Poglavje 5

DENOTACIJSKA SEMANTIKA

Denotacijska semantika uporablja bolj abstraktno definicijo pomena od operacijske semantike: namesto sekvence stanj stroja je pomen izrazov definiran z matematičnimi objekti kot so na primer števila, funkcije, itd.

Definicija denotacijske semantike za nek konkreten jezik zahteva najprej definicijo *domen* in nato definicijo *interpretacijskih funkcij*, ki preslikujejo izraze programskega jezika v elemente domen.

Iskanje primernih domen, matematičnih struktur, ki se jih uporablja za definicijo denotacijske semantike se je razvilo v področje, ki ga imenujemo *teorija domen*. Kratek uvod v teorijo domen bo predstavljen v naslednji sekciji.

Teorija domen preučuje prostore, ki jih sestavljajo izrazi nekega jezika. Evaluacijo programov vidimo kot sekvenco uporab pravil na stavku jezika. Sekvence instanc pravil tvorijo verige, ki so končne.

Zaključitev programov obravnavamo s spodnjo mejo in urejenostjo. Rekurzivne programe prevedemo v rekurzivne enačbe in fiksne točke. Prostori vrednosti so predstavljeni s specifično domeno.

Domena je kompletna delna urejenost (cpo, angl. complete partial order). Gledano iz stališča čiste abstraktne predstavitve izvajanja programov je teorija domen in denotacijska semantika, ki sloni na njej, bolj direkten način za opis semantike.

Stavki nas zanimajo samo v oviru ekvivalence $\sim$.

$$c_0 \sim c_1 \iff \begin{array}{l}\{(\sigma, \sigma') \mid \langle c_0, \sigma \rangle \rightarrow \sigma'\} = \\ \{(\sigma, \sigma') \mid \langle c_1, \sigma \rangle \rightarrow \sigma'\}\end{array}$$

Z drugimi besedami $c_0 \sim c_1$, če stavka določata isto parcialno funkcijo na stanjih.

V primerjavi z operacijsko semantiko denotacijska semantika nudi bolj abstrakten opis jezika in semantike. Stavek programskega jezika je predstavljen z interpretacijo stavka, z vsemi vrednostmi, ki jih lahko stavek zavzame. Interpretacijo stavka imenujemo *denotacija*, ki je predstavljena s parcialno funkcijo na stanjih.

5.1 Domene

V tej sekciji bo predstavljen uvod v matematično teorijo, *teorijo domen*, ki med drugim nudi formalno ogrodje za konstrukcijo najmanjših fiksnih točk. Konstrukcije bomo potrebovali pri definiciji denotacijske semantike in pri študiju različnih gradnikov (konceptov) programskih jezikov.

Teorija domen opisuje stavke jezika bolj iz matematičnega vidika: preučuje urejenost izrazov z uporabo klasičnih matematičnih orodij kot so npr. relacije, množice in funkcije. Objekti, ki jih preučuje predstavljajo abstrakcije stavkov programskega jezika– objekti teorije domen so bolj splošni izrazi, ki jih lahko apliciramo na matematičnih objektih kot tudi na sorodnih jezikih.

5.1.1 Delne urejenosti

Teorija domen uporablja *delno urejene množice*, ki zadoščajo izbranim lastnostim. Delno urejene množice so podane z naslednjo definicijo. Množico D imenujemo *osnovno množico* delne urejenosti $(D, \sqsubseteq)$. Večinoma bomo imenovali osnovno množico D kar delna urejenost; relacijo $\sqsubseteq$ bomo uporabljali za različne delno urejene množice.

Definicija 5.1.1 *Binarna relacija $\sqsubseteq$ nad množico D je delna urejenost, čče je:*

1. *refleksivna:* $\forall d \in D.d \sqsubseteq d$
2. *tranzitivna:* $\forall d, d', d'' \in D.d \sqsubseteq d' \wedge d' \sqsubseteq d'' \Rightarrow d \sqsubseteq d''$
3. *antisimetrična:* $\forall d, d' \in D.d \sqsubseteq d' \wedge d' \sqsubseteq d \Rightarrow d = d'$ □

Par $(D, \sqsubseteq)$ *imenujemo* delno urejena množica.

Definicija 5.1.2 (Najmanjši element) *Naj bo D delno urejena množica in $S \subseteq D$. Element $d \in S$ je* najmanjši *element, če zadošča naslednjem pogoju.*

$$\forall x \in S.d \sqsubseteq x.$$

Minimalen element bomo imenovali dno *in označili z $\perp_D$ ali samo $\perp$, če je D jasna iz konteksta.* □

Ker je $\sqsubseteq$ anti-simetrična, ima S največ en najmanjši element. Seveda najmanši element množice ni potrebno, da obstaja. Na primer, $\mathbb{Z}$ nima najmanjšega elementa.

Definicija 5.1.3 (Veriga) *Končna naraščajoča veriga D je sekvenvca elementov D, ki zadoščajo naslednjem pogoju.*

$$d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \dots$$

Definicija 5.1.4 (Najmanjša zgornja meja) *Zgornja meja verige $d_0 \sqsubseteq d_1 \sqsubseteq \dots$ je $d \in D$ za katerega velja $\forall n \in \mathbb{N}.d_n \sqsubseteq d$. Če obstaja najmanjša zgornja meja (okr. lub), potem je določena z:*

$$\bigsqcup_{n \geq 0} d_n$$

Poglejmo zakaj je najmanjša zgornja meja dobro definirana. Po definiciji velja $\forall m \in \mathbb{N}.d_m \sqsubseteq \bigsqcup_{n \geq 0} d_n$. Prav tako velja, da vse druge zgornje meje nad najmanjšo zgornjo mejo: $\forall d \in D$, če $\forall m \in \mathbb{N}.d_m \sqsubseteq d$, potem $\bigsqcup_{n \geq 0} d_n \sqsubseteq d$.

Opazka 5.1.1 *Sledi predstavitev nekaterih lastnosti prej definiranih verig in najmanše zgornje meje.*

(i) Ne bomo se ukvarjali z neskončnimi in padajočimi verigami: "veriga" bo vedno pomenila naraščajočo in končno verigo.

(ii) Elementi verige ni nujno, da so različni. Pravimo, da je veriga $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \dots$ preide v konstanto*, če za nek $N \in \mathbb{N}$ velja $\forall n \geq N.d_n = d_N$. V tem primeru velja $\bigsqcup_{n \geq 0} d_n = d_N$.*

(iii) Kot v primeru najmanjšega elementa poljubne podmnožice delno urejene množice, je najmanjša zgornja meja verige unikatna, če obstaja. Ni pa nujno, da obstaja, npr. veriga $0 \leq 1 \leq 2 \leq 3 \leq \dots$ nima zgornje meje.

(iv) Če izpustimo končno mnogo elementov iz začetka verige se zgornje meja ohranjajo in s tem tudi lub.

$$\bigsqcup_{n \geq 0} d_n = \bigsqcup_{n \geq 0} d_{N+n}, \text{ za vse } N \in \mathbb{N}$$

□

Ukvarjali se bomo z delno urejenimi množicami, ki zadoščajo nekarim lastnostim polnosti: vsaka veriga bo imela najmanjšo zgornjo mejo (lub1+lub2).

Definicija 5.1.5 (ω-polna delno urejena množica; angl. "cpo" t.j. "chain complete poset") *ω-polna delno urejena množica $(D, \sqsubseteq)$ vsebuje samo naraščajoče verige $d_0 \sqsubseteq d_1 \sqsubseteq d_2, \dots$, ki imajo najmanjšo zgornjo mejo $\bigsqcup_{n \geq 0} d_n$:*

(lub1) $\forall m \in \mathbb{N}.d_m \sqsubseteq \bigsqcup_{n \geq 0} d_n$

(lub2) $\forall d \in D.(\forall m \geq 0.d_m \sqsubseteq d) \Rightarrow \bigsqcup_{n \geq 0} d_n \sqsubseteq d$

Pogoj (lub1) zagotavlja, da najmanjša spodnja meja večja od vseh ostalih elementov (verig). Pogoj (lub2) zagotavlja, da je najmanjša spodnja meja manjša od vseh drugih najmanjših spodnjih mej.

V literaturi je več različnih definicij *domen*, ki zadoščajo različnim lastnostim. Tu bomo uporabili "minimalno" definicijo: zahtevali bomo, da cpo vsebuje dno.

Definicija 5.1.6 (Domena) *Domena je cpo, ki vsebuje dno* $\perp$.

$$\forall d \in D. \perp \sqsubseteq d$$

Primer 5.1.1 *V tem primeru si bomo ogledali parcialne funkcije* $X \rightharpoonup Y$. *Domena funkcije* f *je* $dom(f) \subseteq X$ *in zaloga vrednosti je* Y.

Delno urejenost definiramo na osnovi domene in zaloge vrednosti f*:* $f \sqsubseteq g$*, čče* $dom(f) \subseteq dom(g)$ *in* $\forall x \in dom(f). f(x) = g(x)$.

Najmanjša zgornja meja verige $f_o \sqsubseteq f_1 \sqsubseteq f_2 \sqsubseteq ...$ *je parcialna funkcija* f *z domeno* $dom(f) = \bigcup_{n \geq 0} dom(f_n)$ *in zalogo vrednosti:*

$$f(x) = \begin{cases} f_n(x) & x \in dom(f_n) \text{ za nek } n \\ \perp_Y & sicer \end{cases}$$

Definicija funkcije f *se ujema z izračunom lub verige* $f_0 \sqsubseteq f_1 \sqsubseteq f_2 \sqsubseteq \ldots$ *:* $f_{lub} = \bigsqcup_{n \geq 0} f_n$. *Domena funkcije* f_{lub} *je* $\bigcup_{n \geq 0} dom(f_n)$ *in zaloga vrednosti je definirana* $\forall x \in dom(f_i). f_i(x) = f_j(x)$ *za vse* $i \leq j \in \mathbb{N}$. *Velja torej* $f_{lub} = f$.

Na koncu definiramo še $\perp_f$*, ki predstavlja nedefinirano parcialno funkcijo.* □

Primer 5.1.2 *Vsaka delna urejenost* $(D, \sqsubseteq)$ *katere množica* D *je končna je* ω*-polna delna urejenost. V taki množici je vsaka veriga končna. Ni pa potrebno, da ima takšna množica najmanjši element.* □

Primer 5.1.3 *V tem primeru bomo definirali dve domene naravnih števil* $\mathbb{N}$. *Prvo imenujemo* ravninska *naravna števila,* $\mathbb{N}_\perp$*:*

$$\begin{array}{ccccccc} 1 & 2 & 3 & \ldots & n & n+1 & \ldots \\ \\ & & & \perp & & & \end{array}$$

Druga domena predstavlja navpična *naravna števila,* Ω*:*

$$\begin{array}{c} \omega \\ n+1 \\ n \\ \vdots \\ 2 \\ 1 \\ 0 \end{array}$$

□

Poglejmo si zdaj nekaj primerov delno urejenih množic, ki so oz. niso cpo.

Primer 5.1.4 *Množica naravnih števil* $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ *opremljena z relacijo* $\leq$ *ni* ω*-polna delna urejenost. Naraščajoča veriga* $1 \leq 2 \leq 3 \leq 4 \leq \ldots$ *nima zgornje meje.* □

Primer 5.1.5 *Poglejmo si zdaj primer delne urejenosti konstruirane iz* $\mathbb{N} \cup \{\perp\}$*. Za vsako naravno število velja* $\forall n \in \mathbb{N}. \perp \sqsubseteq n$*. Naravna števila med sabo niso v relaciji* $\sqsubseteq$*. Tako domeno imenujemo* ravninska naravna števila. □

Primer 5.1.6 *Množico naravnih števil* $\mathbb{N}$ *lahko uredimo tudi drugače. Množico* $\mathbb{N}$ *razširimo z elementom* ω*, ki predstavlja zgornjo mejo* $\mathbb{N}$*. Relacijo* $\sqsubseteq$ *lahko zdaj definiramo na sledeč način:*

$$d \sqsubseteq d' \iff \begin{cases} & d, d' \in \mathbb{N} \wedge d \leq d' \\ ali & d \in \mathbb{N} \wedge d' = \omega \\ ali & d = d' = \omega \end{cases}$$

Naraščajoča veriga $1 \leq 2 \leq 3 \leq$ *... iz* D *ima tako vedno zgornjo mejo* ω *in* D *je* ω*-polna delna urejenost.* □

Primer 5.1.7 *Poglejmo si še eno možno definicijo delne urejenosti na osnovi naravnih števil. Tokrat razširimo* $\mathbb{N}$ *z dvema zgornjima mejama* $\{\omega_1, \omega_2\}$*. Relacijo* $\sqsubseteq$ *lahko zdaj definiramo na sledeč način:*

$$d \sqsubseteq d' \iff \begin{cases} & d, d' \in \mathbb{N} \wedge d \leq d' \\ ali & d \in \mathbb{N} \wedge d' \in \{\omega_1, \omega_2\} \\ ali & d = d' = \omega_1 \\ ali & d = d' = \omega_2 \end{cases}$$

Naraščajoča veriga $1 \leq 2 \leq 3 \leq$ *... iz* D *ima tako dve zgornji meji* ω_1 *in* ω_2*. Nima pa najmanjše zgornje meje, ker* $\omega_1 \not\sqsubseteq \omega_2$ *in* $\omega_2 \not\sqsubseteq \omega_1$*.* D *torej ni* ω*-polna delna urejenost.* □

Definicija 5.1.7 (Monotona funkcija) *Funkcija* $f : D \to E$ *med* ω*-polnimi delnimi urejenostmi je* monotona*, čče*

$$\forall d, d' \in D. d \sqsubseteq d' \Rightarrow f(d) \sqsubseteq f(d')$$

.

Definicija 5.1.8 (Zvezna funkcija) *Funkcija $f : D \to E$ med ω-polnimi delnimi urejenostmi je* zvezna, *čče je monotona in ohranja najmanjše zgornje meje verig: za vse verige $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \ldots$ iz D velja*

$$f(\bigsqcup_{n\geq 0} d_n) = \bigsqcup_{n\geq 0} f(d_n) \text{ v } E.$$

Definicija 5.1.9 (Striktna funkcija) *Funkcija $f : D \to E$ med ω-polnimi delnimi urejenostmi je* striktna, *čče $f(\bot) = \bot$.*

Opazka 5.1.2 *V primeru, da je $f : D \to E$ monotona in imamo verigo $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \ldots$ v D, potem z aplikacijo f dobimo verigo $f(d_0) \sqsubseteq f(d_1) \sqsubseteq f(d_2) \sqsubseteq \ldots$ v E.*

Še več, v primeru, da je d zgornja meja prve verige, je $f(d)$ zgornja meja druge verige in hkrati tudi večja od najmanjše zgornje meje te verige. Če zapišemo to v matematični notaciji dobimo:

$$\bigsqcup_{n\geq 0} f(d_n) \sqsubseteq f(\bigsqcup_{n\geq 0} d_n)$$

Če zdaj uporabimo antisimetričnost $\sqsubseteq$ lahko pokažemo, da je funkcija f med ω-polnimi delnimi urejenostmi zvezna, *tako da za vsako verigo $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \ldots$ iz E preverimo ali velja:*

$$f(\bigsqcup_{n\geq 0} d_n) \sqsubseteq \bigsqcup_{n\geq 0} f(d_n).$$

□

Primer 5.1.8 *Dana sta cpo D in E, za vsak $e \in E$ je enostavno pokazati, da je funkcija $D \to E$, ki ima konstantno vrednost $\lambda d \in D.\, e$, je* zvezna.

Primer 5.1.9 *Naj bo Ω domena vertikalnih naravnih števil, kot je prikazano na Sliki 8.1.1. Funkcija $f : \Omega \to \Omega$ je definirana takole:*

$$f(n) = \begin{cases} n \in \mathbb{N} & f(n) = 0 \\ n = \omega & f(n) = \omega \end{cases}$$

Funkcija je monotona in striktna, ni pa zvezna ker velja:

$$f(\bigsqcup_{n\geq 0} d_n) = f(\omega) = \omega \neq 0 = \bigsqcup_{n\geq 0} 0 = \bigsqcup_{n\geq 0} f(n)$$

□

5.1.2 Tarskijev izrek o fiksni točki

Fiksna točka funkcije $f : D \to D$ je po definiciji $d \in D$, ki zadošča $f(d) = d$. Poglejmo si najprej šibkejši primer *pre-fiksne* točke.

Definicija 5.1.10 (Pre-fiksna točka) *Naj bo D delno urejena množica in naj bo $f : D \to D$ funkcija. Element $d \in D$ je pre-fiksna točka f, če zadošča $f(d) \sqsubseteq d$.* □

Najmanjšo pre-fiksno točko funkcije f bomo označili $\mathrm{fix}(f)$.

Definicija 5.1.11 (Najmanjša fiksna točka) *Naj bo D delno urejena množica in naj bo $f : D \to D$ funkcija. Najmanjša fiksna točka funkcije f, $fix(f)$, mora zadoščati naslednjim pogojem:*

lfp1 $f(fix(f)) \sqsubseteq fix(f)$

lfp2 $\forall d \in D. f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$. □

Opazka 5.1.3 *(lfp1) $fix(f)$ je tudi pre-fiksna točka, kar pomeni, da se evaluacija f ustavi: $f(fix(f)) \sqsubseteq fix(f)$.*

(lfp2) Vsaka pre-fiksna točka je večja od fiksne točke $fix(f)$: fiksna točka $fix(f)$ je najmanjša pre-fiksna točka. □

Tarskijev izrek o fiksni točki je ključni rezultat, ki omogoča uporabo denotacijske semantike za opis pomena rekurzivnih struktur.

Izrek 5.1.1 (Fiksna točka, Tarski) *Naj bo $f : D \to D$ zvezna funkcija nad domeno D.*

- *f ima* najmanjšo pre-fiksno točko*:*

$$fix(f) = \bigsqcup_{n \geq 0} f^n(\bot)$$

- *$fix(f)$ je fiksna točka f ker zadošča $f(fix(f)) = fix(f)$ in je zato tudi* najmanjša fiksna točka *funkcije f.* □

Opazka 5.1.4 *Notacija $f^n(\bot)$ uporabljena v definiciji:*

$$\begin{cases} f^0(\bot) & = \bot \\ f^{n+1}(\bot) & = f(f^n(\bot)) \end{cases} \tag{5.1}$$

Ker velja $\forall d \in D. \bot \sqsubseteq d$, potem velja tudi $f^0(\bot) = \bot \sqsubseteq f^1(\bot)$. Zaradi monotonosti

$$f^n(\bot) \sqsubseteq f^{n+1}(\bot) \Rightarrow f^{n+1}(\bot) = f(f^n(\bot)) \sqsubseteq f(f^{n+1}(\bot)) = f^{n+2}(\bot).$$

Z uporabo indukcije po $n \in \mathbb{N}$ potem velja $\forall n \in \mathbb{N}. f^n(\bot) \sqsubseteq f^{n+1}(\bot)$. Z drugimi besedami elementi $f^n(\bot)$ tvorijo verigo v D. Ker je D cpo je smiselno definirati fix(f). □

Dokaz. [Izrek o fiksni točki]

$$\begin{aligned} f(\text{fix}(f)) &= f(\bigsqcup_{n\geq 0} f^n(\bot)) \\ &= \bigsqcup_{n\geq 0} f(f^n(\bot)) && \text{(zaradi zveznosti)} \\ &= \bigsqcup_{n\geq 0} f^{n+1}(\bot) && () \\ &= \bigsqcup_{n\geq 0} f^n(\bot)) && \text{(zaradi Opazke 5.1.1)} \\ &= \text{fix}(f) \end{aligned}$$

fix(f) je torej fiksna točka za f in zadovolji prvi pogoj (lfp1) v Definiciji 5.1.11. Da bi preverili drugi pogoj $(lfp2)$ definicije najmanjše pre-fiksne točke, predpostavimo da za $d \in D$ velja $f(d) \sqsubseteq d$. Z indukcijo pokažemo, da $f^n(\bot) \sqsubseteq d$ velja za vse $n \in \mathbb{N}$. Ker je $\bot$ najmanjša vrednost v D velja:

$$f^0(\bot) = \bot \sqsubseteq d$$

in

$$\begin{aligned} f^n(\bot) \sqsubseteq d \Rightarrow f^{n+1}(\bot) = f(f^n(\bot)) &\sqsubseteq f(d) && \text{zaradi monotonosti } f \\ &\sqsubseteq d && \text{zaradi predpostavke o } d \end{aligned}$$

Po indukciji na $n \in \mathbb{N}$ dobimo $\forall n \in \mathbb{N}. f^n(\bot) \sqsubseteq d$. Element d je torej zgornja meja verige in leži nad najmanšo zgornjo mejo:

$$\text{fix}(f) = \bigsqcup_{n\geq 0} f^n(\bot) \sqsubseteq d$$

kot je zahtevano z (lfp2). □

5.1.3 Konstrukcije na domenah

V tej sekciji bodo predstavljeni razni načini za izgradnjo domen in zveznih funkcij. Za konstrukcijo cpo potrebujemo množico, ki je opremljena z binarno relacijo. Potem moramo dokazati, da veljajo naslednji pogoji.

1. Binarna relacija definira delno urejenost.
2. Za vse verige v delni urejenosti obstaja lub.

3. Za domeno je potrebno preveriti še ali ima dno.

V nadaljevanju bodo predstavljene različne metode za konstrukcijo cpo in domen.

Produkti domen

Primer 5.1.10 (Produkt dveh cpo) *Produkt dveh cpo* $(D_1, \sqsubseteq_1)$ *in* $(D_2, \sqsubseteq)$ *ima osnovno množico:*

$$D_1 \times D_2 = \{(d_1, d_2) | d_1 \in D_1 \wedge d_2 \in \mathcal{D}_2\}.$$

Delna urejenost $\sqsubseteq$ *je definirana na sledeč način:*

$$(d_1, d_2) \sqsubseteq (d_1', d_2') \Longleftrightarrow d_1 \sqsubseteq d_1' \wedge d_2 \sqsubseteq d_2'.$$

Najmanjše zgornje meje se izračunajo po komponentah.

$$\bigsqcup_{n \geq 0} (d_{1,n}, d_{2,n}) = (\bigsqcup_{n \geq 0} d_{1,n}, \bigsqcup_{n \geq 0} d_{2,n}) \tag{5.2}$$

Če sta $(D_1, \sqsubseteq_1)$ *in* $(D_2, \sqsubseteq_2)$ *domeni potem je domena tudi* $(D_1 \times D_2, \sqsubseteq)$ *in dno* $\perp_{D_1 \times D_2} = (\perp_{D_1}, \perp_{D_2})$. □

Opazka 5.1.5 *Domena* $(D_1 \times D_2, \sqsubseteq)$ *zahteva uporabo funkcij s katerimi izluščimo eno izmed komponent objekta* (d_1, d_2)*.*

$$\begin{array}{ll} \pi_1 : D_1 \times D_2 \to D_1 & \pi_1(d_1, d_2) =^{def} d_1 \\ \pi_2 : D_1 \times D_2 \to D_2 & \pi_2(d_1, d_2) =^{def} d_2 \end{array}$$

Funkciji π_1 *in* π_2 *sta zvezni, ker sta* D_1 *in* D_2 *domeni.* □

Opazka 5.1.6 *Naj bosta* $f_1 : D \to D_1$ *in* $f_2 : D \to D_2$ *zvezni funkciji. Potem je zvezna funkcija tudi:*

$$\begin{array}{l} \langle f_1, f_2 \rangle : D \to D_1 \times D_2 \\ \langle f_1, f_2 \rangle(d) =^{def} (f_1(d), f_2(d)) \end{array}$$

Dokaz. Zveznost $\langle f_1, f_2 \rangle$ sledi direktno iz definicije najmanjše zgornje meje (lub) verig parov z enačbo 5.2.

$$\begin{array}{rcl} \langle f_1, f_2 \rangle(\bigsqcup_{n \geq 0} d_n) & = & (f_1(\bigsqcup_{n \geq 0} d_n), f_2(\bigsqcup_{n \geq 0} d_n)) \\ & = & (\bigsqcup_{n \geq 0} f_1(d_n), \bigsqcup_{n \geq 0} f_2(d_n)) \\ & = & \bigsqcup_{n \geq 0} \langle f_1, f_2 \rangle(d_n) \end{array}$$

□

Primer 5.1.11 (Odvisni produkti) *Naj bo I množica in $\forall i \in I$ imamo cpo $(D_i, \sqsubseteq)$.* Produkt *cele družine cpo-jev je definiran na sledeč način.*

Osnovna množica I-kratnega kartezijskega produkta je $\Pi_{i \in I} D_i$: sestavljena je iz funkcij p definiranih nad I tako, da je vrednost za vsak $i \in I$ element $p(i) \in D_i$.

Relacija delne urejenosti $\sqsubseteq$ je definirana takole:

$$p \sqsubseteq p' \iff \forall i \in I . p(i) \sqsubseteq_i p'(i)$$

Najmanjše zgornje meje $(\Pi_{i \in I} D_i, \sqsubseteq)$ izračunamo po komponentah. Lub verige $p_0 \sqsubseteq p_1 \sqsubseteq p_2 \sqsubseteq \ldots$ iz produkta cpo-jev je funkcija, ki preslika vsak $i \in I$ v lub verige $p_0(i) \sqsubseteq p_1(i) \sqsubseteq p_2(i) \sqsubseteq \ldots$ iz D_i. Za verige v D_i pa velja:

$$(\bigsqcup_{n \geq 0} p_n)(i) = \bigsqcup_{n \geq 0} p_n(i) \qquad (i \in I)$$

Zdaj lahko definiramo za vsak $i \in I$ i-to projekcijsko funkcijo π_i.

$$\begin{array}{l} \pi_i : \Pi_{i' \in I} D_{i'} \to D_i \\ \pi_i(p) =^{def} p(i) \end{array}$$

Funkcije π_i so zvezne. V primeru, da so vsi D_i domene, potem je tudi $\Pi_{i \in I} D_i$ domena. Lub je funkcija, ki preslika $i \in I$ v najmanjšo zgornjo mejo D_i. □

Funkcijske domene

Iz množice zveznih funkcij med dvema cpo ali domenami lahko naredimo cpo oz. domeno.

Primer 5.1.12 *Naj bosta $(D, \sqsubseteq_D)$ in $(E, \sqsubseteq_E)$ cpo-ja. Funkcijski cpo $(D \to E, \sqsubseteq)$ ima osnovno množico:*

$$D \to E =^{def} \{f | f : D \to E \text{ je zvezna}\}.$$

Delna urejenost je definirana:

$$f \sqsubseteq f' \Leftrightarrow^{def} \forall d \in D . f(d) \sqsubseteq f'(d)$$

Najmanjše zgornje meje se izračunajo z uporabo najmanjših zgornjih mej iz E.

$$(\bigsqcup_{n \geq 0} f_n)(d) = \bigsqcup_{n \geq 0} f_n(d)$$

Če je E domena potem je domena tudi $D \to E$. Dno domene $D \to E$ je $\bot_{D \to E} = \bot_E$. □

Dokaz. Pokazali bi radi, da je lub verige funkcij $\bigsqcup_{n\geq 0} f_n$ zvezna funkcija. Uporabimo zakon zamenjave zveznih funkcij.

$$
\begin{array}{rcll}
(\bigsqcup_{n\geq 0} f_n)(\bigsqcup_{m\geq 0} d_m) & = & \bigsqcup_{n\geq 0}(f_n(\bigsqcup_{m\geq 0} d_m)) & \text{definicija } (\bigsqcup_{n\geq 0} f_n) \\
& = & \bigsqcup_{n\geq 0}(\bigsqcup_{m\geq 0} f_n(d_m)) & \text{zveznost } f_n \\
& = & \bigsqcup_{m\geq 0}(\bigsqcup_{n\geq 0} f_n(d_m)) & \text{zakon zamenjave} \\
& = & \bigsqcup_{m\geq 0}((\bigsqcup_{n\geq 0} f_n)(d_m)) & \text{definicija } (\bigsqcup_{n\geq 0} f_n)
\end{array}
$$

□

Opazka 5.1.7 (Evaluacija in Curry operacija) *Dana sta cpo-ja D in E in zvezna funkcija ev:*

$$
\begin{array}{l}
ev : (D \to E) \times D \to E \\
ev(f, d) =^{def} f(d)
\end{array}
$$

Naj bo $f : D' \times D \to E$ zvezna funkcija, kjer je D' cpo. Za vsak $d' \in D'$ zvezna funkcija $d \in D \to f(d', d)$ določa element funkcijskega cpo $D \to E$, ki ga bomo zapisali cur$(f)(d)$. Funkcija cur(f) je zvezna.

$$
\begin{array}{l}
\textit{cur} : D' \to (D \to E) \\
\textit{cur}(f)(d') =^{def} \lambda d \in D . f(d', d)
\end{array}
$$

□

Dokaz. Da bi pokazali zveznost ev spet uporabimo zakon zamenjave zveznih funkcij.

$$
\begin{array}{rcll}
ev(\bigsqcup_{n\geq 0}(f, d_n)) & = & ev(\bigsqcup_{i\geq 0} f_i, \bigsqcup_{j\geq 0} d_j) & \text{lub para po komponentah} \\
& = & (\bigsqcup_{i\geq 0} f_i)(\bigsqcup_{j\geq 0} d_j) & \text{po definiciji } ev \\
& = & \bigsqcup_{i\geq 0} f_i(\bigsqcup_{j\geq 0} d_j) & \text{lub-i v fun. cpo-jih po argumentih} \\
& = & \bigsqcup_{i\geq 0}\bigsqcup_{j\geq 0} f_i(d_i) & \text{zaradi zveznosti } f_i \\
& = & \bigsqcup_{n\geq 0} f_n(d_n) & \text{zaradi Opazke 8.1.1} \\
& = & \bigsqcup_{n\geq 0} ev(f_n, d_n) & \text{po definiciji } ev
\end{array}
$$

Zveznost vsake funkcije cur$(f)(d')$ in cur(f) sledi iz dejstva, da se lub-i verig iz $D_1 \times D_2$ računajo po komponentah. □

5.2 Denotacijska semantika IMP

Matematične osnove denotacijske semantike sta definirala Christopher Strachey in Dana Scott.

Vsakemu izrazu t jezika priredimo *denotacijo* $[\![t]\!]$, matematični objekt, ki predstavlja pomen izraza t.

Denotacija stavka je določena z denotacijami izrazov, ki sestavljajo stavek.

Semantične funkcije $\mathcal{A}$, $\mathcal{B}$ in $\mathcal{C}$ so definirane na osnovi strukturne indukcije.

$$\begin{array}{l} \mathcal{A} : AExp \to (\Sigma \to \mathbf{N}) \\ \mathcal{B} : BExp \to (\Sigma \to \mathbf{T}) \\ \mathcal{C} : Com \to (\Sigma \to \Sigma) \end{array}$$

Za vsak tip stavka definiramo parcialno funkcijo $\mathcal{A}, \mathcal{B}, \ldots$:

- predpostavljamo, da je denotacija komponent izrazov že definirana in
- pomen stavka s predstavlja $\mathcal{S}[\![s]\!]$, kjer je $\mathcal{S}$ denotacijska funkcija stavkov tipa s.

5.2.1 Aritmetični izrazi

Denotacijo aritmetičnih izrazov definiramo na osnovi strukturne indukcije kot razmerje med stanji in števili.

Aritmetični izrazi $AExp$.

$$\begin{array}{l} \mathcal{A}[\![n]\!] = \{(\sigma, n) \mid \sigma \in \Sigma\} \\ \mathcal{A}[\![X]\!] = \{(\sigma, \sigma(X)) \mid \sigma \in \Sigma\} \\ \mathcal{A}[\![a_0 + a_1]\!] = \{(\sigma, n_0 + n_1) \mid (\sigma, n_0) \in \mathcal{A}[\![a_0]\!] \wedge (\sigma, n_1) \in \mathcal{A}[\![a_1]\!]\} \\ \mathcal{A}[\![a_0 - a_1]\!] = \{(\sigma, n_0 - n_1) \mid (\sigma, n_0) \in \mathcal{A}[\![a_0]\!] \wedge (\sigma, n_1) \in \mathcal{A}[\![a_1]\!]\} \\ \mathcal{A}[\![a_0 * a_1]\!] = \{(\sigma, n_0 * n_1) \mid (\sigma, n_0) \in \mathcal{A}[\![a_0]\!] \wedge (\sigma, n_1) \in \mathcal{A}[\![a_1]\!]\} \end{array}$$

Simboli operacij na levi stani definicije npr. “+”, “-” ali “*” predstavljajo sintaktične elemente, medtem ko simboli operacij na desni strani enakosti predstavljajo operacije, $+$, $-$ in $*$.

Primer 5.2.1 $\mathcal{A}[\![3 * 5]\!]\sigma = \mathcal{A}[\![3]\!]\sigma + \mathcal{A}[\![5]\!]\sigma = 3 + 5 = 8$ □

Denotacija $\mathcal{A}[\![n]\!]$ je *funkcija*.

Definicija denotacijskih pravil aritmetičnih izrazov v λ-notaciji.

$$\begin{array}{l} \mathcal{A}[\![n]\!] = \lambda\sigma \in \Sigma.n \\ \mathcal{A}[\![X]\!] = \lambda\sigma \in \Sigma.\sigma(X) \\ \mathcal{A}[\![a_0 + a_1]\!] = \lambda\sigma \in \Sigma.(A[\![a_0]\!]\sigma + \mathcal{A}[\![a_1]\!]\sigma) \\ \mathcal{A}[\![a_0 - a_1]\!] = \lambda\sigma \in \Sigma.(A[\![a_0]\!]\sigma - \mathcal{A}[\![a_1]\!]\sigma) \\ \mathcal{A}[\![a_0 * a_1]\!] = \lambda\sigma \in \Sigma.(A[\![a_0]\!]\sigma * \mathcal{A}[\![a_1]\!]\sigma) \end{array}$$

5.2.2 Logični izrazi

Uporabimo logične operacije $\wedge_T$, $\vee_T$ in $\neg_T$ nad vrednostmi iz domene $\mathbf{T}$.

$$\begin{array}{lcl}
\mathcal{B}[\![true]\!] & = & \{(\sigma, true) \mid \sigma \in \Sigma\} \\
\mathcal{B}[\![false]\!] & = & \{(\sigma, false) \mid \sigma \in \Sigma\} \\
\mathcal{B}[\![a_0 = a_1]\!] & = & \{(\sigma, true) \mid \sigma \in \Sigma \wedge \mathcal{A}[\![a_0]\!]\sigma = \mathcal{A}[\![a_1]\!]\sigma\} \cup \\
 & & \{(\sigma, false) \mid \sigma \in \Sigma \wedge \mathcal{A}[\![a_0]\!]\sigma \neq \mathcal{A}[\![a_1]\!]\sigma\} \\
\mathcal{B}[\![a_0 \leq a_1]\!] & = & \{(\sigma, true) \mid \sigma \in \Sigma \wedge \mathcal{A}[\![a_0]\!]\sigma \leq \mathcal{A}[\![a_1]\!]\sigma\} \cup \\
 & & \{(\sigma, false) \mid \sigma \in \Sigma \wedge \mathcal{A}[\![a_0]\!]\sigma \not\leq \mathcal{A}[\![a_1]\!]\sigma\} \\
\mathcal{B}[\![\neg b]\!] & = & \{(\sigma, \neg_T t) \mid \sigma \in \Sigma \wedge (\sigma, t) \in \mathcal{B}[\![b]\!]\} \\
\mathcal{B}[\![b_0 \wedge b_1]\!] & = & \{(\sigma, t_0 \wedge_T t_1) \mid \sigma \in \Sigma \wedge (\sigma, t_0) \in \mathcal{B}[\![b_0]\!] \wedge (\sigma, t_1) \in \mathcal{B}[\![b_1]\!]\} \\
\mathcal{B}[\![b_0 \vee b_1]\!] & = & \{(\sigma, t_0 \vee_T t_1) \mid \sigma \in \Sigma \wedge (\sigma, t_0) \in \mathcal{B}[\![b_0]\!] \wedge (\sigma, t_1) \in \mathcal{B}[\![b_1]\!]\}
\end{array}$$

Lahko pokažemo z uporabo strukturne indukcije, da je denotacija $\mathcal{B}[\![b]\!]$ funkcija. Na primer:

$$\mathcal{B}[\![a_0 \leq a_1]\!] = \begin{cases} true & \mathcal{A}[\![a_0]\!]\sigma \leq \mathcal{A}[\![a_1]\!]\sigma \\ false & \mathcal{A}[\![a_0]\!]\sigma \not\leq \mathcal{A}[\![a_1]\!]\sigma \end{cases}$$

za vsako stanje $\sigma \in \Sigma$.

5.2.3 Stavki IMP

Definicija denotacije stavkov $\mathcal{C}[\![c]\!]$ je bolj kompleksna od prejšnjih primerov definicij denotacije aritmetičnih in boolovih izrazov.

Poglejmo najprej definicijo denotacije kot relacijo med stanji. Kasneje bomo videli, da gre pravzaprav za parcialno funkcijo.

$$\begin{array}{lcl}
\mathcal{C}[\![\text{skip}]\!] & = & \{(\sigma, \sigma) \mid \sigma \in \Sigma\} \\
\mathcal{C}[\![X := a]\!] & = & \{(\sigma, \sigma(n/X)) \mid \sigma \in \Sigma \wedge n = \mathcal{A}[\![a]\!]\sigma\} \\
\mathcal{C}[\![c_0; c_1]\!] & = & \mathcal{C}[\![c_1]\!] \circ \mathcal{C}[\![c_0]\!] \\
\mathcal{C}[\![\text{if } b \text{ then } c_0 \text{ else } c_1]\!] & = & \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]\sigma = true \wedge (\sigma, \sigma') \in \mathcal{C}[\![c_0]\!]\} \cup \\
 & & \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]\sigma = false \wedge (\sigma, \sigma') \in \mathcal{C}[\![c_1]\!]\}
\end{array}$$

Denotacija stavka while je bolj kompleksna; predstavili jo bomo kot fiksno točko izvajanja while stavka.

Uporabimo ekvivalenco while stavka, ki smo si jo ogledali že pri predstavitvi operacijske semantike.

$$w \equiv \text{while } b \text{ do } c$$

$$w \sim \text{if } b \text{ then } c; w \text{ else skip}$$

Denotacija stavka while:

$$\begin{aligned}\mathcal{C}[\![w]\!] &= \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = true \wedge (\sigma,\sigma') \in \mathcal{C}[\![c;w]\!]\} \cup \\ &\quad \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]\sigma = false\} \\ &= \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = true \wedge (\sigma,\sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\} \cup \\ &\quad \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]\sigma = false\}\end{aligned}$$

Zapišimo zdaj φ namesto $\mathcal{C}[\![w]\!]$, β namesto $\mathcal{B}[\![b]\!]$ in γ namesto $\mathcal{C}[\![c]\!]$.

$$\begin{aligned}\varphi = \quad &\{(\sigma,\sigma') \mid \beta(\sigma) = true \wedge (\sigma,\sigma') \in \varphi \circ \gamma\} \cup \\ &\{(\sigma,\sigma) \mid \beta(\sigma) = false\}\end{aligned}$$

Imamo rekurzivno enačbo.

$$\begin{aligned}\Gamma(\varphi) = \quad &\{(\sigma,\sigma') \mid \exists \sigma''.\beta(\sigma) = true \wedge (\sigma,\sigma'') \in \gamma \wedge (\sigma'',\sigma') \in \varphi\} \cup \\ &\{(\sigma,\sigma) \mid \beta(\sigma) = false\}\end{aligned}$$

- (σ, σ'') - evaluacija c
- (σ'', σ') - evaluacija w

Želimo izračunati fiksno točko φ v izrazu Γ.

$$\varphi = \Gamma(\varphi)$$

Funkcija Γ je enaka $\widehat{R}$, kjer je R operator na množici določen z naslednjim pravilom.

$$\begin{aligned}R \quad = \quad &\{(\{(\sigma'',\sigma')\}/(\sigma,\sigma') \mid \beta(\sigma) = true \wedge (\sigma,\sigma'') \in \gamma\} \cup \\ &\{(\emptyset/(\sigma,\sigma)) \mid \beta(\sigma) = false\}\end{aligned}$$

V naslednji sekciji bomo pokazali, da ima $\widehat{R}$ fiksno točko.

$$\varphi = \text{fix}(\widehat{R})$$

Izračun fiksne točke $\widehat{R}$

Množica Q je zaprta glede na množico pravil R ali preprosto R-zaprta, čče velja:

$$\forall (X/y) \in R. X \subseteq Q \Rightarrow y \in Q$$

.

Z drugimi besedami je množica zaprta glede na dana pravila, če velja, da vedno ko so v množici premise nekega pravila je tam tudi posledica.

Najmanjša R-zaprta množica je množica Q za katero velja: Q je R-zaprta in za vse R-zaprte množice J velja $Q \subseteq J$.

Operator $\widehat{R}$ je definiran z naslednjim zapisom.

$$\widehat{R}(B) = \{y \mid \exists X \subseteq B.(X/y) \in R\}$$

Operator $\widehat{R}$ omogoča še en način za opis zaprtih množic.

Množica B je zaprta za operator $\widehat{R}$ natako takrat, ko $\widehat{R}(B) \subseteq B$.

Operator $\widehat{R}$ je monoton:

$$A \subseteq B \Rightarrow \widehat{R}(A) \subseteq \widehat{R}(B)$$

Če zaporedoma uporabimo operator $\widehat{R}$ na prazni množici, dobimo sekvenco množic.

$$\begin{aligned}
&A_0 = \widehat{R}^0(\emptyset) = \emptyset \\
&A_1 = \widehat{R}^1(\emptyset) = \widehat{R}(\emptyset) \\
&A_2 = \widehat{R}^2(\emptyset) = \widehat{R}(\widehat{R}(\emptyset)) \\
&\vdots \\
&A_n = \widehat{R}^n(\emptyset)
\end{aligned}$$

Množice so zaporedoma vsebovane druga v drugi.

$$A_0 \subseteq A_1 \subseteq A_2 \subseteq \ldots \subseteq A_n \subseteq \ldots$$

Množica $A = \bigcup_{n \in w} A_n$ je najmanjša R-zaprta množica za katero velja: (i) $\widehat{R}(A) = A$ in (ii) A je najmanjša R-zaprta množica.

Tarskijev izrek pravi, da v primeru, da imamo zvezno funkcijo f nad domenami, lahko izračunamo najmanjšo fiksno točko:

$$\text{fix}(\widehat{R}) =_{def} \bigcup_{n \in \omega} \widehat{R}^n(\emptyset)$$

Denotacija while

Denotacija stavka w je torej določena s fiksno točko φ.

$$\mathcal{C}[\![\text{ while } b \text{ do } c\ ]\!] = fix(\Gamma)$$

$$\Gamma(\varphi) = \{(\sigma, \sigma') \mid \beta(\sigma) = true \wedge (\sigma, \sigma') \in \varphi \circ \mathcal{C}[\![c]\!]\} \cup \{(\sigma, \sigma) \mid \beta(\sigma) = false\}$$

Izrek 5.2.1 *Na bo $w \equiv$ while b do c, potem velja:*

$$\mathcal{C}[\![w]\!] = \mathcal{C}[\![\textit{ if } b \textit{ then } c; w \textit{ else skip }]\!]$$

Dokaz. Denotacija w je fiksna točka funkcije Γ.

$$\begin{aligned} \mathcal{C}[\![w]\!] = & \ \Gamma(\mathcal{C}[\![w]\!]) \\ = & \ \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = true \wedge (\sigma,\sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\} \cup \\ & \ \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]\sigma = false\} \\ = & \ \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = true \wedge (\sigma,\sigma') \in \mathcal{C}[\![c;w]\!]\} \cup \\ & \ \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]\sigma = false\} \\ = & \ \mathcal{C}[\![\text{if } b \text{ then } c; w \text{ else skip}]\!] \end{aligned}$$ □

5.2.4 Zanka while kot fiksna točka

Poglejmo si pomen v konkretnem primeru while stavka:

$$\text{while } X > 0 \text{ do } (Y := Y * X;\ X := X - 1)$$

Spremenljivki X in Y predstavljata lokaciji v spominu.

Stanje abstraktne predstavitve stavka while predstavimo s parom celih števil (x, y), ki predstavljata vrednost spremenljivk X in Y.

$$\begin{aligned} & State =_{def} \mathbb{Z} \times \mathbb{Z} \\ & D =_{def} State \rightharpoonup State \end{aligned}$$

Denotacijo stavka while $X > 0$ do $(Y := Y * X;\ X := X - 1)$ definiramo kot funkcijo $w : (\mathbb{Z} \times \mathbb{Z}) \to (\mathbb{Z} \times \mathbb{Z})$.

Denotacija stavka $[\![\text{while } X > 0 \text{ do } (Y := Y * X;\ X := X - 1)]\!] \in D$ je rešitev enačbe fiksne točke $w = f(w)$, kjer $f : D \to D$ označuje eno iteracijo stavka while.

$$f(w)(x,y) = \begin{cases} (x,y) & \text{če } x \leq 0 \\ w(x-1, x*y) & \text{če } x > 0 \end{cases}$$

Poglejmo si zdaj delno urejenost $\sqsubseteq$ nad D:

$$\begin{aligned} w \sqsubseteq w' \quad & \text{čče velja } \forall (x,y) \in State: \text{ če je } w \text{ definiran za } (x,y), \\ & \text{potem je definiran tudi } w' \text{ in velja tudi } w(x,y) = w'(x,y) \end{aligned}$$

Imamo tudi najmanjši element $\perp \in D$:

$$\begin{aligned} & \perp =_{def} \text{totalno nedefinirana parcialna funkcija} \\ & \forall w \in D : \perp \sqsubseteq w \end{aligned}$$

Delna urejenost $\sqsubseteq$ definira neke vrste 'informacijsko urejenost': $w \sqsubseteq w'$ če se w' ujema z w vedno, ko je w definirana, vendar je w' lahko definirana širše.

Sekvenco $w_0, w_1, w_2, \ldots$ dobimo tako, da začnemo z $\perp$ in apliciramo znova in znova funkcijo f nad prejšnjo vrednostjo:

$$\begin{cases} w_0 =_{def} \perp \\ w_{n+1} =_{def} f(w_n) \end{cases}$$

Če zdaj uporabimo definicijo f dobimo sledečo sekvenco funkcij:

$$w_1(x,y) = f(\perp)(x,y) = \begin{cases} (x,y) & \text{če } x \leq 0 \\ \text{nedefinirano} & \text{če } x \geq 1 \end{cases}$$

$$w_2(x,y) = f(w_1)(x,y) = \begin{cases} (x,y) & \text{če } x \leq 0 \\ (0,y) & \text{če } x = 1 \\ \text{nedefinirano} & \text{če } x \geq 2 \end{cases}$$

$$w_3(x,y) = f(w_2)(x,y) = \begin{cases} (x,y) & \text{če } x \leq 0 \\ (0,y) & \text{če } x = 1 \\ (0, 2 * y) & \text{če } x = 2 \\ \text{nedefinirano} & \text{če } x \geq 3 \end{cases}$$

$$w_4(x,y) = f(w_3)(x,y) = \begin{cases} (x,y) & \text{če } x \leq 0 \\ (0,y) & \text{če } x = 1 \\ (0, 2 * y) & \text{če } x = 2 \\ (0, 6 * y) & \text{če } x = 3 \\ \text{nedefinirano} & \text{če } x \geq 4 \end{cases}$$

V splošnem dobimo sledečo funkcijo:

$$w_n(x,y) = f(w_{n-1})(x,y) = \begin{cases} (x,y) & \text{če } x \leq 0 \\ (0, (!x) * y) & \text{če } 0 < x < n \\ \text{nedefinirano} & \text{če } x \geq n \end{cases}$$

Tako smo dobili naraščajočo sekvenco parcialnih funkcij:

$$w_0 \sqsubseteq w_1 \sqsubseteq w_2 \sqsubseteq \ldots \sqsubseteq w_n \sqsubseteq \ldots$$

Unija parcialnih funkcij v sekvenci je element w_∞

$$w_\infty(x,y) = \begin{cases} (x,y) & \text{če } x \leq 0 \\ (0, (!x) * y) & \text{če } x > 0 \end{cases}$$

Funkcija w_∞ je fiksna točka funkcije f zato ker velja:

$$\begin{aligned} f(w_\infty)(x,y) &= \begin{cases} (x,y) & \text{če } x \leq 0 \\ w_\infty(x-1, x * y) & \text{če } x > 0 \end{cases} && \text{(po definiciji } f\text{)} \\ &= \begin{cases} (x,y) & \text{če } x \leq 0 \\ (0, 1 * y) & \text{če } x = 1 \\ (0, !(x-1) * x * y) & \text{če } x > 1 \end{cases} && \text{(po definiciji } w_\infty\text{)} \\ &= w_\infty(x,y). \end{aligned}$$

Velja tudi, da je w_∞ najmanjša fiksna točka f:

$$\forall w \in D : w = f(w) \Rightarrow w_\infty \sqsubseteq w.$$

Konstrukcija denotacije stavka while $X > 0$ do $(Y := Y * X;\ X := X - 1)$ je primer uporabe Tarskijevega izreka.

5.3 Opombe

Poglavje temelji na prosojnicah Glyn Winskel z naslovom *Denotational semantics* [18] ter na knjigi Glyn Winskel za naslovom *The Formal Semantics of Programming Languages* [16].