Limbaje şi sisteme de tipuri
Semantică denotaţională

- Note de curs şi seminar -

Eneia Nicolae Todoran

2012

Cuprins

1 Spaţii metrice în semantică - preliminarii matematice **3**
1.1 Convenţii de notaţie 3
1.2 Spaţii metrice 4

2 Semantică de continuare pentru un limbaj secvenţial simplu $\mathcal{L}$ **10**
2.1 Sintaxa limbajului $\mathcal{L}$ 10
2.2 Domeniul pentru semantica denotaţională 11
2.3 Funcţia semantică denotaţională 12
2.4 Implementare Haskell a semanticii denotaţionale 15

3 Semantică directă pentru $\mathcal{L}$ **17**

Capitolul 1

Spaţii metrice în semantică - preliminarii matematice

În aceste note de curs şi seminar sunt prezentate modele matematice proiectate utilizând metoda semanticii denotaţionale introdusă de Dana Scott şi Christopher Stratchey [7, 8]. Semantica operaţională şi Modelele denotaţionale sunt proiectate utilizând spaţii metrice. Spaţiile metrice sunt structuri matematice ce pot fi întâlnite în mod frecvent în ştiinţa calcultoarelor. Utilizarea lor în semantică a fost iniţiată de Maurice Nivat [6] şi a fost elaborată de şcoala din Amsterdam condusă de Jaco de Bakker [3]. În cele ce urmează se vor construi modele denotaţionale într-o categorie de spaţii metrice complete şi funcţii contractive [1]. De obicei, în semantica metrică se operează cu spaţii metrice în care distanţa între puncte este ≤ 1. Metodologia semanticii metrice este prezentată pe larg în monografia [3]. O abordare alternativă bine cunoscută în semantica denotaţională este bazată pe utilizarea de mulţimi parţial ordonate complete împreună cu funcţii continue [5].

1.1 Convenţii de notaţie

Notaţia $(x, y, ... \in)X$ introduce o mulţime X cu elemente tipice $x, y, ...$. Dată fiind o mulţime X, notăm prin $\mathcal{P}(X)$ colecţia (mulţimea) tuturor submulţimilor lui X. $\mathcal{P}_\pi(X)$ denotă colecţia tuturor submulţimilor lui X care au proprietatea π. Notaţia $f : X \to Y$ exprimă faptul că f este o funcţie cu domeniu X şi codomeniu Y. Utilizăm notaţia $(\, f \mid x \mapsto y\,)$, unde $x \in X$ şi $y \in Y$, pentru o *variantă* a lui $f : X \to Y$ (o *perturbare* într-un singur punct a funcţiei f), mai precis pentru funcţia $(\, f \mid x \mapsto y\,) : X \to Y$ definită astfel:

$$(\, f \mid x \mapsto y\,)(x') = \begin{cases} y & \textsf{if} \quad x' = x \\ f(x') & \textsf{if} \quad x' \neq x \end{cases}$$

Dacă $f : X \to X$ şi $f(x) = x$, spunem că x este un *punct fix* al lui f. Când acest punct fix este unic pentru f (a se vedea teorema lui Banach) scriem $x = fix(f)$.

1.2 Spaţii metrice

În această secţiune sunt prezentate doar definiţiile şi proprietăţile cele mai necesare pentru prezenta lucrare. Pentru mai multe detalii referitoare la utilizarea spaţiilor metrice în ştiinţa calculatoarelor cititorul poate consulta monografia [3].

Definiţia 1.2.1 *Un* spaţiu metric *este o pereche* (X, d)*, unde* X *este o mulţime iar* d *este o funcţie* $d : X \times X \to [0, 1]$ *care satisface următoarele proprietăţi:*

(1) $\forall x, y \in X : \quad d(x, y) = 0 \Leftrightarrow x = y$

(2) $\forall x, y \in X : \quad d(x, y) = d(y, x)$

(3) $\forall x, y, z \in X : \quad d(x, y) \leq d(x, z) + d(z, y)$

Funcţia d *se numeşte* metrică *sau* distanţă*. În cazul în care* d *satisface (4),*

(4) $\forall x, y, z \in X : \quad d(x, y) \leq max(d(x, z), d(z, y))$

în locul lui (3), d *se numeşte* ultrametrică *iar* (X, d) *se numeşte* spaţiu ultrametric.

În semantica metrică [3] se operează de obicei cu spaţii în care distanţa între puncte este ≤ 1. Proprietatea (3) este aşa-numita *inegalitate a triunghiului.* În mod evident, (4) implică (3), deci orice spaţiu ultrametric este un spaţiu metric. Dăm doua exemple de spaţii ultrametrice utilizate frecvent în semantica metrică.

Exemple 1.2.2

(1) Fie $(x \in)X$ *o mulţime oarecare.* Metrica discretă *peste* X *se defineşte astfel:*

$$d_X(x, x') = \textsf{if } x = x' \textsf{ then } 0 \textsf{ else } 1$$

Lăsăm ca exerciţiu pentru cititorul verificarea faptului că d_X *este o ultrametrică, şi* (X, d_X) *este spaţiu ultrametric complet (vezi 1.2.3(3)).*

(2) Pentru orice mulţime $(a \in)A$*, notăm prin* $(u \in)A^\infty = A^* \cup A^\omega$ *mulţimea tuturor secveţelor finite şi infinite peste* A*. Aşa-numita* metrică (sau distanţă) Baire *(vezi [6, 3])* $d_B : A^\infty \times A^\infty \to [0, 1]$ *este definită astfel:*

$$d_B(u, u') = 2^{-sup\,\{\, n \,|\, u[n] = u'[n] \,\}}$$

unde $u[n], u'[n]$ *sunt prefixele de lungime* n *ale secvenţelor* u *şi* u'*. Prin convenţie* $2^{-\infty} = 0$*. Secvenţa vidă* ϵ *este prefix pentru orice altă secvenţă. Astfel* $u[0] = u'[0] = \epsilon$*, pentru orice* $u, u' \in A^\infty$*, şi* $d_B(u, u') \leq 1$*. Utilizând simbolul '$\cdot$' drept operator de concatenare peste secvenţe, avem* $d_B(a{\cdot}u, a{\cdot}u') = \frac{1}{2} \cdot d_B(u, u')$ *pentru orice* $a \in A, u, u' \in A^\infty$*. De asemenea* $d_B(\epsilon, u) = 1$ *pentru orice* $u \neq \epsilon$*,* $d_B(a, a') = 1$ *(când* $a \neq a'$*) şi* $d_B(a^n, a^\omega) = 2^{-n}$ *(*a^n *este secvenţa care conţine* n *repetiţii ale simbolului* a*, iar* a^ω *este secvenţa infinită). Distanţa Baire satisface*

$$d_B(u, u') \leq max(d_B(u, u''), d_B(u'', u'))$$

pentru orice $u, u', u'' \in A^\infty$. Lăsăm ca exerciţiu pentru cititor verificarea inegalităţii de mai sus. Urmează că (A^∞, d_B) este un spaţiu ultrametric.

Definiţia 1.2.3 *Fie (X, d) un spaţiu metric şi fie $(x_i)_i$ o secvenţă în X.*

(1) Spunem că $(x_i)_i$ este o secvenţă Cauchy *dacă:*

$$\forall \epsilon > 0 \, \exists k \in \mathbb{N} \, \forall m, n > k \, [d(x_m, x_n) < \epsilon]$$

(2) Fie $x \in X$. Spunem că $(x_i)_i$ converge *la x şi numim pe x* limita *lui $(x_i)_i$ dacă*

$$\forall \epsilon > 0 \, \exists k \in \mathbb{N} \, \forall n > k \, [d(x, x_n) < \epsilon]$$

Spunem că secvenţa $(x_i)_i$ este convergentă *şi scriem $x = lim_i x_i$.*

(3) Spaţiul metric (X, d) se zice complet*, dacă orice secvenţă Cauchy conţinută în X converge la un element din X.*

Definiţia 1.2.4 *Fie $(X_1, d_1), (X_2, d_2)$ spaţii metrice.*

(1) Spunem că spaţiile $(X1, d_1)$ şi (X_2, d_2) sunt izometrice *dacă există o bijecţie $f : X_1 \to X_2$ astfel încât:*

$$\forall x, y \in X_1 [d_2(f(x), f(y)) = d_1(x, y)]$$

Atunci când X_1 şi X_2 sunt izometrice scriem $X_1 \cong X_2$.

(2) Spunem că funcţia $f : X_1 \to X_2$ este o contracţie *dacă există un număr real k, $0 \leq k < 1$, astfel încât:*

$$\forall x, y \in X_1 [d_2(f(x), f(y)) \leq k \cdot d_1(x, y)]$$

Se notezăm prin $X_1 \xrightarrow{k} X_2$ mulţimea tuturor funcţiilor k-contractive de la X_1 la X_2.

(3) O funcţie $f : X_1 \to X_2$ se zice nonexpansivă *dacă*

$$\forall x, y \in X_1 [d_2(f(x), f(y)) \leq d_1(x, y)]$$

Se notezăm prin $X_1 \xrightarrow{1} X_2$ mulţimea tuturor funcţiilor nonexpansive de la X_1 la X_2.

Teorema 1.2.5 *(Banach) Fie (X, d) un spaţiu metric complet. Orice funcţie contractivă $f : X \to X$ are un punct fix unic.*

Definiţia 1.2.6 *Fie (X, d) un spaţiu metric. O submulţime C a lui X ($C \subseteq X$) se zice* compactă *dacă orice secvenţă infinită din C are o subsecvenţă convergentă cu limită în C.*

Definiţia 1.2.7 *Fie (X, d_X), (X_i, d_{X_i}) spaţii (ultra) metrice, pentru $i \in \mathbb{N}$. Peste mulţimile $(x \in)X$, $(f\in)X_1 \to X_2$ (mulţimea funcţiior cu domeniu X_1 şi codomeniu X_2), $((x_1, x_2)\in)X_1\times X_2$ (produsul cartezian), $(u, v\in)X_1+X_2$ (sumă sau reuniunea disjunctă), $(u, v \in)\Sigma_{i\in\mathbb{N}}X_i$ (sumă infinită, sau reuniunea disjunctă infinită) şi $(U, V\in)\mathcal{P}(X)$ (mulţimea submulţimilor lui X) se pot defini următoarele metrici:*

(1) $d_{\frac{1}{2}\cdot X} : X \times X \to [0, 1]$

$$d_{\frac{1}{2}\cdot X}(x, x') = \tfrac{1}{2} \cdot d_X(x, x')$$

(2) $d_{X_1\to X_2} : (X_1 \to X_2) \times (X_1 \to X_2) \to [0, 1]$

$$d_{X_1\to X_2}(f_1, f_2) = sup_{x\in X_1}\ d_{X_2}(f_1(x), f_2(x))$$

(3) $d_{X_1\times X_2} : (X_1 \times X_2) \times (X_1 \times X_2) \to [0, 1]$

$$d_{X_1\times X_2}((x_1, x_2), (x_1', x_2')) = max\{d_{X_1}(x_1, x_1'), d_{X_2}(x_2, x_2')\}$$[1]

(4) $d_{X_1+X_2} : (X_1 + X_2) \times (X_1 + X_2) \to [0, 1]$

$$d_{X_1+X_2}(u, v) = \\ \textsf{if } u, v\in X_1 \textsf{ then } d_{X_1}(u, v) \textsf{ else if } u, v\in X_2 \textsf{ then } d_{X_2}(u, v) \textsf{ else } 1$$

Spaţiul $X_1 + X_2$ poate fi definit astfel $X_1 + X_2 = (\{1\} \times X_1) \cup (\{2\} \times X_2)$ iar metrica $d_{X_1+X_2}$ este

$$d_{X_1+X_2}(u, v) \begin{cases} d_{X_1}(x_1, x_1') & \textsf{if} \quad u = (1, x_1), \quad v = (1, x_1') \\ d_{X_2}(x_2, x_2') & \textsf{if} \quad u = (2, x_2), \quad v = (2, x_2') \\ 1 & \textsf{if} \quad u = (1, x_1), \quad v = (2, x_2) \\ 1 & \textsf{if} \quad u = (2, x_2), \quad v = (1, x_1) \end{cases}$$

(5) *Conceptul poate fi generalizat la sume infinite. Se defineşte spaţiul $\Sigma_{i\in\mathbb{N}}X_i = \bigcup_{i\in\mathbb{N}}(\{i\} \times X_i)$, iar metrica peste $\Sigma_{i\in\mathbb{N}}X_i$ se defineşte astfel:*

$$d_{\Sigma_i X_i}(u, v) \begin{cases} d_{X_i}(x_i, x_i') & \textsf{if} \qquad u = (i, x_i), \quad v = (i, x_i') \\ 1 & \textsf{otherwise} \end{cases}$$

Conceptul de sumă infinită poate fi utilizat, în particular, pentru a defini o noţiune de listă finită. Notăm prin X^n produsul cartezian $X \times \cdots \times X$ în care spaţiul X apare de n ori. De exemplu $X^3 = X \times X \times X$. Atunci $\Sigma_i X^i$ este spaţiul listelor finite de elemente de tip X. Desigur, pentru fiecare spaţiu X^i metrica d_{X^i} este:

$$d_{X^i}((x_1, \cdots, x_i), (x_1', \cdots, x_i')) = max_{1\leq i\leq n}\{d_X(x_i, x_i')\}$$

[1] Definiţia produsului cartezian se generalizează cu uşurină la produse n-are (pentru $n \in \mathbb{N}$) de forma $X_1 \times \cdots \times X_n$. Distanţa între tuplele din $X_1 \times \cdots \times X_n$ se calculează tot pe componente:

$$d_{X_1\times\cdots\times X_n}((x_1, \cdots, x_n), (x_1', \cdots, x_n')) = max_{1\leq i\leq n}\{d_{X_i}(x_i, x_i')\}$$

(6) $d_H : \mathcal{P}(X) \times \mathcal{P}(X) \to [0,1]$ *(d_H este aşa-numita distanţă* Hausdorff*):*

$$d_H(U,V) = max\{sup_{u\in U} d(u,V), sup_{v\in V} d(v,U)\}$$

unde $d(u,W) = inf_{w\in W} d(u,w)$ *şi prin convenţie* $sup\emptyset = 0$, $inf\emptyset = 1$.

În continuare vom utiliza abrevierile $\mathcal{P}_{\text{co}}(\cdot)$ ($\mathcal{P}_{\text{nco}}(\cdot)$) pentru a ne referi la mulţimea submulţimilor compacte (nevide şi compacte) ale '$\cdot$'. În cazul spaţiului $\Sigma_{i\in\mathbb{N}} X_i$, în locul lui $\mathbb{N}$ se poate utiliza orice mulţime numărabilă. Pentru a uşura notaţia, uneori (atunci când semnificaţia indicelui i rezultă din context) pentru construcţia $\Sigma_{i\in\mathbb{N}} X_i$ scriem $\Sigma_i X_i$ sau chiar ΣX_i. In continuare, folosim conceptul de sumă infinită mai mult pentru a defini noţiunea de listă finită. Sperăm să nu apară nici o confuzie între notaţia $\Sigma_{i\in\mathbb{N}} X_i$ şi notaţia $\Sigma_{i\in\mathbb{N}} X^i$; în cazul notaţiei X_i simbolul i este utilizat ca indice; în cazul notaţiei X^i simbolul i specifică numărul de apariţii ale spaţiului X în produsul cartezian $X^i = X \times \cdots \times X$ (aici X apare de i ori). Construcţia $\Sigma_{i\in\mathbb{N}} X_i$ este mai generală decât $\Sigma_{i\in\mathbb{N}} X^i$; în particular fiecare X_i ar putea fi X^i. Are loc următoarea

Teorema 1.2.8 *Fie* $(X, d_X), (X_i, d_{X_i}), d_{\frac{1}{2}\cdot X}, d_{X_1\to X_2}, d_{X_1\times X_2}, d_{X_1+X_2}, d_{\Sigma_i X_i}$ *şi* d_H *ca în 1.2.7. Dacă* d_X, d_{X_i} *sunt ultrametrice, atunci şi* $d_{\frac{1}{2}\cdot X}, d_{X_1\to X_2}, d_{X_1\times X_2}, d_{X_1+X_2}, d_{\Sigma_i X_i}$, d_H *sunt ultrametrice. Dacă în plus* $(X, d_X), (X_i, d_{X_i})$ *sunt spaţii metrice complete, atunci* $\frac{1}{2}\cdot X$, $X_1 \to X_2$, $X_1 \xrightarrow{1} X_2$, $X_1 \times X_2, X_1 + X_2, \Sigma_i X_i$, $\mathcal{P}_{\text{co}}(X)$ *şi* $\mathcal{P}_{\text{nco}}(X)$ *(împreună cu metricile definite mai sus) sunt de asemenea spaţii metrice complete.*

În cele ce urmează vom omite adesea partea de metrică când ne referim la spaţiile metrice. În particular vom scrie $\frac{1}{2}\cdot X$ în loc de $(X, d_{\frac{1}{2}\cdot X})$. Pentru celelalte construcţii vom scrie $X_1 \to X_2, X_1 \times X_2, X_1 + X_2, \Sigma_i X_i$ şi $\mathcal{P}_{co}(X)$ $(\mathcal{P}_{nco}(X))$ şi vom subînţelege că ne referim la spaţiile metrice definite mai sus.

Ecuaţii de domeniu

Domeniile semantice utilizate în continuare sunt spaţii metrice complete. Domeniile semantice pot fi definite ca soluţii ale unor *ecuaţii de domeniu* [1]. Pentru definirea de spaţii compuse din spaţii mai simple se utilizează operaţiile de construcţie introduse în 1.2.7. Ecuaţiile de domeniu considerate sunt de forma:

$$\mathbb{X} \cong E$$

şi definesc spaţiul $\mathbb{X}$ ca fiind izometric (vezi 1.2.4(1)) cu E, unde E este construit din:

- spaţii metrice date, de obicei mulţimi arbitrare echipate cu metrica discretă,
- operaţiile de construire spaţii metrice compuse introduse în 1.2.7 şi
- posibile apariţii recursive ale spaţiului $\mathbb{X}$.

Spunem că spaţiul $\mathbb{X}$ definit astfel este *soluţia* ecuaţiei. O ecuaţie de domeniu poate să nu aibă nici o soluţie sau poate să aibă mai multe soluţii. Identificăm o clasă de ecuaţii de domeniu care au soluţie unică până la izometrie [1].[2] Facem acest lucru specificând o gramatică pentru E. Are loc teorema 1.2.9. Se remarcă faptul că în această teoremă este utilizată construcţia $\Sigma_i E^i$ pentru liste finite (bazată pe o sumă infinită de produse carteziene finite) iar nu construcţia mai generală pentru sume infinite arbitrare.

Teorema 1.2.9 *Fie*

$$E ::= \mathbb{A} \mid \tfrac{1}{2} \cdot \mathbb{X} \mid \mathcal{P}_{nco}(E) \mid \mathcal{P}_{co}(E) \mid E_1 \times E_2 \mid E_1 + E_2 \mid \Sigma_i E^i \mid E_1 \xrightarrow{1} E_2$$

unde $\mathbb{A}$ este un spaţiu ultrametric arbitrar, dar constant (care nu depinde de $\mathbb{X}$). Atunci

1. *Ecuaţia $\mathbb{X} \cong E$ are o soluţie unică (până la izometrie).*
2. *$\mathbb{X}$ este un spaţiu ultrametric complet.*

Observaţii 1.2.10

(1) *Orice mulţime echipată cu metrica discretă dă naştere unui spaţiu ultrametric complet (vezi exemplul 1.2.2(1)).*

(2) *Metrica pentru un spaţiu compus se obţine din metricile pentru spaţiile componente utilizând construcţiile date în 1.2.7.*

(3) *Atunci când este prezentă construcţia $E \xrightarrow{1} E$, se pot rezolva ecuaţii de forma:*

$$\mathbb{X} \cong ...\mathbb{X}... \xrightarrow{1} ...\mathbb{X}...$$

în care $\mathbb{X}$ apare în partea stângă a unui spaţiu de funcţii nonexpansiv. Asemenea domenii apar (în particular) atunci când se utilizează tehnica semanticii de continuare [1, 9].

(4) *În multe cazuri spaţiul E_1 dintr-o construcţie $E_1 \xrightarrow{1} E_2$ este o simplă mulţime (echipată cu metrica discretă). Deoarece în aceste cazuri orice spaţiu de funcţii este nonexpansiv, se poate scrie simplu $E_1 \to E_2$, în loc de $E_1 \xrightarrow{1} E_2$. Modul de rezolvare a unei ecuaţii de forma $\mathbb{X} \cong E$ în care E este dat de o gramatică în care regula $E \xrightarrow{1} E$ este simplificată la $\mathbb{A} \to E$ (unde $\mathbb{A}$ este un spaţiu constant) a fost introdus în [4] (soluţia în acest caz este un spaţiu metric complet). Cazul în care este prezentă construcţia $E \xrightarrow{1} E$ a fost soluţionat în [1].*

Între altele, în cele ce urmează vom utiliza domenii de forma:

[2] În [1] este prezentată o metoda generală de rezolvare a ecuaţiilor recursive (sau reflexive) de domeniu într-o categorie de spaţii metrice complete.

$\mathcal{P}_{nco}(A^{\infty})$

unde A este o mulţime oarecare, ale cărei elemente sunt de obicei utilizate pentru a modela *acţiunile atomice* ale limbajului dat. Aşa cum s-a explicat în 1.2.2(2), $A^{\infty} = A^{*} \cup A^{\omega}$ este mulţimea tuturor secvenţelor finite şi infinite de elemente din A. În aplicaţii, spaţiul A^{∞} este echipat cu metrica *Baire* (aşa cum s-a arătat în 1.2.2(2)). De asemenea, $\mathcal{P}_{nco}(A^{\infty})$ este echipat cu metrica *Hausdorff* (vezi 1.2.7(5)) indusă de metrica Baire definită peste A^{∞}. Construcţia A^{∞} poate fi definită printr-o ecuaţie de domeniu.

Definiţia 1.2.11 *Fie* $(x \in)X$ *un spaţiu nevid şi complet. Spaţiul* X^{∞} *poate fi definit astfel:*

$$X^{\infty} \cong \{\epsilon\} + (X \times \tfrac{1}{2} \cdot X^{\infty})$$

ϵ *modelează secvenţa vidă şi se utilizează o notaţie specifică pentru secvenţe. Astfel, în loc de* $(x_1, (x_2, ..., (x_n, \epsilon)...))$ *şi* $(x_1, (x_2, ...))$ *se scrie simplu* $x_1x_2...x_n$*, respectiv* $x_1x_2....$

Elementele domeniului $\mathcal{P}_{nco}(A^{\infty})$ sunt mulţimi nevide şi compacte de secvenţe. De exemplu, următoarele structuri sunt elemente ale $\mathcal{P}_{nco}(A^{\infty})$:

$$\{\epsilon\}, \{a_1a_2\}, \{\epsilon, a_1, a_1a_2, a_1a_2a_3\}, \{\epsilon, a, aa, ..., a^n, ..., a^{\omega}\}$$

(vezi 1.2.2(2) unde sunt explicate notaţiile a^n şi a^{ω}) dar următoarele două structuri nu sunt elemente ale $\mathcal{P}_{nco}(A^{\infty})$

$$\emptyset, \{\epsilon, a, a^2, ..., a^n, ...\}$$

deoarece $\emptyset$ este mulţimea vidă (iar $\mathcal{P}_{nco}(A^{\infty})$ conţine numai mulţimi nevide) iar $\{\epsilon, a, aa, ..., a^n, ...\}$ nu este o mulţime compactă (vezi 1.2.6) deoarece conţine secvenţa infinită: $a, a^2, ..., a^n, ...$ dar nu conţine limita (în raport cu metrica Baire a) acestei secvenţe care este a^{ω}.

Capitolul 2

Semantică de continuare pentru un limbaj secvenţial simplu $\mathcal{L}$

2.1 Sintaxa limbajului $\mathcal{L}$

Se consideră date o mulţime $(a \in)Act$ de *acţiuni atomice* şi o mulţime $(y \in)Y$ de variabile de procedură. Sintaxa limbajului $\mathcal{L}$ este dată în 2.1.1 în BNF. Recursivitatea este tratată utilizând conceptele de *declaraţie* şi *instrucţiune gardată*, urmând abordarea utilizată sistematic în [3]. Restricţia la recursivitate gardată asigură faptul că orice apel recursiv din corpul unei proceduri se poate face numai după executarea unei acţiuni atomice.

Definiţia 2.1.1

(a) (Instrucţiuni) $x(\in X) ::= a \mid y \mid x; x$

(b) (Instrucţiuni gardate) $g(\in G) ::= a \mid g; x$

(c) (Declaraţii) $(D \in)Decl = Y \to G$

(d) (Programe) $(\pi \in)\mathcal{L} = Decl \times X$

$(x \in)X$ este clasa (sintactică a) instrucţiunilor iar $(g \in)G$ este clasa instrucţiunilor gardate. ';' este operatorul sintactic de compunere secvenţială. O *declaraţie* $D(\in Decl)$ este o funcţie ce mapează variabile de procedură la instrucţiuni gardate. Un program este o pereche constând dintr-o declaraţie şi o instrucţiune.

Observaţia 2.1.2 *Definiţiile semantice pot utiliza drept parametru explicit o declaraţie. În abordarea clasică, recursivitatea în semantica denotaţională este definită pe baza unui argument de punct fix, care implică utilizarea unui mediu de proceduri (engl.* semantic environment*). În semantica metrică [3] este urmată de obicei o abordare mai simplă, în care definiţiile inductive, inclusiv definiţia (compoziţională) a semanticii denotaţionale se face nu prin inducţie structurală, ci prin inducţie după o măsură de complexitate care scade la apelul recursiv. Această strategie depinde de restricţia la recursivităţea gardată. Abordarea este utilizată sistematic în [3].*

În cele ce urmează, vom urma şi noi această abordare. De asemenea, pentru simplitate şi fără a reduce din generalitate, vom lucra cu o declaraţie de procedură fixă D şi toate consideraţiile se vor referi la acest D fixat.

Măsura de complexitate utilizată pentru $\mathcal{L}$ în definiţii şi raţionamente inductive este dată mai jos.

Definiţia 2.1.3 *(Măsură de complexitate)* $c : X \to \mathbb{N}$ *este definită astfel:*

$$\begin{array}{rcl} c(a) & = & 1 \\ c(y) & = & 1 + c(D(y)) \\ c(x_1; x_2) & = & 1 + c(x_1) \end{array}$$

2.2 Domeniul pentru semantica denotaţională

În acest capitol proiectăm o semantică de continuare pentru $\mathcal{L}$, utilizând continuări structurate [9, 10]. În cazul $\mathcal{L}$ o continuare este o stivă de calcule (denotaţii).

$$\begin{array}{rcl} (\phi \in)\mathbb{D} & \cong & \mathbb{K} \stackrel{1}{\to} \mathbb{P} \\ (\kappa \in)\mathbb{K} & = & \{nil\} + \Sigma_{i\in\mathbb{N}}\mathbb{F}^i \qquad (= \{nil\} + \mathbb{F} + \mathbb{F}^2 + \cdots) \\ \mathbb{F} & = & \frac{1}{2} \cdot \mathbb{D} \\ \mathbb{P} & = & Act^\infty \qquad (Act^\infty \cong \{\epsilon\} + (Act \times (\frac{1}{2} \cdot Act^\infty))) \end{array}$$

În această definiţie se presupune că mulţimea Act este echipată cu metrica discretă, iar spaţiile compuse sunt construite utilizând metricile din definiţia 1.2.7. Am notat cu $\mathbb{F}^n$ produsul cartezian $\mathbb{F} \times \cdots \times \mathbb{F}$, în care domeniul $\mathbb{F}$ apare de n ori. De exemplu, $\mathbb{F}^4 = \mathbb{F} \times \mathbb{F} \times \mathbb{F} \times \mathbb{F}$. Elementele spaţiului $\mathbb{F}^n$ sunt n-tuple de forma $(\phi_1, \cdots, \phi_n)$, pentru $n > 0$. Utilizăm următoarea *convenţie de notaţie*:

$$\phi : (\phi_1, \cdots, \phi_n)$$

este o notaţie alternativă pentru $(n + 1)$-tupla:

$$(\phi, \phi_1, \cdots, \phi_n) \quad (\in \mathbb{F}^{n+1})$$

Aici, $n > 0$. De asemenea,

$$\phi : nil$$

este o notaţie alternativă pentru tupla de lungime 1:

$$(\phi) \quad (\in \mathbb{F})$$

2.3 Funcţia semantică denotaţională

Definiţia 2.3.1 *(Semantică denotaţională pentru $\mathcal{L}$)*

(a) Fie $kc : \mathbb{K} \to \mathbb{P}$,

$$\begin{array}{rcl} kc(nil) & = & \epsilon \\ kc(\phi : \kappa) & = & \phi\kappa \end{array}$$

(b) Definim $\Psi : Sem_D \to Sem_d$, *unde* $(S \in) Sem_d = X \to \mathbb{D}$:

$$\begin{array}{rcl} \Psi(S)(a)(\kappa) & = & a \cdot kc(\kappa) \\ \Psi(S)(y)(\kappa) & = & \Psi(S)(D(y))(\kappa) \\ \Psi(S)(x_1; x_2)(\kappa) & = & \Psi(S)(x_1)(S(x_2) : \kappa) \end{array}$$

(c) Definim $[\![\cdot]\!] : X \to \mathbb{D}, [\![\cdot]\!] = fix(\Psi)$. *De asemenea, punem* $\mathcal{D}[\![\cdot]\!] : X \to \mathbb{P}$

$$\mathcal{D}[\![x]\!] = [\![x]\!](nil)$$

Această definiţie este justificată în mod riguros în continuare.

Lemma 2.3.2 *Fie* $\kappa_1, \kappa_2 \in \mathbb{K}$.

(a) Dacă $\kappa_1 \in \mathbb{F}^{n_1}, \kappa_2 \in \mathbb{F}^{n_2}$ *şi* $n_1 \neq n_2$ *atunci* $d_{\mathbb{K}}(\kappa_1, \kappa_2) = 1$

(b) Dacă $\kappa_1 = (\phi_1^1, \cdots, \phi_1^n), \kappa_2 = (\phi_2^1, \cdots, \phi_2^n) \in \mathbb{F}^n$ *atunci:*

$$d_{\mathbb{K}}(\kappa_1, \kappa_2) = \frac{1}{2} \cdot max_{1 \leq i \leq n}\ d_{\mathbb{D}}(\phi_1^i, \phi_2^i)$$

Demonstraţie 2.3.2(a) este evidentă în baza definiţiei reuniunii disjuncte Σ. Pentru 2.3.2(b) avem:

$$d_{\mathbb{K}}(\kappa_1, \kappa_2) = d_{\mathbb{K}}((\phi_1^1, \cdots, \phi_1^n), (\phi_2^1, \cdots, \phi_2^n))$$

$$= max_{1 \leq i \leq n} d_{\mathbb{F}}(\phi_1^i, \phi_2^i) = \frac{1}{2} \cdot max_{1 \leq i \leq n} d_{\mathbb{D}}(\phi_1^i, \phi_2^i)$$

□

Lemma 2.3.3

(a) Funcţia kc este bine definită ($\forall \kappa \in \mathbb{K}, kc(\kappa) \in \mathbb{P}$).

(b) Pentru orice $\kappa_1, \kappa_2 \in \mathbb{K}$: $d_{\mathbb{P}}(kc(\kappa_1), kc(\kappa_2)) \leq 2 \cdot d_{\mathbb{K}}(\kappa_1, \kappa_2)$.

Demonstraţie Pentru 2.3.3(a) remarcăm că $kc(nil) = \epsilon \in \mathbb{P}$ şi $kc(\phi : \kappa) = \phi\kappa \in \mathbb{P}$. Pentru 2.3.3(b) identificăm 3 subcazuri.

- Dacă $\kappa_1 = nil$ şi $\kappa_2 = nil$ atunci $d_{\mathbb{P}}(kc(\kappa_1), kc(\kappa_2)) = d_{\mathbb{P}}(\epsilon, \epsilon) = 0 \leq 2 \cdot d_{\mathbb{K}}(\kappa_1, \kappa_2)$.

- Dacă $\kappa_1 \in \mathbb{F}^{n_1}, \kappa_2 \in \mathbb{F}^{n_2}$ şi $n_1 \neq n_2$ atunci $d_{\mathbb{K}}(\kappa_1, \kappa_2) = 1$ şi rezultatul dorit urmează imediat.

- Dacă $\kappa_1 = (\phi_1^1 : \kappa_1')$, $\kappa_2 = (\phi_2^1 : \kappa_2')$, $\kappa_1' = (\phi_1^2, \cdots, \phi_1^n), \kappa_2' = (\phi_2^2, \cdots, \phi_2^n)$, deci $\kappa_1, \kappa_2 \in \mathbb{F}^n, n > 0$, atunci calculăm astfel:

  $d_{\mathbb{P}}(kc(\kappa_1), kc(\kappa_2)) = d_{\mathbb{P}}(\phi_1^1 \kappa_1', \phi_2^1 \kappa_2')$
  $\leq max\{d_{\mathbb{P}}(\phi_1^1 \kappa_1', \phi_1^1 \kappa_2')^{(*)}, d_{\mathbb{P}}(\phi_1^1 \kappa_2', \phi_2^1 \kappa_2')^{(**)}\}$
  Avem
  $^{(*)} \leq d_{\mathbb{K}}(\kappa_1', \kappa_2')$ [ϕ_1^1 este nonexpansivă, $\phi_1^1 \in \mathbb{K} \xrightarrow{1} \mathbb{P}$]
  $\leq d_{\mathbb{K}}(\kappa_1, \kappa_2)$ [definiţia metricii pentru produsul cartezian]
  şi
  $^{(**)} \leq d_{\mathbb{D}}(\phi_1^1, \phi_2^1)$ [definiţia distanţei între funcţii]
  $\leq max_{1 \leq i \leq n} d_{\mathbb{D}}(\phi_1^i, \phi_2^i) = 2 \cdot d_{\mathbb{K}}(\kappa_1, \kappa_2)$ [lemma 2.3.2(b)]

□

Lemma 2.3.4 *Pentru orice $S \in Sem_D, x \in X, \kappa \in \mathbb{K}$:*

(a) $\Psi(S)(x)(\kappa) \in \mathbb{P}$ *(este bine definită)*

(b) $\Psi(S)(x)$ *este nonexpansivă (în κ)*

(c) Ψ *este $\frac{1}{2}$-contractivă (în S).*

Demonstraţie Tratăm toate cele trei proprietăţi, 2.3.4(a), (b) şi (c), prin inducţie după $c(x)$. Începem cu 2.3.4(a). Folosim simbolul $\equiv$ pentru identitatea sintactică.

Cazul $x \equiv a$. $\Psi(S)(a)(\kappa) = a \cdot kc(\kappa) \in \mathbb{P}$, utilizând aici şi lemma 2.3.3(a).

Cazul $x \equiv y$. $\Psi(S)(y)(\kappa) = \Psi(S)(D(y))(\kappa) \in \mathbb{P}$, prin ipoteza inducţiei, deoarece $c(y) > c(D(y))$.

Cazul $x \equiv x_1; x_2$. $\Psi(S)(x_1; x_2)(\kappa) = \Psi(S)(x_1)(S(x_2) : \kappa)$. Deoarece $S \in Sem_D$ rezultă că $S(x) \in \mathbb{D}$ şi urmează că $(S(x) : \kappa) \in \mathbb{K}$. Acum se poate utiliza ipoteza inducţiei ($c(x_1) < c(x_1; x_2)$) şi rezultă imediat că $\Psi(S)(x_1)(S(x_2) : \kappa) \in \mathbb{P}$.

În continuare demonstrăm 2.3.4(b). Arătăm prin inducţie după $c(x)$ că pentru orice $S \in Sem_D, x \in X$ şi $\kappa_1, \kappa_2 \in \mathbb{K}$ avem $d(\Psi(S)(x)(\kappa_1), \Psi(S)(x)(\kappa_2)) \leq d(\kappa_1, \kappa_2)$.

Cazul $x \equiv a$.

$d(\Psi(S)(a)(\kappa_1), \Psi(S)(a)(\kappa_2)) = d(a \cdot kc(\kappa_1), a \cdot kc(\kappa_2))$
$= \frac{1}{2} \cdot d(kc(\kappa_1), kc(\kappa_2))$
$\leq d(\kappa_1, \kappa_2)$ [Lemma 2.3.3(b)]

Cazul $x \equiv y$.

$$d(\Psi(S)(y)(\kappa_1), \Psi(S)(y)(\kappa_2))$$
$$d(\Psi(S)(D(y))(\kappa_1), \Psi(S)(D(y))(\kappa_2))$$
$$\leq d(\kappa_1, \kappa_2) \quad \text{[prin ipoteza inducţiei } (c(y) > c(D(y)))\text{]}$$

Cazul $x \equiv x_1; x_2$.

$$d(\Psi(S)(x_1; x_2)(\kappa_1), \Psi(S)(x_1; x_2)(\kappa_2))$$
$$= d(\Psi(S)(x_1)(S(x_2) : \kappa_1), \Psi(S)(x_1)(S(x_2) : \kappa_2))$$
$$\leq d(S(x_2) : \kappa_1, S(x_2) : \kappa_2) \quad \text{[ipoteza inducţiei } (c(x_1; x_2) > c(x_1))\text{]}$$
$$\leq d(\kappa_1, \kappa_2) \quad \text{[definiţia distanţei pentru produsul cartezian]}$$

Pentru punctul 2.3.4(c) procedăm tot prin inducţie după $c(x)$. Arătăm că pentru orice $S_1, S_2 \in Sem_D, x \in X$ şi $\kappa \in \mathbb{K}$ avem $d(\Psi(S_1)(x)(\kappa), \Psi(S_2)(x)(\kappa)) \leq \frac{1}{2} \cdot d(S_1, S_2)$.

Cazul $x \equiv a$.

$$d(\Psi(S_1)(a)(\kappa), \Psi(S_2)(a)(\kappa)) = d(a \cdot kc(\kappa), a \cdot kc(\kappa)) = 0$$
$$\leq \tfrac{1}{2} \cdot d(S_1, S_2)$$

Cazul $x \equiv y$.

$$d(\Psi(S_1)(y)(\kappa), \Psi(S_2)(y)(\kappa))$$
$$= d(\Psi(S_1)(D(y))(\kappa), \Psi(S_2)(D(y))(\kappa))$$
$$\leq \tfrac{1}{2} \cdot d(S_1, S_2) \quad \text{[ipoteza inducţiei } (c(y) > c(D(y)))\text{]}$$

Cazul $x \equiv x_1; x_2$.

$$d(\Psi(S_1)(x_1; x_2)(\kappa), \Psi(S_2)(x_1; x_2)(\kappa))$$
$$= d(\Psi(S_1)(x_1)(S_1(x_2) : \kappa), \Psi(S_2)(x_1)(S(x_2) : \kappa))$$
$$\leq max\{d(\Psi(S_1)(x_1)(S_1(x_2) : \kappa), \Psi(S_1)(x_1)(S_2(x_2) : \kappa))^{(+)},$$
$$d(\Psi(S_1)(x_1)(S_2(x_2) : \kappa), \Psi(S_2)(x_1)(S_2(x_2) : \kappa))^{(++)}\}$$

Pentru $^{(+)}$ şi $^{(++)}$ calculăm astfel:

$$^{(+)} \leq d(S_1(x_2) : \kappa, S_2(x_2) : \kappa) \quad \text{[Lemma 2.3.4(b)]}$$
$$= \tfrac{1}{2} \cdot d(S_1(x_2), S_2(x_2)) \quad \text{[Lemma 2.3.2(b)]}$$
$$\leq \tfrac{1}{2} \cdot d(S_1, S_2) \quad \text{[definiţia distanţei pentru spaţiul de funcţii]}$$
$$^{(++)} \leq \tfrac{1}{2} \cdot d(S_1, S_2) \quad \text{[prin ipoteza inducţiei } (c(x_1; x_2) > c(x_1))\text{]}$$

□

Reproducem mai jos definiţia funcţiei $[\![\cdot]\!] : X \to \mathbb{D}$.

$$\begin{array}{rcl} [\![a]\!](\kappa) & = & a \cdot kc(\kappa) \\ [\![y]\!](\kappa) & = & [\![D(y)]\!](\kappa) \\ [\![x_1; x_2]\!](\kappa) & = & [\![x_1]\!]([\![x_2]\!] : \kappa) \end{array}$$

2.4 Implementare Haskell a semanticii denotaţionale

Limbajul Haskell [12] furnizează funcţii de ordin superior şi evaluare leneşă. Acest limbaj este un instrument util în implementarea funcţiilor denotaţionale. Prin implementarea directă în Haskell a unei descrieri formale denotaţionale se poate obţine un prototip executabil (în general ineficient dar) corect pentru limbajul studiat.

Sintaxa $\mathcal{L}$

```
type Act = String
type Y   = String

data X    = A Act | Y Y | Seq X X
type Decl = Y -> X
```

Domenii semantice

```
type D = K -> P
data F = D D
type K = [F]
type P = [Act]
```

Funcţia denotaţională

```
kc :: K -> P
kc []       = []
kc (D d:k) = d k

sem :: X -> D
sem (A a)        k = a : kc k
sem (Y y)        k = sem (decl y) k
sem (Seq x1 x2) k = sem x1 ((D (sem x2)) : k)

den :: X -> P
den x = sem x []
```

Suport pentru testarea interpretorului semantic

Pentru testarea interpretorului semantic se consideră o declaraţie fixata `decl :: Decl` şi instrucţiunile `x1,x2 :: X`.

```
decl :: Decl
decl "y1" = Seq (A "a1") (Seq (A "a2") (Y "y2"))
decl "y2" = Seq (A "a3") (A "a4")
decl "y"  = Seq (A "a") (Y "y")
```

```
x1,x2 :: X
x1 = Seq (A "a0") (Y "y1")
x2 = Seq (A "a0") (Y "y")
```

Utilizând un sistem Haskell cum este GHC [12], se pot efectua următoarele experimente:

```
Main> den x1
["a0","a1","a2","a3","a4"]
Main> den x2
["a0","a","a","a",...
```

Capitolul 3

Semantică directă pentru $\mathcal{L}$

Acest model denotaţional este prezentat în capitolul 1 din [3].

Bibliografie

[1] P. America and J.J.M.M. Rutten. Solving reflexive domain equations in a category of complete metric spaces. *J. of Comput. System Sci.*, 39:343–375, 1989.

[2] S. Banach. Sur les operations dans les ensembles abstraits et leurs applications aux equations integrales. *Fundamenta Mathematicae* 3:133–181, 1922.

[3] J.W. de Bakker, E.P. de Vink. *Control flow semantics*, MIT Press, 1996.

[4] J.W. de Bakker, J.I. Zucker. Processes and the denotational semantics of concurrency. *Information and Control* 54:70–120, 1982.

[5] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, D. S. Scott. *Continuous lattices and domains.* Cambridge University Press, 2003.

[6] M. Nivat. Infinite words, infinite trees, infinite computations. In J.W. de Bakker and J. van Leeuwen, editors, *Foundations of Computer Science III, part 2: Languages, Logic, Semantics*, 109:3–52, Mathematical Centre Tracts, Amsterdam, 1979.

[7] D.S. Scott. Outline of a mathematical theory of computation. In *Proc. 4th Annual Princetown Conference on Inf. Sciences and Systems*, pages 169–176, Princetwon, 1970.

[8] D.S. Scott and C. Stratchey. Toward a mathematical semantics for computer languages. In *Proc. Symp. on Computers and Automata*, 21:19–46, Microwave Research Institute Symposia Series, New York, 1971.

[9] E.N. Todoran. Metric semantics for synchronous and asynchronous communication: a continuation-based approach. *Electronic Notes in Theoretical Computer Science (ENTCS)* 28:119–146, Elsevier, 2000.

[10] E.N. Todoran, N. Papaspyrou. Continuations for prototyping concurrent languages. Technical Report CSD-SW-TR-1-06, National Technical University of Athens, School of Electrical and Computer Engineering, Software Engineering Laboratory, 2006, `http://www.softlab.ntua.gr/research/techrep/CSD-SW-TR-1-06.pdf`

[11] F. Turbak, D. Gifford. Design concepts in programming languages. MIT Press, 2008.

[12] `www.haskell.org`