatura ampliada. Als elements de $(S \backslash S_{Obs})$ i de $(F \backslash F_{Obs})$ els anomenarem respectivament gèneres ocults i símbols de funció ocults[2].

Definició 16 Sigui $\Sigma = (\Sigma_{Obs}, \Sigma_{All})$ una signatura estesa. Definim $sig\_obs(\Sigma) = \Sigma_{Obs}$

Ara definim el concepte de terme en una signatura estesa. Un terme de la signatura estesa no és més que un terme de la signatura ampliada. Diem que un terme és totalment observable si tots els símbols de funció que el componen són observables (és a dir, si el terme pertany a la signatura observable).

Definició 17. Sigui $\Sigma = (\Sigma_{Obs}, \Sigma_{All})$ una signatura estesa. Direm que A és una Σ-àlgebra si és una Σ_{All}-àlgebra.

Definim T_Σ, $T_\Sigma(X)$ com $T_{\Sigma_{All}}$, $T_{\Sigma_{All}}(X)$, respectivament. Anàlogament, per a tot gènere $s \in S$, definim $T_{\Sigma_s}(X)$, T_{Σ_s} com a $(T_{\Sigma_{All}})_s(X)$, $(T_{\Sigma_{All}})_s$, respectivament

Definició 18. Sigui $\Sigma = (\Sigma_{Obs}, \Sigma_{All})$ una signatura estesa i X un conjunt de variables. Definim $Tot_\Sigma(X) = T_{\Sigma_{Obs}}(X)$.

---

[1] Nota sobre terminologia: Encara que el nom de la semàntica és "externa", el tipus de signatura (i especificació) sobre el qual es basa es diu "estesa". Això és perquè "signatura externa" podria haver-se confòs amb "signatura observable", degut a les connotacions semàntiques d'aquests dos mots.

[2] Alguna vegada, utilitzarem "no observable" com a sinònim d'ocult

Òbviament, $Tot_\Sigma(X)$ és un subconjunt de $T_\Sigma(X)$.

Anàlogament definim $Tot_\Sigma = Tot_\Sigma(\emptyset)$ i als seus elements els anomenem termes base totalment observables.

Com déiem a la introducció, un terme totalment observable d'una especificació algebraica representa un resultat de computació visible externament.

Definició 19. Sigui A una Σ-àlgebra. Definim $A_{Tot} = \{\varepsilon_A(t) \mid t \in Tot_\Sigma\}$

Ara anem a introduïr la categoria adequada per parlar de semàntica externa. Per fer-ho, introduim primer els conceptes de morfisme extern i d'equivalència externa.

Definició 20. Siguin Σ_A i Σ_B dues signatures esteses tal que $sig\_obs(\Sigma_A) = sig\_obs(\Sigma_B)$. Sigui A una Σ_A-àlgebra i B una Σ_B-àlgebra. Un morfisme extern és una funció f entre A_{Tot} i B_{Tot} tal que per a tot terme base totalment observable t sobre A es compleix: $f(\varepsilon_A(t)) = \varepsilon_B(t)$.

Si aquesta aplicació és bijectiva, se li diu isomorfisme extern.

Lema 21. Siguin Σ_A i Σ_B dues signatures esteses tal que $sig\_obs(\Sigma_A) = sig\_obs(\Sigma_B)$. Sigui A una Σ_A-àlgebra i B una Σ_B-àlgebra. Només es pot donar un dels dos casos següents:

1. No hi ha morfisme extern entre A i B.
2. Hi ha un únic morfisme extern entre A i B.

Prova Es troba a l'annex.

Fixem-nos que, per definir un morfisme extern entre dues àlgebres, no cal que tinguin la mateixa signatura sinó que només han de tenir la mateixa signatura observable. Això és coherent amb la nostra filosofia segons la qual els gèneres i símbols de funció ocults només serveixen per definir el comportament dels observables però no són visibles "des de fora". El mateix passa amb la categoria que definim a continuació.

Definició 22. La categoria $Extern(\Sigma_{Obs})$ conté, com a objectes, totes les Σ-àlgebres tal que $sig\_obs(\Sigma) = \Sigma_{Obs}$ i, com a morfismes, els morfismes externs.

Definim ara el concepte d'equivalència externa entre dues àlgebres.

Definició 23. Siguin Σ_A i Σ_B dues signatures esteses tal que $sig\_obs(\Sigma_A) = sig\_obs(\Sigma_B)$. Sigui A una Σ_A-àlgebra i B una Σ_B-àlgebra. Diem que A i B són equivalents externament (i ho notarem $A \equiv_E B$) si existeix un isomorfisme extern entre A i B.

Com es veu en aquesta definició, dues àlgebres equivalents no cal que tinguin la mateixa signatura: tan sols han de tenir els mateixos gèneres i operacions observables, tal com volíem a la introducció. A la introducció també afirmàvem que l'equivalència que buscàvem, havia de tenir en compte només els termes totalment observables[3]*. Això es pot observar al següent lema.*

Lema 24. Siguin Σ_A i Σ_B dues signatures esteses tal que $sig\_obs(\Sigma_A) = sig\_obs(\Sigma_B)$. Sigui A una Σ_A-àlgebra i B una Σ_B-àlgebra. A i B són equivalents externament si i només si:

$\forall t_1, t_2 \in Tot_{\Sigma_A}$ es compleix que $A \models t_1 = t_2$ si i només si $B \models t_1 = t_2$

Prova. Es troba a l'annex.

4 Satisfacció externa d'una equació

Seguint la filosofia de la semàntica externa, una equació se satisfà si ho fan totes les seves conseqüències totalment observables (que són les úniques que ens importen, tal com ja vam veure a la introducció). Per formalitzar aquesta idea intuïtiva hem de fer primer les següents definicions:

Definició 25. Sigui Σ una signatura estesa. Sigui X un conjunt de variables. Anomenem Σ-equació a una Σ_{All}-equació.

Definició 26 Sigui Σ una signatura estesa . Un Σ-context extern sobre el gènere s és un terme $ec[z]$ pertanyent a $T_\Sigma(z)$ on z és una variable de gènere s.

Definició 27 Donat un terme $t \in T_\Sigma(X)$ i un Σ-context extern $ec[z]$, indicarem per $ec[t]$ al terme que s'obté substituïnt z per t dins $ec[z]$, és a dir, $ec[t] = ec[z \leftarrow t]$.

$ec[t]$ és qualsevol terme que té t com a subterme. Una conseqüència totalment observable d'un terme t es pot definir com un terme totalment observable que es pot derivar a partir de t (o dels contextos de t). Per tant, la satisfacció externa pot definir-se de la següent manera:

Definició 28 Sigui Σ una signatura estesa. Sigui $e : t_1 = t_2$ una Σ-equació *base* sense condicions de gènere s (és a dir, $t_1, t_2 \in T_{\Sigma_s}$). Direm que una Σ-àlgebra A satisfà externament e (i ho notem $A \models_E e$) si per a tot context extern $ec[z]$ de gènere s es compleix que:

$\forall l_1, l_2 \in Tot_\Sigma \quad (l_1 \equiv_A ec[t_1]) \wedge (l_2 \equiv_A ec[t_2])$ implica $l_1 \equiv_A l_2$

[3]Fixem-nos d'altra banda que $Tot_{\Sigma_A} = Tot_{\Sigma_B}$

Definició 29 Sigui Σ una signatura estesa. Sigui $e : c_1 = d_1 \,\&...\&\, c_n = d_n \Rightarrow t_1 = t_2$, una Σ-equació. Direm que una Σ-àlgebra A satisfà externament e (i ho notem $A \models_E e$) si

$\forall v : vars(e) \longrightarrow T_\Sigma$ es compleix que
$(A \models_E v^*(c_1) = v^*(d_1)) \wedge ... \wedge (A \models_E v^*(c_n) = v^*(d_n))$ implica $A \models_E v^*(t_1) = v^*(t_2)$

Anàlogament al que passa en semàntica inicial i de comportament podem definir el concepte de teoria a partir de la definició de la satisfacció d'una equació.

Definició 30 Sigui Σ una signatura estesa. Sigui M un conjunt de Σ-àlgebres. Anomenarem teoria externa de M (i notarem $ExTheo(M)$) al conjunt de totes les Σ-equacions que satisfan externament totes les àlgebres de M.

$ExTheo(M) = \{e \mid \forall A \in M \quad A \models_E e\}$

Si A és una Σ-àlgebra, anomenarem $ExTheo(A)$ a $ExTheo(\{A\})$.

5 Especificacions esteses

Donat el concepte de signatura estesa, el d'especificació estesa és fàcilment definible.

Definició 31. Una especificació estesa és una tupla $SPEC = (\Sigma, E)$, on $\Sigma = (\Sigma_{Obs}, \Sigma_{All})$ és una signatura estesa i E és un conjunt de Σ-equacions. Definim $sig(SPEC) = \Sigma$, $sig\_obs(SPEC) = \Sigma_{Obs}$ i $eqns(SPEC)$=E.

Definició 32. Sigui $SPEC$ una especificació estesa tal que $sig(SPEC) = \Sigma$. Anomenem $T_{\Sigma_{SPEC}}$ a T_Σ. Anàlogament, definim $T_{\Sigma_{SPEC}}(X)$ i $(T_{\Sigma_{SPEC}})_s$. Anomenem Tot_{SPEC} i $Tot_{SPEC}(X)$ a Tot_Σ i $Tot_\Sigma(X)$.

Definició 33 Sigui $SPEC = (\Sigma, E)$ una especificació estesa on $\Sigma = (\Sigma_{Obs}, \Sigma_{All})$. Definim els següents conceptes:

1. $\equiv_{SPEC}$ (*congrüència definida per l'especificació SPEC*)
2. $[t]_{\equiv_{SPEC}}$ (*classe d'equivalència a què pertany el terme t segons aquesta congrüència*)
3. T_{SPEC} (*àlgebra inicial de SPEC*)

com a, respectivament,

1. $\equiv_{SPEC'}$
2. $[t]_{\equiv_{SPEC'}}$
3. $T_{SPEC'}$

on SPEC' és l'especificació simple definida com $SPEC' = (\Sigma_{All}, E)$

6 Semàntica d'una especificació estesa

En aquesta secció definirem la semàntica externa d'una especificació. Per això, ens basarem en el principi que dues àlgebres que són equivalents externament haurien de ser considerades iguals en relació a la semàntica externa.

Definició 34 Sigui *SPEC* una especificació estesa. Definim la semàntica externa laxa de *SPEC* com:

$$Extern[SPEC] = \{A \mid \exists\, B \in Alg[SPEC] \text{ tal que } B \equiv_E A\}$$

És a dir, les àlgebres de Extern[SPEC] són aquelles que són externament equivalents a les que compleixen les equacions de SPEC.

En semàntica de comportament, hi ha la "bona" propietat que dues àlgebres són equivalents si comparteixen la mateixa teoria. Això permet definir la semàntica comportacional (laxa) d'una especificació algebraica com les àlgebres que satisfan les seves equacions en comportament.

En semàntica externa, en canvi, dues àlgebres equivalents poden no satisfer les mateixes equacions ja que poden tenir signatures diferents i, per tant, les equacions d'una poden no tenir sentit en la signatura de l'altra. Per tant, l'única forma de definir la semàntica externa (laxa) d'una especificació algebraica és com ho hem fet: a partir de la noció d'equivalència externa.

Definim també la semàntica externa inicial i la semàntica externa final a partir d'aquesta noció, com veurem tot seguit.

Definició 35 La semàntica externa inicial i final es defineixen respectivament com a:

$$Ext - I[SPEC] = \{A \mid A \equiv_E T_{SPEC}\}$$
$$Ext - F[SPEC] = \{A \mid A \equiv_E F[SPEC]\}$$

, on T_{SPEC} i $F[SPEC]$ són respectivament les àlgebres inicial i final de *SPEC*.

7 Conclusions

Hem vist quines propietats ha de tenir una semàntica que serveixi per tractar el problema de la correcció d'un programa. Hem definit un nou tipus de semàntica (anomenada "externa") que compleix totes aquestes propietats. S'han descrit els nous conceptes de morfisme extern, categoria externa, equivalència externa entre àlgebres, satisfacció externa d'una equació, semàntica externa laxa, inicial i final. També s'ha vist que, al contrari del que passava en semàntica de comportament, el concepte de satisfacció no serveix per

definir la semàntica externa d'una especificació algebraica.

Les línies de treball futur són dues. D'una banda, es pot estendre la semàntica externa perquè suporti especificacions parametritzades del tipus "PILA[T]",on T pot ser instanciada pel qualsevol tipus (per més detalls sobre aquest tipus d'especificacions veure [TWW79b]).

D'altra banda, aquest article és part de la meva futura tesi, que proposa un mètode per deduir automàticament la correcció o no d'un programa respecte una especificació algebraïca. La noció de correcció que emprarem es definirà a partir de la semàntica externa que s'ha descrit en aquest paper.

8 Referències

[BBK94] BERNOT,G. BIDOIT,M. KNAPIK,T. *Behavioural approaches to algebraic specifications.* Acta Informatica, 31 (1994), pp. 651-671.

[BeT87] BERGSTRA,J.A TUCKER,J.V *Algebraic Specifications of Computable and Semicomputable Data Types.* Theoretical Computer Science, 50 (1987), pp. 186-200.

[GoP81] GOGUEN,J.A PARSAYE-GHOMI,K. *Algebraic Denotational Semantics Using Parameterized Abstract Modules.* Formalization of Programming Concepts, LNCS 107 (1981), pp. 292-309.

[GTW75] GOGUEN,J.A THATCHER,J.W WAGNER,E.G WRIGHT,J.B *Abstract data types as initial algebras and correctness of data representations.* Proc. ACM Conference on Computer Graphics, Pattern and Data Structure, New York (1975), pp. 89-93.

[GTW78] GOGUEN,J.A THATCHER,J.W WAGNER,E.G *An initial algebra approach to the specification, correctness and implementations of abstract data types.*, in: R.T.Yeh, ed., Current Trends in Programming Methodology;IV Data Structuring (Prentice-Hall, Englewood Clifts, NJ, 1978) pp. 80-149.

[Kna93] KNAPIK,T. *Spécifications algébriques observationnelles modulaires: une semantique fondée sur une relation de satisfaction observationnelle.* Thèse de l'Université de Paris-Sud, Orsay 1993.

[Niv87] NIVELA,P. *Semántica de Comportamiento en Lenguajes de Especificación* PhD thesis, directed by Fernando Orejas Valdés, Barcelona, 1987. Universitat Politècnica de Catalunya. Facultat d'Informàtica.

[Rei81] REICHEL,H. *Behavioural equivalence -a unifying concept for initial and final spec-*

ification methods Proceedings third Hungarian Computer Science Conference. Budapest (1981) 27-39.

[Rei84] REICHEL,H. *Behavioral validity of equations in abstract data types* Contributions to General Algebra 3, Proceedings of the Vienna Conference, Verlag B. G.Teubner, Sttugart (1985) 301-324.

[SaT85] SANNELLA,D. TARLECKY, A. *On observational equivalence and algebraic specification* Journal of Computer and System Science, 34 (1987) 150-178.

[SaW83] SANNELLA,D WIRSING,M. *A kernel language for algebraic specification and implementation* Proceedings International Conference on Foundations of Computation Theory. Sweden. Springer LNCS 158 (1983) 413-427.

[TWW79a] THATCHER,J.W WAGNER,E.G WRIGHT,J.B *Specifications of abstract data types using condicional axioms* IBM Research Report RC 6214, Yorktown Heigths, NY, 1979.

[TWW79b] THATCHER,J.W WAGNER,E.G WRIGHT,J.B *Data type specifications: parametrization and the power of specifications techniqûes* IBM Research Report RC 7757, Yorktown Heigths, NY, 1979.

[Wan80] WAND, M. *First-Order Identities as a Defining Language.* Acta Informatica, 14 (1980) pp. 337-357.

A Annex: Demostració dels lemes 21 i 24

En aquest annex, s'hi troben les demostracions dels lemes 21 i 24 que s'han enunciat dintre de l'article.

A.1 Prova del lema 21

Sembla *a priori* que hi han tres casos possibles:

1. No hi ha morfisme extern entre A i B.
2. Hi ha un únic morfisme extern entre A i B.
3. Hi ha més d'un morfisme extern entre A i B.

Cal, doncs, demostrar que els dos primers casos es poden donar però que el tercer no.

1. Que el primer cas es pot donar podem veure-ho pel següent exemple:

   Suposem una signatura estesa $\Sigma = (\Sigma_{Obs}, \Sigma_{All})$ tal que:

   $\Sigma_{Obs} = \Sigma_{All}$
   $sorts(\Sigma_{All}) = \{sort1\}$
   $opns(\Sigma_{All}) = \{op1, op2\}$ on $op1, op2 :\longrightarrow sort1$

   És fàcil de veure que els termes base totalment observables en aquesta signatura són op_1 i op_2. Ara considerem les dues Σ-àlgebres A i B tal que:

$A_{sort1} = \{\bullet\}$	$B_{sort1} = \{*, \#\}$
$op1_A = \bullet$	$op1_B = *$
$op2_A = \bullet$	$op2_B = \#$

   És obvi que no pot definir-se cap morfisme extern entre A i B, ja que l'única correspondència entre A i B que compleix $f(\varepsilon_A(t)) = \varepsilon_B(t)$ (on t és qualsevol terme base totalment observable) no és una aplicació perquè "$\bullet$" té dues imatges possibles en B (que són "$*$" i "$\#$").

2. Vejam ara que el segon cas es pot donar. Ho veurem amb l'exemple en què $A = B$. En aquest cas, un morfisme extern seria una funció que compleixi $f(\varepsilon_A(t)) = \varepsilon_A(t)$, per a cada terme base totalment observable. És obvi que aquest morfisme extern existeix i que és la funció identitat.

3. Finalment, demostrem que no pot donar-se el tercer cas. Ho farem per reducció a l'absurd. Suposem que hi ha dos morfismes externs entre A i B, anomenats f_1 i f_2. Aplicant la definició, per a cada terme base totalment observable, s'ha de complir

$$f_1(\varepsilon_A(t)) = \varepsilon_B(t)$$
$$f_2(\varepsilon_A(t)) = \varepsilon_B(t)$$

Per propietats de la igualtat, això vol dir que, per a tot terme base totalment observable t, s'ha de complir

$$f_1(\varepsilon_A(t)) = f_2(\varepsilon_A(t))$$

És a dir, els dos morfismes són el mateix i, per tant, la unicitat queda provada. □

A.2 Prova del lema 24

Com que el lema 24, conté un "si i només si", caldra demostrar els dos sentits de la implicació:

- Demostrem primer que, si A i B són equivalents externament, llavors:

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \text{ es compleix que } A \models t_1 = t_2 \text{ si i només si } B \models t_1 = t_2$$

Suposem que A i B són equivalents externament. Això vol dir que existeix un isomorfisme extern f entre A i B. Com que f és una bijecció es compleix:

$$\forall\, a_1, a_2 \in A_{Tot} \quad a_1 = a_2 \text{ si i només si } f(a_1) = f(a_2)$$

Per la definició de A_{Tot}, $x \in A_{Tot}$ si i només si $\exists\, t \in Tot_{\Sigma_A}$ tal que $\varepsilon_A(t) = x$. Llavors l'afirmació anterior es pot transformar en la següent:

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \quad \varepsilon_A(t_1) = \varepsilon_A(t_2) \text{ si i només si } f(\varepsilon_A(t_1)) = f(\varepsilon_A(t_2))$$

Ara bé, com que f és un morfisme extern entre Å i B, per a tot $t \in Tot_{\Sigma_A}$, $f(\varepsilon_A(t)) = \varepsilon_B(t)$. Per tant, obtenim:

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \quad \varepsilon_A(t_1) = \varepsilon_A(t_2) \text{ si i només si } \varepsilon_B(t_1) = \varepsilon_B(t_2)$$

Ara bé, aplicant la definició de satisfacció d'una equació en un àlgebra donada i el fet que t_1 i t_2 són termes bases[4], això es transforma en l'afirmació següent:

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \text{ es compleix que } A \models t_1 = t_2 \text{ si i només si } B \models t_1 = t_2$$

, que era el que volíem demostrar.

[4] I, per tant, per a tota assignació de valors v, es compleix que $v^*(t_1) = t_1$ i $v^*(t_2) = t_2$

- Demostrem ara la implicació recíproca, és a dir, que aquesta afirmació que acabem d'escriure implica que A i B són equivalents externament. Aplicant la definició de satisfacció d'una equació en un àlgebra donada i el fet que t_1 i t_2 són termes bases, l'afirmació que acabem d'escriure resulta ser equivalent a la següent (que anomenarem "afirmació α"):

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \quad \varepsilon_A(t_1) = \varepsilon_A(t_2) \text{ si i només si } \varepsilon_B(t_1) = \varepsilon_B(t_2)$$

Dit d'una altra manera, hem de veure que si l'afirmació α es compleix, existeix un isomorfisme extern entre A i B.

Per fer-ho, suposem que l'afirmació α es compleix. Definim f com la correspondència que compleix $f(\varepsilon_A(t)) = \varepsilon_B(t)$. Si f és una bijecció, haurem demostrat el que volíem. Farem la demostració en tres parts:

- Vejam primer que f és una aplicació, és a dir,

$$\forall\, a_1, a_2 \in A_{Tot} \quad a_1 = a_2 \text{ implica } f(a_1) = f(a_2)$$

Per la definició de A_{Tot}, $x \in A_{Tot}$ si i només si $\exists\, t \in Tot_{\Sigma_A}$ tal que $\varepsilon_A(t) = x$. Llavors l'afirmació anterior es pot transformar en la següent:

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \quad \varepsilon_A(t_1) = \varepsilon_A(t_2) \text{ implica } f(\varepsilon_A(t_1)) = f(\varepsilon_A(t_2))$$

Però, com que hem definit f de forma que $f(\varepsilon_A(t)) = \varepsilon_B(t)$, això és equivalent a:

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \quad \varepsilon_A(t_1) = \varepsilon_A(t_2) \text{ implica } \varepsilon_B(t_1) = \varepsilon_B(t_2)$$

que es compleix, per l'afirmació α.

- Vejam ara que f és injectiva, és a dir,

$$\forall\, a_1, a_2 \in A_{Tot} \quad f(a_1) = f(a_2) \text{ implica } a_1 = a_2$$

Per la definició de A_{Tot}, $x \in A_{Tot}$ si i només si $\exists\, t \in Tot_{\Sigma_A}$ tal que $\varepsilon_A(t) = x$. Llavors l'afirmació anterior es pot transformar en la següent:

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \quad f(\varepsilon_A(t_1)) = f(\varepsilon_A(t_2)) \text{ implica } \varepsilon_A(t_1) = \varepsilon_A(t_2)$$

Però, com que hem definit f de forma que $f(\varepsilon_A(t)) = \varepsilon_B(t)$, això és equivalent a:

$$\forall\, t_1, t_2 \in Tot_{\Sigma_A} \quad \varepsilon_B(t_1) = \varepsilon_B(t_2) \text{ implica } \varepsilon_A(t_1) = \varepsilon_A(t_2)$$

que es compleix, per l'afirmació α.

– Vejam ara que f és exhaustiva, és a dir, que:

$$\forall b \in B_{Tot} \;\; \exists a \in A_{Tot} \text{ tal que } f(a) = b$$

Per la definició de B_{Tot}, $x \in B_{Tot}$ si i només si $\exists t \in Tot_{\Sigma_B}$ tal que $\varepsilon_B(t) = x$. Llavors l'afirmació anterior es pot transformar en la següent (recordem que $Tot_{\Sigma_A} = Tot_{\Sigma_B}$):

$$\forall t \in Tot_{\Sigma_A} \;\; \exists a \in A_{Tot} \text{ tal que } f(a) = \varepsilon_B(t)$$

Però, sabem, per definició de f, que $f(\varepsilon_A(t)) = \varepsilon_B(t)$. A més, per definició de A_{Tot}, $\varepsilon_A(t) \in A_{Tot}$. Conseqüentment, si agafem $\varepsilon_A(t)$ com a a, ja ho tenim demostrat. □

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Research Reports – 1996

LSI–96–1–R "(Pure) Logic out of Probability", Ton Sales.

LSI–96–2–R "Automatic Generation of Multiresolution Boundary Representations", C. Andújar, D. Ayala, P. Brunet, R. Joan-Arinyo, and J. Solé.

LSI–96–3–R "A Frame-Dependent Oracle for Linear Hierarchical Radiosity: A Step towards Frame-to-Frame Coherent Radiosity", Ignacio Martin, Dani Tost, and Xavier Pueyo.

LSI–96–4–R "Skip-Trees, an Alternative Data Structure to Skip-Lists in a Concurrent Approach", Xavier Messeguer.

LSI–96–5–R "Change of Belief in SKL Model Frames (Automatization Based on Analytic Tableaux)", Matías Alvarado and Gustavo Núñez.

LSI–96–6–R "Compressibility of Infinite Binary Sequences", José L. Balcázar, Ricard Gavaldà, and Montserrat Hermo.

LSI–96–7–R "A Proposal for Word Sense Disambiguation using Conceptual Distance", Eneko Agirre and German Rigau.

LSI–96–8–R "Word Sense Disambiguation Using Conceptual Density", Eneko Agirre and German Rigau.

LSI–96–9–R "Towards Learning a Constraint Grammar from Annotated Corpora Using Decision Trees", Lluis Màrquez and Horacio Rodríguez.

LSI–96–10–R "POS Tagging Using Relaxation Labelling", Lluís Padró..

LSI–96–11–R "Hybrid Techniques for Training HMM Part-of-Speech Taggers", Ted Briscoe, Greg Grefenstette, Lluís Padró, and Iskander Serail.

LSI–96–12–R "Using Bidirectional Chart Parsing for Corpus Analysis", A. Ageno and H. Rodríguez.

LSI–96–13–R "Limited Logical Belief Analysis", Antonio Moreno.

LSI–96–14–R "Logic as General Rationality: A Survey", Ton Sales.

LSI–96–15–R "A Syntactic Characterization of Bounded-Rank Decision Trees in Terms of Decision Lists", Nicola Galesi.

LSI–96–16–R "Algebraic Transformation of Unary Partial Algebras I: Double-Pushout Approach", P. Burmeister, F. Rosselló, J. Torrens, and G. Valiente.

LSI–96–17–R "Rewriting in Categories of Spans", Miquel Monserrat, Francesc Rosselló, Joan Torrens, and Gabriel Valiente.

LSI–96–18–R "Strong Law for the Depth of Circuits", Tatsuie Tsukiji and Fatos Xhafa.

LSI–96–19–R "Learning Causal Networks from Data", Ramon Sangüesa i Solé.

LSI–96–20–R "Boundary Generation from Voxel-based Volume Representations", R. Joan-Arinyo and J. Solé.

LSI–96–21–R "Exact Learning of Subclasses of CDNF Formulas with Membership Queries", Carlos Domingo.

LSI–96–22–R "Modeling the Thermal Behavior of Biosphere 2 in a Non-Controlled Environment Using Bond Graphs", Angela Nebot, François E. Cellier, and Francisco Mugica.

LSI–96–23–R "Obtaining Synchronization-Free Code with Maximum Parallelism", Ricard Gavaldá, Eduard Ayguadé, and Jordi Torres.

LSI–96–24–R "Memoisation of Categorial Proof Nets: Parallelism in Categorial Processing", Glyn Morrill.

LSI–96–25–R "Decision Trees Have Approximate Fingerprints", Víctor Lavín and Vijay Raghavan.

LSI–96–26–R "Visible Semantics: An Algebraic Semantics for Automatic Verification of Algorithms", Vicent-Ramon Palasí Lallana.

LSI–96–27–R "Massively Parallel and Distributed Dictionaries on AVL and Brother Trees", Joaquim Gabarró and Xavier Messeguer.

LSI–96–28–R "A Maple package for semidefinite programming", Fatos Xhafa and Gonzalo Navarro.

LSI–96–29–R "Bounding the expected length of longest common subsequences and forests", Ricardo A. Baeza-Yates, Ricard Gavaldà, and Gonzalo Navarro.

LSI–96–30–R "Parallel Computation: Models and Complexity Issues", Raymond Greenlaw and H. James Hoover.

LSI–96–31–R "ParaDict, a Data Parallel Library for Dictionaries (Extended Abstract)", Joaquim Gabarró and Jordi Petit i Silvestre.

LSI–96–32–R "Neural Networks as Pattern Recognition Systems", Lourdes Calderón.

LSI–96–33–R "Semàntica externa: una variant interessant de la semàntica de comportament", (written in Catalan) Vicent-Ramon Palasí Lallana.

LSI–96–34–R "Automatic verification of programs: algorithm ALICE", V.R. Palasí Lallana.

Hardcopies of reports can be ordered from:

Nuria Sánchez
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Pau Gargallo, 5
08028 Barcelona, Spain
secrelsi@lsi.upc.es

See also the Department WWW pages, http://www-lsi.upc.es/www/