Poglavje 4

OPERACIJSKA SEMANTIKA

Operacijska semantika definira obnašanje programskega jezika z definicijo enostavnega *abstraktnega stroja*.

Stroj je abstrakten zato, ker uporabljamo izraze programskega jezika kot strojni jezik. Jezik ne prevajamo v bolj konkreten jezik kot to delajo prevajalniki.

V primeru enostavnih jezikov pomeni stanje stroja preprosto stavek. Prehodi med stanji stroja so prehodi iz enega izraza v poenostavitev izraza ali v primeru, da se evaluacija zaključi, prehod v normalni izraz kjer se abstraktni stroj ustavi.

Računalniške jezike bomo predstavljali iz večih izbranih vidikov. Najpomembnejša aspekta jezikov sta statična struktura in evaluacija, ki ju predstavimo s *statično* in *dinamično* semantiko.

Katerikoli aspek programskega jezika bomo predstavili s pomočjo *pravil*, ki ustrezajo logičnim sodbam ali izjavam. Množica pravil predstavlja *teorijo* s katero opišemo željen aspekt jezika.

Lastnosti jezika lahko zdaj opazujemo preko lastnosti teorije. Pomen izrazov jezika izrazimo s pomočjo izbrane teorije. Interpretacija izraza je sekvenca aplikacij pravil teorije na danem izrazu.

Semantiko jezika torej opišemo z apliciranjem pravil teorije na izrazih jezika. Konkreten pomen izraza je *veriga aplikacij pravil* s katero opišemo bodisi izpeljavo statične strukture, izpeljavo ovrednotenja izrazov, preverjanje tipov izrazov, itd.

Izpeljava je torej osnovni način za podrobnejši opis pomena izraza–iz izpeljave lahko vidimo podrobnosti pri strukturi, interpretaciji, evaluaciji, itd.

4.1 Osnovne lastnosti programskega jezika

Opomba. [1]

Pri predstavitvi semantike bomo uporabili primer enostavnega jezika $\mathcal{L}(\text{nat bool})$ s katerim lahko opišemo boolove in numerične vrednosti ter pogojne izraze za boolove in numerične vrednosti.

Najprej bomo predstavili analizo statične strukture jezika z uporabo *statične semantike*. Sintakso jezika bomo predstavili s pravili, ki definirajo zgradbo jezika. Ogledali si bomo tudi alternativne načine opisov pravilnih stavkov danega jezika.

Nadaljevali bomo z *dinamično semantiko*: opisom mehanizmov za predstavitev evaluacije izrazov danega jezika. Evaluacijo bomo predstavili z množico pravil, katerih izvajanje opisuje ovrednotenje izrazov.

Po tem si bomo ogledali kako lahko predstavimo sklepanje z izpeljevanjem pravil. Predstavili bomo osnovne mehanizme uporabljene pri izpeljevanju pravil.

Sledila bo predstavitev dveh osnovnih lastnosti jezikov ter dokazovanje le-teh z indukcijo. Predstavljene lastnosti jezikov bodo: *determinističnost*, *zaključitev izvajanja*, in *uvrstitev v hierarhijo teorije izračunljivosti*. Naštete lastnosti imajo zelo različna imena v literaturi.

[1] Katere lastnosti jezika modeliramo in preučujemo z operacijsko semantiko? Predstavitev posameznih lastnosti PJ, ki so zanimive za obravnavo. □

Predstavljene bomo tudi *varne* teorije. Pod tem izrazom razumemo teorije, katerih pravila ne povzročijo mrtvega stanja, kar imenujemo tudi *napredek* in ohranjajo lastnosti jezika oz. tipe izrazov pri izpeljavi, kar imenujemo tudi *ohranitev*.

4.2 Statična semantika IMP

IMP je enostaven imperativen jezik z while zanko za katerega bomo definirali operacijsko in v naslednjem poglavju tudi denotacijsko semantiko.

Domene:

naravna števila	$\mathbb{N}$
boolove vrednosti	$t = \{true, false\}$
lokacije	Loc
aritmetični izrazi	$AExp$
boolovi izrazi	$BExp$
ukazi	Com

Meta spremenljivke:

$n, m \in N$
$X, Y \in Loc$
$a \in AExp$
$b \in BExp$
$c \in Com$

Imenu je implicitno pripisan tip.

Izrazi IMP:

$$\begin{array}{lcl} AExp\, a & ::= & n \mid X \mid \\ & & a_0 + a_1 \mid a_0 - a_1 \mid a_0 \times a_1 \\ BExp\, b & ::= & \text{true} \mid \text{false} \mid \\ & & a_0 = a_1 \mid a_0 \leq a_1 \mid \\ & & \neg b \mid b_0 \wedge b_1 \mid b_0 \vee b_1 \\ Com\, c & ::= & \text{skip} \mid X := a \mid c_0; c_1 \mid \\ & & \text{if } b \text{ then } c_0 \text{ else } c_1 \mid \\ & & \text{while } b \text{ do } c \end{array} \tag{4.1}$$

IMP je zelo enostaven imperativen jezik, ki vsebovan v vseh programskih jezikih.

Premisli:

- Kaj je pomen IMP izrazov?
- Kako sta povezana evaluacija in pomen izrazov?

Definirajmo zdaj statično semantiko izrazov IMP s katero bomo opisali kako so stavki jezika IMP zgrajeni.

Poglejmo najprej konstruiranje aritmetičnih izrazov. BNF oblika zapisa sintakse aritmetičnih izrazov je naslednja.

$$a ::= n \mid X \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 \times a_1$$

Statična semantika aritmetičnih izrazov napisana z uporabo indukcijskih pravil je sledeča.

$$\frac{n \text{ nat}}{n \text{ aexp}} \qquad \frac{X \text{ loc}}{X \text{ aexp}} \tag{4.2}$$

$$\frac{a_0 \text{ aexp} \quad a_1 \text{ aexp}}{a_0 + a_1 \text{ aexp}} \qquad \frac{a_0 \text{ aexp} \quad a_1 \text{ aexp}}{a_0 - a_1 \text{ aexp}} \qquad \frac{a_0 \text{ aexp} \quad a_1 \text{ aexp}}{a_0 * a_1 \text{ aexp}} \tag{4.3}$$

BNF oblika logičnih izrazov je definirana z naslednjim izrazom.

$$b ::= \text{true} \mid \text{false} \mid a_0 = a_1 \mid a_0 \leq a_1 \mid \neg b \mid b_0 \wedge b_1 \mid b_0 \vee b_1$$

Statična semantika logičnih izrazov je definirana z naslednjimi pravili.

$$\frac{}{\text{true bool}} \qquad \frac{}{\text{true bexp}} \tag{4.4}$$

$$\frac{a_0 \text{ aexp} \quad a_1 \text{ aexp}}{a_0 = a_1 \text{ bexp}} \qquad \frac{a_0 \text{ aexp} \quad a_1 \text{ aexp}}{a_0 \leq a_1 \text{ bexp}} \tag{4.5}$$

$$\frac{b \text{ bexp}}{\neg b \text{ bexp}} \qquad \frac{b_0 \text{ bexp} \quad b_1 \text{ bexp}}{b_0 \wedge b_1 \text{ bexp}} \qquad \frac{b_0 \text{ bexp} \quad b_1 \text{ bexp}}{b_0 \vee b_1 \text{ bexp}} \tag{4.6}$$

Končno, BNF sintaksa ukazov IMP je sledeča.

$$c ::= \text{skip} \mid X := a \mid c_0; c_1 \mid \text{if } b \text{ then } c_0 \text{ else } c_1 \mid \text{while } b \text{ do } c$$

Statična semantika ukazov je definirana z naslednjimi pravili.

$$\frac{}{\text{skip com}} \tag{4.7}$$

$$\frac{X \text{ loc} \quad a \text{ aexp}}{X := a \text{ com}} \tag{4.8}$$

$$\frac{c_0 \text{ com} \quad c_1 \text{ com}}{c_0; c_1 \text{ com}} \tag{4.9}$$

$$\frac{b \text{ bexp} \quad c_0 \text{ com} \quad c_1 \text{ com}}{\text{if } b \text{ then } c_0 \text{ else } c_1 \text{ com}} \tag{4.10}$$

$$\frac{b \text{ bexp} \quad c \text{ com}}{\text{while } b \text{ do } c \text{ com}} \tag{4.11}$$

4.3 Dinamična semantika IMP

4.3.1 Evaluacija aritmetičnih izrazov

Evaluacija števil, spremenljivk in aritmetičnih operacij.

$$\frac{}{\langle n, \sigma\rangle \to n} \tag{4.12}$$

$$\frac{}{\langle X, \sigma\rangle \to \sigma(X)} \tag{4.13}$$

$$\frac{\langle a_0, \sigma\rangle \to n_0 \quad \langle a_1, \sigma\rangle \to n_1 \quad n = n_0 + n_1}{\langle a_0 + a_1, \sigma\rangle \to n} \tag{4.14}$$

$$\frac{\langle a_0, \sigma\rangle \to n_0 \quad \langle a_1, \sigma\rangle \to n_1 \quad n = n_0 - n_1}{\langle a_0 - a_1, \sigma\rangle \to n} \tag{4.15}$$

$$\frac{\langle a_0, \sigma\rangle \to n_0 \quad \langle a_1, \sigma\rangle \to n_1 \quad n = n_0 * n_1}{\langle a_0 \times a_1, \sigma\rangle \to n} \tag{4.16}$$

Kako beremo pravila?

Če velja $\langle a_0, \sigma\rangle \to n_0$ in $\langle a_1, \sigma\rangle \to n_1$, potem $\langle a_0 + a_1, \sigma\rangle \to n$, kjer je n vsota n_0 in n_1.

Mehanizmi po katerih se vrednosti dejansko izračunajo (npr. $n = n_0 + n_1$) niso opisani.

Meta-spremenljivke $n, X, a_0, a_1, ...$ imajo zalogo vrednosti množice **N**, Loc, $AExp$, ..., kar je razvidno iz imena meta-spremenljivk..

Instanco pravila dobimo z instanciranjem spremenljivk na konkretne vrednosti.

Primer: $a \equiv 2 * 3$

$$\frac{\langle 2, \sigma_0\rangle \to 2 \quad \langle 3, \sigma\rangle \to 3}{\langle 2 \times 3, \sigma\rangle \to 6} \tag{4.17}$$

Primer: $a \equiv (Init + 5) + (5 + 9)$

$$\dfrac{\dfrac{\overline{\langle Init, \sigma\rangle \to 0} \quad \overline{\langle 5, \sigma\rangle \to 5}}{\langle Init + 5, \sigma_0\rangle \to 5} \quad \dfrac{\overline{\langle 5, \sigma\rangle \to 5} \quad \overline{\langle 9, \sigma\rangle \to 9}}{\langle 5 + 9, \sigma_0\rangle \to 14}}{\langle (Init + 5) + (5 + 9), \sigma\rangle \to 19} \tag{4.18}$$

4.3.2 Evaluacija logičnih izrazov

Vrednosti

$$\overline{\langle true, \sigma\rangle \to true} \tag{4.19}$$

$$\overline{\langle false, \sigma\rangle \to false} \tag{4.20}$$

Enakost $=$

$$\dfrac{\langle a_0, \sigma\rangle \to n_0 \quad \langle a_1, \sigma\rangle \to n_1 \quad n_0 = n_1}{\langle a_0 = a_1, \sigma\rangle \to true} \tag{4.21}$$

$$\dfrac{\langle a_0, \sigma\rangle \to n_0 \quad \langle a_1, \sigma\rangle \to n_1 \quad n_0 \neq n_1}{\langle a_0 = a_1, \sigma\rangle \to false} \tag{4.22}$$

Manjše ali enako $\leq$

$$\dfrac{\langle a_0, \sigma\rangle \to n_0 \quad \langle a_1, \sigma\rangle \to n_1 \quad n_0 \leq n_1}{\langle a_0 \leq a_1, \sigma\rangle \to true} \tag{4.23}$$

$$\dfrac{\langle a_0, \sigma\rangle \to n_0 \quad \langle a_1, \sigma\rangle \to n_1 \quad n_0 \not\leq n_1}{\langle a_0 \leq a_1, \sigma\rangle \to false} \tag{4.24}$$

Negacija

$$\dfrac{\langle b, \sigma\rangle \to true}{\langle \neg b, \sigma\rangle \to false} \qquad \dfrac{\langle b, \sigma\rangle \to false}{\langle \neg b, \sigma\rangle \to true}$$

Konjunkcija

$$\frac{\langle b_0, \sigma\rangle \to t_0 \quad \langle b_1, \sigma\rangle \to t_1 \quad t = t_0 \& t_1}{\langle b_0 \wedge b_1, \sigma\rangle \to t} \tag{4.25}$$

Disjunkcija

$$\frac{\langle b_0, \sigma\rangle \to t_0 \quad \langle b_1, \sigma\rangle \to t_1 \quad t = t_0 \| t_1}{\langle b_0 \vee b_1, \sigma\rangle \to t} \tag{4.26}$$

Ekvivalenca logičnih izrazov

$$b_0 \equiv b_1 \text{ iff } \forall t \forall \sigma \in \Sigma. \langle b_0, \sigma\rangle \to t \Longleftrightarrow \langle b_1, \sigma\rangle \to t \tag{4.27}$$

Bolj učinkovito evaluacijo konjunkcije $b_0 \wedge b_1$ dobimo, če najprej preverimo prvi parameter $b_0 =$ false in šele potem ovrednotimo drugi parameter b_1. V primeru, da je $b_0 =$ false se tako izognemo evaluaciji b_1.

$$\frac{\langle b_0, \sigma\rangle \to false}{\langle b_0 \wedge b_1, \sigma\rangle \to false}$$

$$\frac{\langle b_0, \sigma\rangle \to true \quad \langle b_1, \sigma\rangle \to false}{\langle b_0 \wedge b_1, \sigma\rangle \to false}$$

$$\frac{\langle b_0, \sigma\rangle \to true \quad \langle b_1, \sigma\rangle \to true}{\langle b_0 \wedge b_1, \sigma\rangle \to true}$$

4.3.3 Evaluacija ukazov

Začetno stanje ima lastnost $\forall X \in Loc : \sigma_0(X) = 0$.

Izvajanje se konča v *končnem stanju* ali divergira v neskončno.

Par $\langle c, \sigma\rangle$ predstavlja (ukaz) *konfigurcijo* v kateri izvedemo ukaz c v stanju σ.

$$\langle c, \sigma\rangle \to \sigma'$$

Primer:

$$\langle X := 5, \sigma\rangle \to \sigma'$$

Notacija: Naj bodo $\sigma \in \Sigma, m \in \mathbf{N}$ in $X \in \mathbf{Loc}$. $\sigma[m/X](Y)$ je stanje, ki ga dobimo iz σ, če zamenjamo vse pojavitve Xz m:

$$\sigma[m/X](Y) = \begin{cases} m & \text{if } Y = X \\ \sigma(Y) & \text{if } X \neq Y \end{cases}$$

Atomični ukazi

$$\frac{}{\langle \text{skip}, \sigma\rangle \to \sigma} \tag{4.28}$$

$$\frac{\langle a, \sigma\rangle \to m}{\langle X := a, \sigma\rangle \to \sigma[m/X]} \tag{4.29}$$

Kompozicija

$$\frac{\langle c_0, \sigma\rangle \to \sigma'' \quad \langle c_1, \sigma''\rangle \to \sigma'}{\langle c_0; c_1, \sigma\rangle \to \sigma'} \tag{4.30}$$

Vejitveni stavek

$$\frac{\langle b, \sigma\rangle \to true \quad \langle c_0, \sigma\rangle \to \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma\rangle \to \sigma'} \tag{4.31}$$

$$\frac{\langle b, \sigma\rangle \to false \quad \langle c_1, \sigma\rangle \to \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma\rangle \to \sigma'} \tag{4.32}$$

Stavek while

$$\frac{\langle b, \sigma\rangle \to false}{\langle \text{while } b \text{ do } c, \sigma\rangle \to \sigma} \tag{4.33}$$

$$\frac{\langle b, \sigma\rangle \to true \quad \langle c, \sigma\rangle \to \sigma'' \quad \langle \text{while } b \text{ do } c, \sigma''\rangle \to \sigma'}{\langle \text{while } b \text{ do } c, \sigma\rangle \to \sigma'} \tag{4.34}$$

4.4 Indukcija in izpeljave stavkov

4.4.1 Dobro-definirana indukcija

4.4.2 Indukcija na izpeljavah

Oglejmo si zdaj uporabo dobro-definirane indukcije na strukturi izpeljav stavkov.

Vse možne izpeljave so definirane s pravili, ki definirajo teorijo. Instance pravil imajo naslednjo obliko.

$$\frac{}{x} \quad ali \quad \frac{x_1, \ldots, x_n}{x}$$

Prvo pravilo je aksiom, ki nima premis in ima posledico x. Drugo pravilo ima množico premis $\{x_1, \ldots, x_n\}$ in posledico x. Pravila povejo kako konstruirati izpeljave: definirajo množico stavkov, ki so izpeljivi z uporabo pravil.

Izpeljava stavka x ima obliko *drevesa*. Drevo je določeno bodisi z aksomom $\frac{}{x}$ ali z drevesom izpeljave:

$$\frac{\frac{\vdots}{x_1}, \ldots, \frac{\vdots}{x_n}}{x}$$

Izpeljava stavka x vključuje tudi izpeljave stavkov $x_1, \ldots, x_n$, ki so premise pri izpeljavi x. Izpeljave $\frac{\vdots}{x_1}, \ldots, \frac{\vdots}{x_n}$ predstavljajo pod-izpeljave daljše izpeljave $\frac{\vdots}{x}$.

Instance pravil dobimo z zamenjavo *dejanskih* izrazov (vrednosti) na mestu meta-spremenljivk. Pravila, ki nas zanimajo so *končna*, ker so premise končne. Instanca

pravila ima torej končno množico premis in posledico.

Množica primerkov pravil R vsebuje elemente, ki so pari: (X/y), kjer je X končna množica in je y stavek. Par imenujemo *instanca pravila* s premiso X in posledico y.

Naj bo R množica instanc pravil. R-*izpeljava* stavka y je bodisi instanca pravila $(\emptyset/y)$ ali par $(\{d_1, \ldots, d_n\}/y)$ kjer je $(\{x_1, \ldots, x_n\}/y)$ instanca pravila in so d_1 izpeljava x_1, ..., d_n R-izpeljava x_n. Napišemo $d \Vdash_R y$ kar pomeni, da je d R-izpeljava y.

$$(\emptyset/y) \Vdash_R y) \Longleftarrow (\emptyset/y) \in R$$
$$(\{d_1, \ldots, d_n\}/y) \Vdash y \Longleftarrow (\{x_1, \ldots, x_n\}/y) \in R \,\&\, d_1 \Vdash_R x_1 \,\&\, \ldots \,\&\, d_n \Vdash_R x_n$$

Pravimo, da je y izpeljan iz R, če obstaja R-izpeljava y: $d \Vdash_R y$ za neko izpeljavo d. Napišemo $\Vdash_R y$, če je y izpeljan iz R.

4.4.3 Pomen stavka while

Intuitivna razlaga stavka $w \equiv$ while b do c je sledeča.

- Če je vrednost $b = true$ potem se ovrednoti stavek c, čemur sledi spet stavek w.
- Če je vrednost $b = false$ potem se izvajanje w ustavi in ovrednoti se stavek skip.
- Intuitivno razlago zapišemo v sledeči trditvi.

Izrek 4.4.1 *Naj bo $w \equiv$ while b do c, kjer je $c \in Bexp, c \in Comm$, potem velja naslednja enačba.*

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

Dokaz. Pokazali bi radi

$$\langle w, \sigma\rangle \to \sigma' \Leftrightarrow \langle \text{if } b \text{ then } c; w \text{ else skip}, \sigma\rangle \to \sigma',$$

za vsa stanja σ, σ'.

"$\Longrightarrow$":

Naj bo $\langle w, \sigma\rangle \to \sigma'$, za stanja σ, σ'. Potem mora obstajati izpeljava $\langle w, \sigma\rangle \to \sigma'$. Poglejmo si oblike izpeljave, ki lahko vodijo do w. Vidimo, da je končno pravilo v izpeljavi lahko le:

$$\frac{\langle b, \sigma\rangle \to false}{\langle w, \sigma\rangle \to \sigma} \quad (1 \Rightarrow)$$

ali

$$\frac{\langle b, \sigma\rangle \to true \quad \langle c, \sigma\rangle \to \sigma'' \quad \langle w, \sigma''\rangle \to \sigma'}{\langle w, \sigma\rangle \to \sigma'} \quad (2 \Rightarrow)$$

V primeru (1$\Rightarrow$) mora izpeljava $\langle w, \sigma\rangle \to \sigma'$ imeti obliko:

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \to false}}{\langle w, \sigma\rangle \to \sigma}$$

Z uporabo izpeljave $\langle b, \sigma\rangle \to false$ lahko konstruiramo naslednjo izpeljavo stavka $\langle \text{if } b \text{ then } c; w \text{ else skip}, \sigma\rangle \to \sigma$:

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \to false} \quad \dfrac{}{\langle \text{skip}, \sigma\rangle \to \sigma}}{\langle \text{if } b \text{ then } c; w \text{ else skip}, \sigma\rangle \to \sigma}$$

V primeru (2$\Rightarrow$) mora izpeljava $\langle w, \sigma\rangle \to \sigma$ imeti obliko:

$$\dfrac{\dfrac{\vdots}{\langle b,\sigma\rangle \to true} \quad \dfrac{\vdots}{\langle c,\sigma\rangle \to \sigma''} \quad \dfrac{\vdots}{\langle w,\sigma''\rangle \to \sigma'}}{\langle w,\sigma\rangle \to \sigma'},$$

ki vsebuje izraze $\langle b,\sigma\rangle \to true, \langle c,\sigma\rangle \to \sigma''$ in $\langle w,\sigma''\rangle \to \sigma'$. Iz teh izrazov lahko dobimo $\langle c; w,\sigma\rangle \to \sigma'$:

$$\dfrac{\dfrac{\vdots}{\langle c,\sigma\rangle \to \sigma''} \quad \dfrac{\vdots}{\langle w,\sigma''\rangle \to \sigma'}}{\langle c; w,\sigma\rangle \to \sigma'}$$

Damo še izpeljave skupaj:

$$\dfrac{\dfrac{\vdots}{\langle b,\sigma\rangle \to true} \quad \dfrac{\dfrac{\vdots}{\langle c,\sigma\rangle \to \sigma''} \quad \dfrac{\vdots}{\langle w,\sigma''\rangle \to \sigma'}}{\langle c; w,\sigma''\rangle \to \sigma'}}{\langle \text{if } b \text{ then } c; w \text{ else skip},\sigma\rangle \to \sigma'}$$

V obeh primerih $(1 \Rightarrow)$ in $(2 \Rightarrow)$ dobimo:

$$\langle w,\sigma\rangle \to \sigma' \Longrightarrow \langle \text{if } b \text{ then } c; w \text{ else skip},\sigma\rangle \to \sigma'.$$

"$\Longleftarrow$":

Če velja $\langle \text{if } b \text{ then } c; w \text{ else skip},\sigma\rangle \to \sigma'$, potem obstaja tudi izpeljava za izraz $\langle w,\sigma\rangle \to \sigma'$ za vsa stanja σ, σ'. Poglejmo si oblike izpeljave, ki lahko vodijo do izraza $\langle \text{if } b \text{ then } c; w \text{ else skip},\sigma\rangle$:

$$\dfrac{\dfrac{\vdots}{\langle b,\sigma\rangle \to false} \quad \overline{\langle skip,\sigma\rangle \to \sigma}}{\langle \text{if } b \text{ then } c; w \text{ else skip},\sigma\rangle \to \sigma'} \quad (1 \Leftarrow)$$

ali

$$\frac{\begin{array}{cc}\vdots & \vdots \\ \overline{\langle b, \sigma\rangle \to true} & \overline{\langle c; w, \sigma\rangle \to \sigma'}\end{array}}{\langle \text{if } b \text{ then } c; w \text{ else skip}, \sigma\rangle \to \sigma'} \quad (2 \Leftarrow)$$

V prvem primeru $(1 \Leftarrow)$ imamo edino izpeljavo, ki vodi do skip.

$$\overline{\langle \text{skip}, \sigma\rangle \to \sigma}$$

Zdaj lahko enostavno konstruiramo izpeljavo $\langle w, \sigma\rangle \to \sigma'$. Poglejmo še izpeljavo iz primera $(2 \Leftarrow)$, ki je težja. Izpeljava $(2 \Leftarrow)$ vsebuje izpeljavo $\langle c; w, \sigma\rangle \to \sigma'$, ki mora imeti obliko:

$$\frac{\begin{array}{cc}\vdots & \vdots \\ \overline{\langle c, \sigma\rangle \to \sigma''} & \overline{\langle w, \sigma''\rangle \to \sigma'}\end{array}}{\langle c; w, \sigma\rangle \to \sigma'}$$

za neko stanje σ''. Z uporabo izpeljav $\langle c, \sigma\rangle \to \sigma''$, $\langle w, \sigma''\rangle \to \sigma'$ in $\langle b, \sigma\rangle \to true$ lahko konstruiramo izpeljavo $\langle w, \sigma\rangle \to \sigma'$ na sledeč način:

$$\frac{\begin{array}{ccc}\vdots & \vdots & \vdots \\ \overline{\langle b, \sigma\rangle \to true} & \overline{\langle c, \sigma\rangle \to \sigma''} & \overline{\langle w, \sigma''\rangle \to \sigma'}\end{array}}{\langle w, \sigma\rangle \to \sigma'}$$

Izpeljavo za $\langle w, \sigma\rangle \to \sigma'$ iz izpeljave $(1 \Leftarrow)$ konstruiramo podobno. Zaključimo lahko, da v obeh primerih $(1 \Leftarrow)$ in $(2 \Leftarrow)$ dobimo:

$$\langle \text{if } b \text{ then } c; w \text{ else skip}, \sigma\rangle \to \sigma' \Longrightarrow \langle w, \sigma\rangle \to \sigma'.$$

Velja torej:

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

□

4.5 Determinističnost IMP

Naj bosta d in d' izpeljavi. Izpeljava d' je *takojšnja izpeljava* d, kar zapišemo $d' \prec_1 d$, čče ima d obliko (D/y) in velja $d' \in D$.

Relacija $\prec$ naj predstavlja tranzitivno zaprtje $\prec_1$: $\prec = \prec_1^+$. Pravimo, da je d' podizpeljava d, čče $d' \prec d$.

Izrek 4.5.1 *Naj bo c ukaz in σ_0 stanje. Če velja $\langle c, \sigma_0\rangle \to \sigma_1$ in $\langle c, \sigma_0\rangle \to \sigma$ potem $\sigma = \sigma_1$ za vsa stanje σ in σ_1.*

Dokaz. Dokaz uporablja dobro-definirano indukcijo na relaciji izpeljave $\prec$. Lastnost, ki naj velja za vse izpeljave d je sledeča.

$$P(d) \Leftrightarrow \forall c \in Com, \sigma_0, \sigma, \sigma_1 \in \Sigma : d \Vdash \langle c, \sigma_0\rangle \to \sigma_1 \;\&\; \langle c, \sigma_0\rangle \to \sigma \Rightarrow \sigma_1 = \sigma$$

Z uporabo indukcije na izpeljavah je zadosti, da pokažemo $\forall d' \prec d : P(d') \Rightarrow P(d)$. Pa predpostavimo $\forall d' \prec d : P(d')$ in $d \Vdash \langle c, \sigma_0\rangle \to \sigma_1 \;\&\; d_1 \Vdash \langle c, \sigma_0\rangle \to \sigma$.

Pokažimo zdaj po primerih, ki temeljijo na pravilih izpeljave, da $\sigma_1 = \sigma$.

$c \equiv$ skip:

$$d = d_1 = \frac{}{\langle \text{skip}, \sigma_0\rangle \to \sigma_0}$$

$c \equiv X := a$:

$$d = \frac{\begin{array}{c}\vdots\\ \langle a, \sigma_0\rangle \to m\end{array}}{\langle X := a, \sigma_0\rangle \to \sigma_0[m/X]} \qquad d_1 = \frac{\begin{array}{c}\vdots\\ \langle a, \sigma_0\rangle \to m_1\end{array}}{\langle X := a, \sigma_0\rangle \to \sigma_0[m_1/X]}$$

Velja $\sigma = \sigma_0[m/X]$ in $\sigma_1 = \sigma_0[m_1/X]$. Ker je izračun aritmetične operacije deterministično, lahko sklepamo, da $m = m_1$ in $\sigma = \sigma_1$

$c \equiv c_0; c_1$:

$$d = \frac{\begin{array}{cc}\vdots & \vdots \\ \overline{\langle c_0, \sigma_0\rangle \to \sigma'} & \overline{\langle c_1, \sigma'\rangle \to \sigma}\end{array}}{\langle c_0; c_1, \sigma_0\rangle \to \sigma} \qquad d_1 = \frac{\begin{array}{cc}\vdots & \vdots \\ \overline{\langle c_0, \sigma_0\rangle \to \sigma'_1} & \overline{\langle c_1, \sigma'_1\rangle \to \sigma_1}\end{array}}{\langle c_0; c_1, \sigma_0\rangle \to \sigma_1}$$

Naj bodo d^0 in d^1 pod-izpeljave d.

$$d^0 = \frac{\vdots}{\langle c_0, \sigma_0\rangle \to \sigma'} \qquad d^1 = \frac{\vdots}{\langle c_0, \sigma'\rangle \to \sigma}$$

Po predpostavki velja $d^0 \prec d$ in $d^1 \prec d$, torej $\sigma' = \sigma'_1$ in $\sigma = \sigma_1$.

$c \equiv$ if b then c_0 else c_1:
Izbira pravila v tem primeru je odvisna od vrednosti b. Dokazati bi bilo potrebno, da je evaluacija logičnih vrednosti v IMP deterministična, kar bomo prepustili za vajo. Ker je evaluacija b determinističa imamo lahko vrednost true ali false in ne oboje.

$$d = \frac{\begin{array}{cc}\vdots & \vdots \\ \overline{\langle b, \sigma_0\rangle \to \text{true}} & \overline{\langle c_0, \sigma_0\rangle \to \sigma}\end{array}}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma_0\rangle \to \sigma} \qquad d_1 = \frac{\begin{array}{cc}\vdots & \vdots \\ \overline{\langle b, \sigma_0\rangle \to \text{true}} & \overline{\langle c_0, \sigma_0\rangle \to \sigma_1}\end{array}}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma_0\rangle \to \sigma_1}$$

Naj bo d' pod-izpeljava d s katero izpeljemo $\langle c_0, \sigma_0\rangle \to \sigma$. Velja $d' \prec d$ in torej po predpostavki $P(d')$. Sklepamo lahko, da $\sigma = \sigma_1$. Dokaz za primer $\langle b, \sigma_0\rangle \to$ false dokažemo podobno.

$c \equiv$ while b do c:
Spet imamo dve pravili v odvisnosti od vrednosti b. Poglejmo si prvo pravilo, kjer sigurno velja, da $\sigma = \sigma_1 = \sigma_0$ v obeh izpeljavah.

$$d = \dfrac{\dfrac{\vdots}{\langle b, \sigma_0\rangle \to \text{false}}}{\langle \text{while } b \text{ do } c, \sigma_0\rangle \to \sigma_0} \qquad d_1 = \dfrac{\dfrac{\vdots}{\langle b, \sigma_0\rangle \to \text{false}}}{\langle \text{while } b \text{ do } c, \sigma_0\rangle \to \sigma_0}$$

Poglejmo si še drugo pravilo za $\langle b, \sigma_0\rangle \to \text{true}$.

$$d = \dfrac{\dfrac{\vdots}{\langle b, \sigma_0\rangle \to \text{true}} \quad \dfrac{\vdots}{\langle c, \sigma_0\rangle \to \sigma'} \quad \dfrac{\vdots}{\langle \text{while } b \text{ do } c, \sigma'\rangle \to \sigma}}{\langle \text{while } b \text{ do } c, \sigma_0\rangle \to \sigma}$$

$$d_1 = \dfrac{\dfrac{\vdots}{\langle b, \sigma_0\rangle \to \text{true}} \quad \dfrac{\vdots}{\langle c, \sigma_0\rangle \to \sigma_1'} \quad \dfrac{\vdots}{\langle \text{while } b \text{ do } c, \sigma_1'\rangle \to \sigma_1}}{\langle \text{while } b \text{ do } c, \sigma_0\rangle \to \sigma_1}$$

Naj bodo d' izpeljava stavka $\langle c, \sigma_0\rangle \to \sigma'$ in d'' izpeljava stavka $\langle \text{while } b \text{ do } c, \sigma'\rangle \to \sigma$. Velja $d' \prec d$ in $d'' \prec d$, torej po induktivni hipotezi velja $P(d')$ in $P(d'')$. Sklepamo lahko, da velja $\sigma' = \sigma_1'$ in posledično tudi $\sigma = \sigma_1$.

Pokazali smo, da v vseh primerih izračun $\langle c, \sigma_0\rangle \to \sigma$ in $\langle c, \sigma_0\rangle \to \sigma_1$ implicira $\sigma = \sigma_1$. □

4.6 Opombe

Predstavljen material o operacijski semantiki podaja pregled rezultatov uporabe operacijske semantike pri opisovanju formalnih sistemov. Poglavje vsebuje prevode izbranih sekcij učbenika G.Winskel [16, 17].