Sémantiky programovacích jazyků

Materiály ke kurzu IA011

Poslední modifikace: jaro 2023

Antonín Kučera

http://www.fi.muni.cz/usr/kucera/teaching.html

Definice programovacího jazyka

- Syntaxe definuje „správně utvořené“ programy (akceptované překladačem).
  - lexikální jednotky (klíčová slova, identifikátory, konstanty, operátory, . . .)
  - frázová struktura (určuje jaké posloupnosti lexikálních jednotek jsou „přípustné“
- Sémantika popisuje chování programu (co program „dělá“)
  - neformální (učebnice programovacích jazyků)
  - formální („matematická“)

Účel a použitelnost formální sémantiky

- Korektnost implementace
  - překladač (co musí splňovat, aby byl „správný“?)
  - optimalizátor (jaké úpravy kódu jsou „přípustné“?)
- Verifikace programů
  - vlastnosti programů (jak je vyjádřit, jak dokázat že daný program má danou vlastnost?)
  - ekvivalence programů (co znamená, že se dva programy „chovají stejně“?)
  - systémy, které jsou paralelní, distribuované, pracují s reálným časem, nebo jsou řízené událostmi, je obtížné „ladit“!

    $X := X + 1 \parallel X := X + 1$
- Návrh programovacích jazyků.

Základní „styly“ formálních sémantik

- Operační sémantika definuje jak se program provádí
- Denotační sémantika definuje co program počítá
- Axiomatická sémantika umožňuje odvodit vlastnosti programu

Abstraktní syntaxe programovacích jazyků

Definice 1 (Abstraktní syntaktická rovnice)

je rovnice tvaru

$$X \quad ::= \quad a_1 \mid \cdots \mid a_n \mid o_1(\alpha_{(1,1)}, \ldots, \alpha_{(1,n_1)}) \mid \cdots \mid o_m(\alpha_{(m,1)}, \ldots, \alpha_{(m,n_m)})$$

kde

- $a_1, \cdots, a_n$ *jsou atomy. Pro každé* a_i *je dána jeho syntaktická doména (množina)* A_i.
- $o_1, \cdots, o_m$ *jsou operační symboly, které mohou mít i nulovou aritu (pak jde o konstanty).*
- $\alpha_{(i,j)}$ *je buď* X *nebo atom (opakované výskyty jsou rozlišeny indexy).*

Abstraktní syntaxe programovacích jazyků

Příklad 2

Následující abstraktní syntaktická rovnice definuje syntaxi jednoduchých aritmetických výrazů:

$$X \quad ::= \quad num \mid \omega \mid X_0 + X_1 \mid X_0 - X_1$$

Syntaktickou doménou atomu *num* jsou dekadické zápisy celých čísel, ω je konstanta.

Syntaktické stromy

Definice 3 (Syntaktické stromy)

Uvažme abstraktní syntaktickou rovnici

$$X \quad ::= \quad a_1 \mid \cdots \mid a_n \mid o_1(\alpha_{(1,1)}, \ldots, \alpha_{(1,n_1)}) \mid \cdots \mid o_m(\alpha_{(m,1)}, \ldots, \alpha_{(m,n_m)})$$

Množina syntaktických stromů pro X je definována induktivně:

- *konstanty a prvky syntaktických domén atomů jsou syntaktické stromy pro X.*
- *strom T (rozlišujeme pořadí následníků) je syntaktickým stromem pro X pokud pro nějaké $1 \leq i \leq m$ platí:*
  - *Kořen T je operační symbol o_i.*
  - *Kořen T má přesně n_i následníků, kde j-tý následník je buď konstanta c, prvek syntaktické domény atomu a, nebo kořen syntaktického stromu pro X podle toho, zda $\alpha_{(i,j)}$ je rovno konstantě c, atomu a, nebo X.*

Příklad definice abstraktní syntaxe

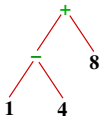
Je možné definovat i systémy syntaktických rovnic, kde množina syntaktických stromů určená jednou rovnicí definuje syntaktickou doménu atomu jiné rovnice.

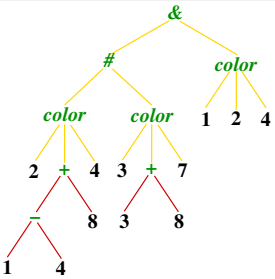
$X ::= num \mid X_0 + X_1 \mid X_0 - X_1$

Syntaktickou doménou *num* jsou dekadické zápisy celých čísel.

$Y ::= color(X_0, X_1, X_2) \mid Y_0 \# Y_1 \mid Y_0 \& Y_1$

Syntaktickou doménou X je množina všech synt. stromů pro X.

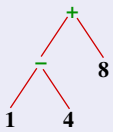




Konkrétní syntaxe programovacích jazyků

Určuje, jak jednoznačně zapisovat syntaktické stromy jako řetězce symbolů.

Aritmetické výrazy: $X ::= num \mid X_0 + X_1 \mid X_0 * X_1$



- zápis $1 - 4 + 8$ není jednoznačný
- prefixová notace: $+ \; - \; 1 \; 4 \; 8$
- postfixová notace: $8 \; 4 \; 1 \; - \; +$
- závorky: $(1 - 4) + 8$

Volitelná else klauzule příkazu if – then – else

- zápis **if** $P > 0$ **then if** $Q < 1$ **then** $K := 0$ **else** $K := 1$ není jednoznačný
- závorky
- klíčové slovo **fi**

Odvozovací systémy, důkazy a dokazatelná tvrzení

Definice 4

- *Odvozovací systém* je dán konečnou množinou *schémat axiomů* a *odvozovacích pravidel* tvaru

$$\frac{\text{předpoklad}_1 \quad \cdots \quad \text{předpoklad}_n}{\text{závěr}} \quad \text{podmínky}$$

- *Důkaz* je konečný strom, jehož listy jsou instance axiomů a vnitřní uzly instance pravidel.
- *Tvrzení* α *je dokazatelné, jestliže existuje důkaz s kořenem* α.
- *Důkazové stromy je zvykem psát „kořenem dolů“.*

Odvozovací systém výrokové logiky I

Abstraktní syntax formulí výrokové logiky

$$\varphi \quad ::= \quad \mathit{výrok} \mid \varphi_0 \to \varphi_1 \mid \neg\varphi$$

kde syntaktická doména atomu *výrok* je spočetná množina atomických výroků $\{A, B, C \ldots\}$.

Schémata axiomů odvozovacího systému výrokové logiky

- $\varphi \to (\psi \to \varphi)$
- $(\varphi \to (\psi \to \xi)) \to ((\varphi \to \psi) \to (\varphi \to \xi))$
- $(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$

Odvozovací pravidlo odvozovacího systému výrokové logiky

- $\dfrac{\varphi \qquad \varphi \to \psi}{\psi}$ „modus ponens“

Odvozovací systém výrokové logiky II

Příklad 5

$A \to A$ je dokazatelná formule:

$$\dfrac{\dfrac{A \to ((A \to A) \to A) \quad (A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))}{(A \to (A \to A)) \to (A \to A)} \qquad A \to (A \to A)}{A \to A}$$

Indukce k výšce stromu I

Buď $\mathcal{M}$ (nějaká) množina stromů konečné výšky a V (nějaká) vlastnost, která je pro každý strom $T \in \mathcal{M}$ buď pravdivá nebo nepravdivá.

Věta 6 (Tvrzení o indukci k výšce stromu)

Nechť je splněna následující podmínka:

- *Pro každé $n \in \mathbb{N}_0$ platí: Je-li V je pravdivá pro každé $T \in \mathcal{M}$ výšky menší než n, pak V je pravdivá pro každé $T' \in \mathcal{M}$ výšky právě n.*

Pak V je pravdivá pro všechny stromy z $\mathcal{M}$

Indukce k výšce stromu II

Strukturální indukce

je indukce k výšce (syntaktického) stromu, kde $\mathcal{M}$ je množina všech syntaktických stromů určená danou abstraktní syntaktickou rovnicí

Indukce k výšce odvození

indukce k výšce (důkazového) stromu, kde $\mathcal{M}$ je množina všech důkazů daného odvozovacího systému.

Abstraktní syntaxe jazyka IMP

- Základní syntaktické domény

$$\begin{array}{lcl} \textbf{Num} & = & \{0, 1, -1, 2, -2, \ldots\} \\ \textbf{Bool} & = & \{\textbf{tt}, \textbf{ff}\} \\ \textbf{Var} & = & \{A, B, C, \ldots\} \end{array}$$

- Aritmetické výrazy **Aexp**

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

kde $n \in$ **Num** a $X \in$ **Var**.

- Pravdivostní výrazy **Bexp**

$$b \ ::= \ t \mid a_0 = a_1 \mid a_0 \leq a_1 \mid \textbf{not } b \mid b_0 \textbf{ and } b_1 \mid b_0 \textbf{ or } b_1$$

kde $t \in$ **Bool** a $a_0, a_1 \in$ **Aexp**.

- Příkazy **Com**

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

kde $X \in$ **Var**, $a \in$ **Aexp** a $b \in$ **Bexp**.

Operační sémantika

Programům v daném jazyce je přiřazen přechodový systém, který popisuje výpočetní procesy jednotlivých programů.

Definice 7 (přechodový systém)

Přechodový systém je trojice $(S, \mathcal{A}, \rightarrow)$, kde

- S je množina *konfigurací* (ne nutně konečná!)
- $\mathcal{A}$ je množina *akcí*
- $\rightarrow \subseteq S \times \mathcal{A} \times S$ je *přechodová relace*

Typy operační sémantiky

Jednotlivé „typy“ operační sémantiky se liší definicí množiny konfigurací a přechodové relace

- SMC stroj: Stack – Memory – Control stack
- λ-kalkul
- SOS: Strukturální Operační Sémantika

SOS sémantika IMP 1. typu („big step“)

Definice 8 (stav programu)

Stav je zobrazení $\sigma : \mathbf{Var} \to \mathbb{Z}$. Množina všech stavů se značí Σ.

Přechodový systém příslušný SOS sémantice 1. typu

SOS sémantika 1. typu definuje přechodový systém kde

- Σ je množina stavů,
- **Com** je množina akcí,
- $\sigma \xrightarrow{c} \sigma'$ právě když program c zahájený ve stavu σ skončí a přejde do stavu σ'.

Za tímto účelem zavedeme odvozovací systémy pro tři relace:

- $\to_A \subseteq \mathbf{Aexp} \times \Sigma \times \mathbb{Z}$; prvky zapisujeme ve tvaru $\langle a, \sigma \rangle \to_A n$.
- $\to_B \subseteq \mathbf{Bexp} \times \Sigma \times \mathbb{T}$; prvky zapisujeme ve tvaru $\langle b, \sigma \rangle \to_B t$.
- $\to_C \subseteq \mathbf{Com} \times \Sigma \times \Sigma$; prvky zapisujeme ve tvaru $\langle c, \sigma \rangle \to_C \sigma'$.

Aritmetické výrazy Aexp

$\langle a, \sigma \rangle \to \mathbf{n}$ „výraz a se ve stavu σ vyhodnotí na $\mathbf{n} \in \mathbb{Z}$“

$$\langle n, \sigma \rangle \to \mathbf{n}$$

$$\langle X, \sigma \rangle \to \sigma(X)$$

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

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

$$\frac{\langle a_0, \sigma \rangle \to \mathbf{n_0} \quad \langle a_1, \sigma \rangle \to \mathbf{n_1}}{\langle a_0 * a_1, \sigma \rangle \to \mathbf{n}} \; \mathbf{n = n_0 * n_1}$$

Aritmetické výrazy Aexp

Příklad 9

Uvažme výraz $(X + 4) * (2 * Y)$ a stav σ kde $\sigma(X) = 2$ a $\sigma(Y) = 5$. Pak $\langle (X + 4) * (2 * Y), \sigma \rangle \to \mathbf{60}$, neboť

$$\dfrac{\dfrac{\langle X, \sigma \rangle \to \mathbf{2} \quad \langle 4, \sigma \rangle \to \mathbf{4}}{\langle X + 4, \sigma \rangle \to \mathbf{6}} \qquad \dfrac{\langle 2, \sigma \rangle \to \mathbf{2} \quad \langle Y, \sigma \rangle \to \mathbf{5}}{\langle 2 * Y, \sigma \rangle \to \mathbf{10}}}{\langle (X + 4) * (2 * Y), \sigma \rangle \to \mathbf{60}}$$

Pravdivostní výrazy Bexp

$\langle b, \sigma \rangle \to t$ „výraz b se ve stavu σ vyhodnotí na $t \in \mathbb{T}$“

$\langle \mathbf{tt}, \sigma \rangle \to \mathbf{true}$

$\langle \mathbf{ff}, \sigma \rangle \to \mathbf{false}$

$$\frac{\langle a_0, \sigma \rangle \to n_0 \quad \langle a_1, \sigma \rangle \to n_1}{\langle a_0 = a_1, \sigma \rangle \to \mathbf{true}} \; n_0 = n_1$$

$$\frac{\langle a_0, \sigma \rangle \to n_0 \quad \langle a_1, \sigma \rangle \to n_1}{\langle a_0 = a_1, \sigma \rangle \to \mathbf{false}} \; n_0 \neq n_1$$

$$\frac{\langle a_0, \sigma \rangle \to n_0 \quad \langle a_1, \sigma \rangle \to n_1}{\langle a_0 \leq a_1, \sigma \rangle \to \mathbf{true}} \; n_0 \leq n_1$$

$$\frac{\langle a_0, \sigma \rangle \to n_0 \quad \langle a_1, \sigma \rangle \to n_1}{\langle a_0 \leq a_1, \sigma \rangle \to \mathbf{false}} \; n_0 > n_1$$

$$\frac{\langle b, \sigma \rangle \to \mathbf{false}}{\langle \mathbf{not}\, b, \sigma \rangle \to \mathbf{true}}$$

$$\frac{\langle b, \sigma \rangle \to \mathbf{true}}{\langle \mathbf{not}\, b, \sigma \rangle \to \mathbf{false}}$$

$$\frac{\langle b_0, \sigma \rangle \to t_0 \quad \langle b_1, \sigma \rangle \to t_1}{\langle b_0 \,\mathbf{and}\, b_1, \sigma \rangle \to t} \; t = t_0 \wedge t_1$$

$$\frac{\langle b_0, \sigma \rangle \to t_0 \quad \langle b_1, \sigma \rangle \to t_1}{\langle b_0 \,\mathbf{or}\, b_1, \sigma \rangle \to t} \; t = t_0 \vee t_1$$

Příkazy Com

$\langle c, \sigma \rangle \to \sigma'$ „c aktivovaný ve stavu σ skončí ve stavu σ'“

$$\langle \textbf{skip}, \sigma \rangle \to \sigma$$

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

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

$$\frac{\langle b, \sigma \rangle \to \textbf{true} \quad \langle c_0, \sigma \rangle \to \sigma'}{\langle \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1, \sigma \rangle \to \sigma'} \qquad \frac{\langle b, \sigma \rangle \to \textbf{false} \quad \langle c_1, \sigma \rangle \to \sigma'}{\langle \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1, \sigma \rangle \to \sigma'}$$

$$\frac{\langle b, \sigma \rangle \to \textbf{false}}{\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \to \sigma} \qquad \frac{\langle b, \sigma \rangle \to \textbf{true} \quad \langle c, \sigma \rangle \to \sigma'' \quad \langle \textbf{while } b \textbf{ do } c, \sigma'' \rangle \to \sigma'}{\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \to \sigma'}$$

Příklad důkazového stromu v SOS sémantice 1. typu

Příklad 10

Uvažme program **while** $A \leq 2$ **do** $A := A + C$ a stav σ definovaný takto: $\sigma(A) = 1$, $\sigma(C) = 2$, a pro každé $B \in$ **Var** kde $A \neq B \neq C$ je $\sigma(B) = 10$. Pak $\langle$**while** $A \leq 2$ **do** $A := A + C, \sigma\rangle \to \sigma[3/A]$, neboť

$$\dfrac{\dfrac{\dfrac{\langle A,\sigma\rangle \to 1 \quad \langle 2,\sigma\rangle \to 2}{\langle A \leq 2,\sigma\rangle \to \textbf{true}} \qquad \dfrac{\langle A,\sigma\rangle \to 1 \quad \langle C,\sigma\rangle \to 2}{\langle A := A + C,\sigma\rangle \to 3}}{\langle A := A + C,\sigma\rangle \to \sigma[3/A]} \qquad \dfrac{\langle A,\sigma[3/A]\rangle \to 3 \quad \langle 2,\sigma[3/A]\rangle \to 2}{\langle A \leq 2,\sigma[3/A]\rangle \to \textbf{false}}}{\langle \textbf{while } A \leq 2 \textbf{ do } A := A + C,\sigma[3/A]\rangle \to \sigma[3/A]}}{\langle \textbf{while } A \leq 2 \textbf{ do } A := A + C,\sigma\rangle \to \sigma[3/A]}$$

Příklad 11

Nechť $\sigma' = \sigma[0/C]$, kde σ je stav z předchozího příkladu. Důkazový strom s kořenem tvaru

$$\langle \textbf{while } A \leq 2 \textbf{ do } A := A + C, \sigma'\rangle \to \sigma''$$

sestrojit nelze (pro žádné σ'').

SOS sémantika 1. typu je deterministická

Věta 12

1. *Pro každé* $a \in$ **Aexp** *a* $\sigma \in \Sigma$ *existuje právě jedno* $n \in \mathbb{Z}$ *takové, že* $\langle a, \sigma \rangle \to n$.
2. *Pro každé* $b \in$ **Bexp** *a* $\sigma \in \Sigma$ *existuje právě jedno* $t \in \mathbb{T}$ *takové, že* $\langle b, \sigma \rangle \to t$.
3. *Pro každé* $c \in$ **Com** *a* $\sigma \in \Sigma$ *existuje nejvýše jedno* $\sigma' \in \Sigma$ *takové, že* $\langle c, \sigma \rangle \to \sigma'$.

SOS sémantika 1. typu je deterministická (důkaz)

Důkaz.

(1) Indukcí ke struktuře a.

- $a \equiv n$. Pak $\langle n, \sigma \rangle \to n$ dle definice.
- $a \equiv X$. Pak $\langle X, \sigma \rangle \to \sigma(X)$ dle definice.
- $a \equiv a_0 + a_1$. Podle indukčního předpokladu
  - existuje právě jedno n_0 takové, že $\langle a_0, \sigma \rangle \to n_0$,
  - existuje právě jedno n_1 takové, že $\langle a_1, \sigma \rangle \to n_1$.

  Máme ukázat, že existuje právě jedno n takové, že $\langle a_0 + a_1, \sigma \rangle \to n$.
  - Takové n jistě existuje alespoň jedno, stačí vzít $n = n_0 + n_1$.
  - Pokud by existovala $k \neq m$ taková, že $\langle a_0 + a_1, \sigma \rangle \to k$ a $\langle a_0 + a_1, \sigma \rangle \to m$, musela by existovat k_0, k_1, m_0, m_1 taková, že $\langle a_0, \sigma \rangle \to k_0$, $\langle a_0, \sigma \rangle \to m_0$, $\langle a_1, \sigma \rangle \to k_1$, $\langle a_1, \sigma \rangle \to m_1$, $k = k_0 + k_1$ a $m = m_0 + m_1$. Podle i.p. ovšem platí $k_0 = m_0$ a $k_1 = m_1$, proto také $k = m$, což je spor.
- $a \equiv a_0 - a_1$. Podobně.
- $a \equiv a_0 * a_1$. Podobně.

(2) Indukcí ke struktuře b, podobně jako v bodě (1).

SOS sémantika 1. typu je deterministická (důkaz)

(3) Indukcí k výšce odvození $\langle c, \sigma \rangle \to \sigma'$ dokážeme, že σ' je určeno jednoznačně, tj. pokud pro uvažované c a σ existuje nějaké další $\hat{\sigma}$ takové, že $\langle c, \sigma \rangle \to \hat{\sigma}$, pak $\sigma' = \hat{\sigma}$ (důkaz nelze vést strukturální indukcí). Uvážíme možné tvary c.

- $c \equiv$ **skip**. Pak je uvažovaný strom listem tvaru $\langle$**skip**$, \sigma \rangle \to \sigma$. Pokud platí $\langle$**skip**$, \sigma \rangle \to \hat{\sigma}$, pak zjevně $\sigma = \hat{\sigma}$.
- $c \equiv X := a$. Pak kořen $\langle X := a, \sigma \rangle \to \sigma'$ má následníka $\langle a, \sigma \rangle \to n$ a platí $\sigma' = \sigma[n/X]$. Podle (1) existuje právě jedno takové n, proto σ' je určeno jednoznačně.
- $c \equiv c_0; c_1$. Pak kořen $\langle c_0; c_1, \sigma \rangle \to \sigma'$ má následníky $\langle c_0, \sigma \rangle \to \sigma''$ a $\langle c_1, \sigma'' \rangle \to \sigma'$. Podle indukčního předpokladu je σ'' i σ' určeno jednoznačně.
- $c \equiv$ **if** b **then** c_0 **else** c_1. Pak kořen $\langle$**if** b **then** c_0 **else** $c_1, \sigma \to \sigma' \rangle$ má buď následníky $\langle b, \sigma \rangle \to$ **true** a $\langle c_0, \sigma \rangle \to \sigma'$, nebo $\langle b, \sigma \rangle \to$ **false** a $\langle c_1, \sigma \rangle \to \sigma'$. Podle (2) nastává právě jedna z těchto možností, proto je σ' určeno jednoznačně.
- $c \equiv$ **while** b **do** c. Pak kořen $\langle$**while** b **do** $c, \sigma \rangle \to \sigma'$ má buď jediného následníka $\langle b, \sigma \rangle \to$ **false** a $\sigma' = \sigma$, nebo tři následníky $\langle b, \sigma \rangle \to$ **true**, $\langle c, \sigma \rangle \to \sigma''$ a $\langle$**while** b **do** $c, \sigma'' \rangle \to \sigma'$. Podle (2) nastává právě jedna z těchto možností. V prvním případě jsme hotovi ihned; v druhém použijeme indukční předpoklad podle něhož je σ'' a σ' určeno jednoznačně.

Sémantická ekvivalence výrazů a příkazů (I)

- Aritmetické výrazy **Aexp**

$$a_0 \sim a_1 \quad \stackrel{def}{\iff} \quad (\forall n \in \mathbb{Z} \;\; \forall \sigma \in \Sigma : \; \langle a_0, \sigma \rangle \to n \iff \langle a_1, \sigma \rangle \to n)$$

- Pravdivostní výrazy **Bexp**

$$b_0 \sim b_1 \quad \stackrel{def}{\iff} \quad (\forall t \in \mathbb{T} \;\; \forall \sigma \in \Sigma : \; \langle b_0, \sigma \rangle \to t \iff \langle b_1, \sigma \rangle \to t)$$

- Příkazy **Com**

$$c_0 \sim c_1 \quad \stackrel{def}{\iff} \quad (\forall \sigma, \sigma' \in \Sigma : \; \langle c_0, \sigma \rangle \to \sigma' \iff \langle c_1, \sigma \rangle \to \sigma')$$

Příklad ekvivaletních programů

Dokážeme, že **while** b **do** c $\sim$ **if** b **then** $(c;$ **while** b **do** $c)$ **else skip**

- $\langle$**while** b **do** $c, \sigma\rangle \to \sigma' \quad \Rightarrow \quad \langle$**if** b **then** $(c;$ **while** b **do** $c)$ **else skip**$, \sigma\rangle \to \sigma'$

  Jsou dvě možnosti:

  - $\dfrac{\dfrac{\dots}{\langle b, \sigma\rangle \to \textbf{false}}}{\langle \textbf{while } b \textbf{ do } c, \sigma\rangle \to \sigma}$. Pak ale také $\dfrac{\dfrac{\dots}{\langle b, \sigma\rangle \to \textbf{false}} \qquad \langle \textbf{skip}, \sigma\rangle \to \sigma}{\langle \textbf{if } b \textbf{ then } (c; \textbf{while } b \textbf{ do } c) \textbf{ else skip}, \sigma\rangle \to \sigma}$

  - $\dfrac{\dfrac{\dots}{\langle b, \sigma\rangle \to \textbf{true}} \quad \dfrac{\dots}{\langle c, \sigma\rangle \to \sigma''} \quad \dfrac{\dots}{\langle \textbf{while } b \textbf{ do } c, \sigma''\rangle \to \sigma'}}{\langle \textbf{while } b \textbf{ do } c, \sigma\rangle \to \sigma'}$. Pak ale také

    $$\dfrac{\dfrac{\dots}{\langle b, \sigma\rangle \to \textbf{true}} \quad \dfrac{\dfrac{\dots}{\langle c, \sigma\rangle \to \sigma''} \quad \dfrac{\dots}{\langle \textbf{while } b \textbf{ do } c, \sigma''\rangle \to \sigma'}}{\langle c; \textbf{while } b \textbf{ do } c, \sigma\rangle \to \sigma'}}{\langle \textbf{if } b \textbf{ then } (c; \textbf{while } b \textbf{ do } c) \textbf{ else skip}, \sigma\rangle \to \sigma'}$$

- Opačná implikace se ukáže podobně.

Operační sémantika IMP druhého typu („small step“)

- Cílem je definovat přechodový systém, kde
  - množina konfigurací je $\mathbf{Com} \times \Sigma$,
  - množina akcí je $\{\tau\}$,
  - přechodová relace odpovídá „kroku výpočtu“ programů, tj. $\langle c, \sigma \rangle \xrightarrow{\tau} \langle c', \sigma' \rangle$ právě když program c přejde ze stavu σ vykonáním jedné instrukce do stavu σ' a z tohoto stavu se dále provádí program c'.
- Definujeme odvozovací systémy pro tři relace:
  - $\mapsto_A \subseteq (\mathbf{Aexp} \times \Sigma) \times (\mathbf{Aexp} \times \Sigma)$; prvky zapisujeme ve tvaru $\langle a, \sigma \rangle \mapsto_A \langle a', \sigma' \rangle$.
  - $\mapsto_B \subseteq (\mathbf{Bexp} \times \Sigma) \times (\mathbf{Bexp} \times \Sigma)$; prvky zapisujeme ve tvaru $\langle b, \sigma \rangle \mapsto_B \langle b', \sigma' \rangle$.
  - $\mapsto_C \subseteq (\mathbf{Com} \times \Sigma) \times (\mathbf{Com} \times \Sigma)$; prvky zapisujeme ve tvaru $\langle c, \sigma \rangle \mapsto_C \langle c', \sigma' \rangle$.

  Indexy A, B, C u $\mapsto$ budou obvykle vynechány.
- Pak již lze definovat: $\langle c, \sigma \rangle \xrightarrow{\tau} \langle c', \sigma' \rangle \iff \langle c, \sigma \rangle \mapsto_C \langle c', \sigma' \rangle$

Aritmetické výrazy Aexp

- $\langle n, \sigma \rangle$ konečná konfigurace
- $\langle X, \sigma \rangle \mapsto \langle \sigma(X), \sigma \rangle$
- $\langle n_0 + n_1, \sigma \rangle \mapsto \langle m, \sigma \rangle$, kde $m = n_0 + n_1$
- $$\frac{\langle a_0, \sigma \rangle \mapsto \langle a_0', \sigma \rangle}{\langle a_0 + a_1, \sigma \rangle \mapsto \langle a_0' + a_1, \sigma \rangle}$$
- $$\frac{\langle a_1, \sigma \rangle \mapsto \langle a_1', \sigma \rangle}{\langle n + a_1, \sigma \rangle \mapsto \langle n + a_1', \sigma \rangle}$$
- podobně pro „$-$“ a „$*$“

Příklad:

- Nechť $\sigma(X) = 1$, $\sigma(Y) = 2$
- $\langle (X + 3) * Y, \sigma \rangle \mapsto \langle (1 + 3) * Y, \sigma \rangle \mapsto \langle 4 * Y, \sigma \rangle \mapsto \langle 4 * 2, \sigma \rangle \mapsto \langle 8, \sigma \rangle$

Pravdivostní výrazy Bexp

- $\langle \mathbf{tt}, \sigma \rangle$ konečná konfigurace
- $\langle \mathbf{ff}, \sigma \rangle$ konečná konfigurace
- $\langle n_0 = n_1, \sigma \rangle \mapsto \langle \mathbf{tt}, \sigma \rangle$, je-li $n_0 = n_1$
- $\langle n_0 = n_1, \sigma \rangle \mapsto \langle \mathbf{ff}, \sigma \rangle$, je-li $n_0 \neq n_1$
- $$\frac{\langle a_0, \sigma \rangle \mapsto \langle a_0', \sigma \rangle}{\langle a_0 = a_1, \sigma \rangle \mapsto \langle a_0' = a_1, \sigma \rangle}$$
- $$\frac{\langle a_1, \sigma \rangle \mapsto \langle a_1', \sigma \rangle}{\langle n = a_1, \sigma \rangle \mapsto \langle n = a_1', \sigma \rangle}$$
- podobně pro „$\leq$“

- $\langle \mathbf{not\ tt}, \sigma \rangle \mapsto \langle \mathbf{ff}, \sigma \rangle$, $\langle \mathbf{not\ ff}, \sigma \rangle \mapsto \langle \mathbf{tt}, \sigma \rangle$
- $$\frac{\langle b, \sigma \rangle \mapsto \langle b', \sigma \rangle}{\langle \mathbf{not}\ b, \sigma \rangle \mapsto \langle \mathbf{not}\ b', \sigma \rangle}$$
- $\langle t_1 \mathbf{\ and\ } t_2, \sigma \rangle \mapsto \langle \mathbf{tt}, \sigma \rangle$ je-li $t_1 = \mathbf{tt}$ a $t_2 = \mathbf{tt}$
- $\langle t_1 \mathbf{\ and\ } t_2, \sigma \rangle \mapsto \langle \mathbf{ff}, \sigma \rangle$ je-li $t_1 = \mathbf{ff}$ nebo $t_2 = \mathbf{ff}$
- $$\frac{\langle b_0, \sigma \rangle \mapsto \langle b_0', \sigma \rangle}{\langle b_0 \mathbf{\ and\ } b_1, \sigma \rangle \mapsto \langle b_0' \mathbf{\ and\ } b_1, \sigma \rangle}$$
- $$\frac{\langle b_1, \sigma \rangle \mapsto \langle b_1', \sigma \rangle}{\langle t \mathbf{\ and\ } b_1, \sigma \rangle \mapsto \langle t \mathbf{\ and\ } b_1', \sigma \rangle} \quad t \in \{\mathbf{tt}, \mathbf{ff}\}$$
- podobně pro „**or**“

Příkazy Com

- $\langle \textbf{skip}, \sigma \rangle$ konečná konfigurace
- $\langle X := n, \sigma \rangle \mapsto \langle \textbf{skip}, \sigma[n/X] \rangle$
- $\dfrac{\langle a, \sigma \rangle \mapsto \langle a', \sigma \rangle}{\langle X := a, \sigma \rangle \mapsto \langle X := a', \sigma \rangle}$
- $\langle \textbf{skip}; c, \sigma \rangle \mapsto \langle c, \sigma \rangle$
- $\dfrac{\langle c_0, \sigma \rangle \mapsto \langle c_0', \sigma' \rangle}{\langle c_0; c_1, \sigma \rangle \mapsto \langle c_0'; c_1, \sigma' \rangle}$
- $\langle \textbf{if tt then } c_0 \textbf{ else } c_1, \sigma \rangle \mapsto \langle c_0, \sigma \rangle$
- $\langle \textbf{if ff then } c_0 \textbf{ else } c_1, \sigma \rangle \mapsto \langle c_1, \sigma \rangle$
- $\dfrac{\langle b, \sigma \rangle \mapsto \langle b', \sigma \rangle}{\langle \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1, \sigma \rangle \mapsto \langle \textbf{if } b' \textbf{ then } c_0 \textbf{ else } c_1, \sigma \rangle}$
- $\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \mapsto \langle \textbf{if } b \textbf{ then } (c; \textbf{while } b \textbf{ do } c) \textbf{ else skip}, \sigma \rangle$

Sémantická ekvivalence výrazů a příkazů (II)

- Pro každé $k \in \mathbb{N}_0$ definujeme (induktivně) relaci $\mapsto^k \subseteq (\mathbf{Com} \times \Sigma) \times (\mathbf{Com} \times \Sigma)$:

$$\begin{array}{lcl} \mapsto^0 & = & id_{\mathbf{Com}\times\Sigma} \\ \mapsto^{i+1} & = & \mapsto^i \circ \mapsto \end{array}$$

- Dále definujeme $\mapsto^* = \bigcup_{k=0}^{\infty} \mapsto^k$
- Aritmetické výrazy **Aexp**

$$a_0 \approx a_1 \quad \stackrel{def}{\iff} \quad (\forall n \in \mathbb{Z} \;\; \forall \sigma \in \Sigma : \; \langle a_0, \sigma \rangle \mapsto^* \langle n, \sigma \rangle \iff \langle a_1, \sigma \rangle \mapsto^* \langle n, \sigma \rangle)$$

- Pravdivostní výrazy **Bexp**

$$b_0 \approx b_1 \quad \stackrel{def}{\iff} \quad (\forall t \in \mathbb{T} \;\; \forall \sigma \in \Sigma : \; \langle b_0, \sigma \rangle \mapsto^* \langle t, \sigma \rangle \iff \langle b_1, \sigma \rangle \mapsto^* \langle t, \sigma \rangle)$$

- Příkazy **Com**

$$c_0 \approx c_1 \quad \stackrel{def}{\iff} \quad (\forall \sigma, \sigma' \in \Sigma : \; \langle c_0, \sigma \rangle \mapsto^* \langle \mathbf{skip}, \sigma' \rangle \iff \langle c_1, \sigma \rangle \mapsto^* \langle \mathbf{skip}, \sigma' \rangle)$$

Ekvivalence SOS sémantik 1. a 2. typu

Lema 13

1. *Jestliže* $\langle a,\sigma\rangle \mapsto^k \langle a',\sigma\rangle$, *pak* $\langle a \odot a_1,\sigma\rangle \mapsto^k \langle a' \odot a_1,\sigma\rangle$ *a* $\langle n \odot a,\sigma\rangle \mapsto^k \langle n \odot a',\sigma\rangle$ *pro každé* $\odot \in \{+,-,*\}$.
2. *Jestliže* $\langle a,\sigma\rangle \mapsto^k \langle a',\sigma\rangle$, *pak* $\langle X := a,\sigma\rangle \mapsto^k \langle X := a',\sigma\rangle$
3. *Jestliže* $\langle c,\sigma\rangle \mapsto^k \langle c',\sigma'\rangle$, *pak* $\langle c; c_1,\sigma\rangle \mapsto^k \langle c'; c_1,\sigma'\rangle$.
4. *Jestliže* $\langle b,\sigma\rangle \mapsto^k \langle b',\sigma\rangle$, *pak* $\langle \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1,\sigma\rangle \mapsto^k \langle \textbf{if } b' \textbf{ then } c_0 \textbf{ else } c_1,\sigma\rangle$.

Důkaz. Indukcí ke k.

Ekvivalence SOS sémantik 1. a 2. typu

Lema 14

1. *Jestliže* $\langle a_0 \odot a_1, \sigma\rangle \mapsto^k \langle n, \sigma\rangle$ *kde* $\odot \in \{+, -, *\}$, *pak* $\langle a_0 \odot a_1, \sigma\rangle \mapsto^l \langle n_0 \odot a_1, \sigma\rangle \mapsto^m \langle n_0 \odot n_1, \sigma\rangle \mapsto \langle n, \sigma\rangle$, *kde* $n = n_0 \odot n_1$, $\langle a_0, \sigma\rangle \mapsto^l \langle n_0, \sigma\rangle$ *a* $\langle a_1, \sigma\rangle \mapsto^m \langle n_1, \sigma\rangle$.
2. *Jestliže* $\langle X := a, \sigma\rangle \mapsto^k \langle \textbf{skip}, \sigma'\rangle$, *pak* $\langle X := a, \sigma\rangle \mapsto^{k-1} \langle X := n, \sigma\rangle \mapsto \langle \textbf{skip}, \sigma[n/X]\rangle$, *kde* $\sigma' = \sigma[n/X]$ *a* $\langle a, \sigma\rangle \mapsto^{k-1} \langle n, \sigma\rangle$.
3. *Jestliže* $\langle c_0; c_1, \sigma\rangle \mapsto^k \langle \textbf{skip}, \sigma'\rangle$, *pak* $\langle c_0; c_1, \sigma\rangle \mapsto^l \langle \textbf{skip}; c_1, \sigma''\rangle \mapsto \langle c_1, \sigma''\rangle \mapsto^m \langle \textbf{skip}, \sigma'\rangle$, *kde* $l, m < k$ *a* $\langle c_0, \sigma\rangle \mapsto^l \langle \textbf{skip}, \sigma''\rangle$.
4. *Jestliže* $\langle$**if** b **then** c_0 **else** $c_1, \sigma\rangle \mapsto^k \langle \textbf{skip}, \sigma'\rangle$, *pak platí jedna z následujících možností:*
   - $\langle$**if** b **then** c_0 **else** $c_1, \sigma\rangle \mapsto^l \langle$**if tt then** c_0 **else** $c_1, \sigma\rangle \mapsto \langle c_0, \sigma\rangle \mapsto^m \langle \textbf{skip}, \sigma'\rangle$, *kde* $l, m < k$ *a* $\langle b, \sigma\rangle \mapsto^l \langle \textbf{tt}, \sigma\rangle$.
   - $\langle$**if** b **then** c_0 **else** $c_1, \sigma\rangle \mapsto^l \langle$**if ff then** c_0 **else** $c_1, \sigma\rangle \mapsto \langle c_1, \sigma\rangle \mapsto^m \langle \textbf{skip}, \sigma'\rangle$, *kde* $l, m < k$ *a* $\langle b, \sigma\rangle \mapsto^l \langle \textbf{ff}, \sigma\rangle$.

Ekvivalence SOS sémantik 1. a 2. typu

Věta 15

1. *Pro každé* a, σ *a* n *platí:* $\langle a,\sigma\rangle \to n \iff \langle a,\sigma\rangle \mapsto^* \langle n,\sigma\rangle$
2. *Pro každé* b *a* σ *platí:*
   - $\langle b,\sigma\rangle \to \mathbf{true} \iff \langle b,\sigma\rangle \mapsto^* \langle \mathbf{tt},\sigma\rangle$
   - $\langle b,\sigma\rangle \to \mathbf{false} \iff \langle b,\sigma\rangle \mapsto^* \langle \mathbf{ff},\sigma\rangle$
3. *Pro každé* c *a* σ,σ' *platí:* $\langle c,\sigma\rangle \to \sigma' \iff \langle c,\sigma\rangle \mapsto^* \langle \mathbf{skip},\sigma'\rangle$

Ekvivalence SOS sémantik 1. a 2. typu

Důkaz.

ad 1. Indukcí ke struktuře a.

- $a \equiv n$. Pak $\langle n, \sigma \rangle \to n$ a $\langle n, \sigma \rangle \mapsto^0 \langle n, \sigma \rangle$ dle definice.
- $a \equiv X$. Pak $\langle X, \sigma \rangle \to \sigma(X)$ a $\langle X, \sigma \rangle \mapsto \langle \sigma(X), \sigma \rangle$ dle definice.
- $a \equiv a_0 + a_1$. Podle I.P. platí
  - $\langle a_0, \sigma \rangle \to n_0 \iff \langle a_0, \sigma \rangle \mapsto^* \langle n_0, \sigma \rangle$
  - $\langle a_1, \sigma \rangle \to n_1 \iff \langle a_1, \sigma \rangle \mapsto^* \langle n_1, \sigma \rangle$

  Dále

  $\langle a_0 + a_1, \sigma \rangle \to n \iff \langle a_0, \sigma \rangle \to n_0$ a $\langle a_1, \sigma \rangle \to n_1$ kde $n = n_0 + n_1 \iff$

  $\langle a_0, \sigma \rangle \mapsto^* \langle n_0, \sigma \rangle$ a $\langle a_1, \sigma \rangle \mapsto^* \langle n_1, \sigma \rangle$ kde $n = n_0 + n_1$ (podle I.P.) $\iff$

  $\langle a_0 + a_1, \sigma \rangle \mapsto^* \langle n_0 + a_1 \rangle \mapsto^* \langle n_0 + n_1, \sigma \rangle \mapsto \langle n, \sigma \rangle$ kde $n = n_0 + n_1$ (lema 13 (1)) $\iff$

  $\langle a_0 + a_1, \sigma \rangle \mapsto^* \langle n, \sigma \rangle$ (podle lematu 14 (1))
- $a \equiv a_0 - a_1$. Podobně.
- $a \equiv a_0 * a_1$. Podobně.

Ekvivalence SOS sémantik 1. a 2. typu

ad 2. Indukcí ke struktuře b.

ad 3.

($\Rightarrow$) Indukcí k výšce odvození $\langle c, \sigma \rangle \to \sigma'$. Uvážíme možné tvary c a pravidlo, které mohlo být k odvození $\langle c, \sigma \rangle \to \sigma'$ použito.

($\Leftarrow$) Indukcí ke k pro které $\langle c, \sigma \rangle \mapsto^k \langle \textbf{skip}, \sigma' \rangle$.

Denotační sémantika IMP

- $\mathcal{A} : \textbf{Aexp} \to (\Sigma \to \mathbb{Z})$
- $\mathcal{B} : \textbf{Bexp} \to (\Sigma \to \mathbb{T})$
- $\mathcal{C} : \textbf{Com} \to (\Sigma \to \Sigma)$

Argumenty funkcí $\mathcal{A}, \mathcal{B}, \mathcal{C}$ se píší do „sémantických“ závorek $[\![\ ]\!]$

Aritmetické výrazy Aexp

- $\mathcal{A}[\![n]\!]\sigma = n$
- $\mathcal{A}[\![X]\!]\sigma = \sigma(X)$
- $\mathcal{A}[\![a_0 + a_1]\!]\sigma = \mathcal{A}[\![a_0]\!]\sigma + \mathcal{A}[\![a_1]\!]\sigma$
- $\mathcal{A}[\![a_0 - a_1]\!]\sigma = \mathcal{A}[\![a_0]\!]\sigma - \mathcal{A}[\![a_1]\!]\sigma$
- $\mathcal{A}[\![a_0 * a_1]\!]\sigma = \mathcal{A}[\![a_0]\!]\sigma * \mathcal{A}[\![a_1]\!]\sigma$

Pravdivostní výrazy Bexp

- $\mathcal{B}[\![\mathbf{tt}]\!]\sigma = \mathbf{true}$
- $\mathcal{B}[\![\mathbf{ff}]\!]\sigma = \mathbf{false}$
- $\mathcal{B}[\![a_0 = a_1]\!]\sigma = \mathcal{A}[\![a_0]\!]\sigma = \mathcal{A}[\![a_1]\!]\sigma$
- $\mathcal{B}[\![a_0 \leq a_1]\!]\sigma = \mathcal{A}[\![a_0]\!]\sigma \leq \mathcal{A}[\![a_1]\!]\sigma$
- $\mathcal{B}[\![\mathbf{not}\ b]\!]\sigma = \neg\mathcal{B}[\![b]\!]\sigma$
- $\mathcal{B}[\![b_0\ \mathbf{and}\ b_1]\!]\sigma = \mathcal{B}[\![b_0]\!]\sigma \wedge \mathcal{B}[\![b_1]\!]\sigma$
- $\mathcal{B}[\![b_0\ \mathbf{or}\ b_1]\!]\sigma = \mathcal{B}[\![b_0]\!]\sigma \vee \mathcal{B}[\![b_1]\!]\sigma$

Příkazy Com

- $C[\![\textbf{skip}]\!]\sigma = \sigma$
- $C[\![X := a]\!]\sigma = \sigma[\mathcal{A}[\![a]\!]\sigma / X]$
- $C[\![c_0; c_1]\!]\sigma = C[\![c_1]\!](C[\![c_0]\!]\sigma) = (C[\![c_1]\!] \circ C[\![c_0]\!])\sigma$
- $C[\![\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1]\!]\sigma = \begin{cases} C[\![c_0]\!]\sigma & \text{jestliže } \mathcal{B}[\![b]\!]\sigma = \textbf{true} \\ C[\![c_1]\!]\sigma & \text{jestliže } \mathcal{B}[\![b]\!]\sigma = \textbf{false} \end{cases}$
- $C[\![\textbf{while } b \textbf{ do } c]\!]\sigma = ???$

Úplné částečné uspořádání (CPO)

Definice 16

Uspořádaná množina $(D, \sqsubseteq)$ *je CPO, pokud každý nekonečný řetěz*

$$d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq d_3 \cdots$$

prvků z D *má v* D *suprémum.*

Příklady:

- Každá konečná uspořádaná množina je CPO.
- Každá množina M uspořádaná identitou je CPO (tzv. diskrétní CPO).
- Je-li M množina, je $(2^M, \subseteq)$ CPO.
- Každý úplný svaz je CPO.
- $(\Sigma \to \Sigma, \subseteq)$ je CPO.
  - $f \subseteq g$ je-li g „více definovaná“ než f.

Monotónní a spojité funkce

Definice 17

Nechť $(D, \sqsubseteq)$, $(E, \preceq)$ jsou CPO, $f : D \to E$ totální funkce. f je monotónní, jestliže pro každé $a, b \in D$ platí:

$$a \sqsubseteq b \quad \Rightarrow \quad f(a) \preceq f(b).$$

f je spojitá, je-li monotónní a pro každý nekonečný řetěz

$$a_0 \sqsubseteq a_1 \sqsubseteq a_2 \sqsubseteq a_3 \cdots$$

prvků z D platí

$$\bigvee_{i \in \mathbb{N}} f(a_i) \;=\; f\left(\bigsqcup_{i \in \mathbb{N}} a_i \right)$$

Příklad:

- Každá funkce z diskrétního CPO je spojitá.
- Funkce $f : \mathbb{N}_\infty \to \{0, 1\}$, kde $f(i) = 0$ a $f(\infty) = 1$, je monotonní, ale není spojitá.

Postačující podmínka spojitosti

Věta 18

Buď M množina, $f : 2^M \to 2^M$ taková, že pro každé $A \subseteq M$ platí

$$f(A) = \bigcup_{a \in A} f(\{a\})$$

Pak f je spojitá funkce na CPO $(2^M, \subseteq)$.

Důkaz.

- Monotonie: Nechť $A \subseteq B$. Pak

$$f(A) \quad = \quad \bigcup_{a \in A} f(\{a\}) \quad \subseteq \quad \bigcup_{a \in B} f(\{a\}) \quad = \quad f(B).$$

- Spojitost: Buď $A_1 \subseteq A_2 \subseteq A_3 \subseteq A_4 \cdots$ nekonečný řetěz podmnožin M. Pak

$$\bigcup_{i \in \mathbb{N}} f(A_i) \quad = \quad \bigcup_{i \in \mathbb{N}} \bigcup_{a \in A_i} f(\{a\}) \quad = \quad f\left(\bigcup_{i \in \mathbb{N}} A_i \right).$$

Věta o pevném bodě

Věta 19

Buď $(D, \sqsubseteq)$ CPO mající nejmenší prvek $\bot$ a Γ spojitá funkce na D. Položme

$$\mu\Gamma \quad = \quad \bigsqcup_{i \in \mathbb{N}_0} \Gamma^i(\bot).$$

Pak $\mu\Gamma$ je nejmenší pevný bod Γ.

Důkaz.

- $\mu\Gamma$ je pevný bod Γ: Pro každé $i \in \mathbb{N}_0$ platí $\Gamma^i(\bot) \sqsubseteq \Gamma^{i+1}(\bot)$ (snadno indukcí k i, použije se monotonie Γ). Dále

$$\Gamma(\mu\Gamma) \quad = \quad \Gamma\left(\bigsqcup_{i \in \mathbb{N}_0} \Gamma^i(\bot)\right) \quad = \quad \bigsqcup_{i \in \mathbb{N}_0} \Gamma^{i+1}(\bot) \quad = \quad \bigsqcup_{i \in \mathbb{N}_0} \Gamma^i(\bot) \quad = \quad \mu\Gamma$$

Věta o pevném bodě (2)

- $\mu\Gamma$ je nejmenší pevný bod Γ: Buď d pevný bod Γ (tj. $\Gamma(d) = d$). Stačí ukázat, že d je horní závora množiny $\{\Gamma^i(\bot) \mid i \in \mathbb{N}_0\}$. Pak $\mu\Gamma \sqsubseteq d$ podle definice supréma.

  Indukcí k i dokážeme, že $\Gamma^i(\bot) \sqsubseteq d$ pro každé $i \in \mathbb{N}_0$. Zřejmě $\Gamma^0(\bot) = \bot \sqsubseteq d$; a platí-li $\Gamma^i(\bot) \sqsubseteq d$, pak také $\Gamma^{i+1}(\bot) \sqsubseteq \Gamma(d) = d$ neboť Γ je monotónní a d je pevný bod.

Věta o pevném bodě. Aplikace.

- Uvažme následující gramatiku s kořenem A:

$$A \quad \rightarrow \quad aA \mid bA \mid c$$

- Uvažme jazyk generovaný touto gramatikou. Tento jazyk zjevně splňuje rovnici

$$L \quad = \quad a.L \ \cup \ b.L \ \cup \ \{c\}$$

- Na tuto rovnici nelze nahlížet jako na rovnici definitorickou (definice kruhem).
- Pravá strana rovnice určuje spojitou funkci $\Gamma : 2^{\{a,b,c\}^*} \rightarrow 2^{\{a,b,c\}^*}$, která pro daný jazyk L vrací jazyk $a.L \ \cup \ b.L \ \cup \ \{c\}$. Jazyk určený uvedenou gramatikou nyní můžeme definovat jako $\mu\Gamma$.
- Platí $\mu\Gamma = \bigcup_{i=0} \Gamma^i(\emptyset)$, kde
  - $\Gamma^0(\emptyset) = \emptyset$
  - $\Gamma^1(\emptyset) = \Gamma(\emptyset) = \{c\}$
  - $\Gamma^2(\emptyset) = \Gamma(\Gamma^1(\emptyset)) = \Gamma(\{c\}) = \{c, ac, bc\}$
  - $\vdots$

Věta o pevném bodě. Aplikace.

- Uvažme následující gramatiku:

$$\begin{array}{lcl} A & \to & aA \mid bB \mid c \\ B & \to & bB \mid aA \end{array}$$

- Jazyky generované neterminály A a B splňují rovnice

$$\begin{array}{lcl} L_A & = & a.L_A \cup b.L_B \cup \{c\} \\ L_B & = & b.L_B \cup a.L_A \end{array}$$

  které určují funkci $\Gamma : 2^{\{a,b,c\}^*} \times 2^{\{a,b,c\}^*} \to 2^{\{a,b,c\}^*} \times 2^{\{a,b,c\}^*}$ předpisem

$$\Gamma(L_A, L_B) \quad = \quad (a.L_A \cup b.L_B \cup \{c\},\ b.L_B \cup a.L_A)$$

- Dvojici jazyků určenou neterminály A, B lze nyní definovat jako $\mu\Gamma$. Platí $\mu\Gamma = \bigcup_{i=0} \Gamma^i(\emptyset,\emptyset)$, kde
  - $\Gamma^0(\emptyset,\emptyset) = (\emptyset,\emptyset)$
  - $\Gamma^1(\emptyset,\emptyset) = \Gamma(\emptyset,\emptyset) = (\{c\},\emptyset)$
  - $\Gamma^2(\emptyset,\emptyset) = \Gamma(\Gamma^1(\emptyset,\emptyset)) = \Gamma(\{c\},\emptyset) = (\{c, ac\},\{ac\})$
  - $\Gamma^3(\emptyset,\emptyset) = \Gamma(\Gamma^2(\emptyset,\emptyset)) = \Gamma(\{c, ac\},\{ac\}) = (\{c, ac, aac, bac\},\{ac, bac, ac, aac\})$

Věta o pevném bodě. Aplikace.

- Pomocí věty o pevném bodě lze dát přesný smysl systému rekurzivních rovnic

$$\begin{aligned} X_1 &= g_1(X_1, \ldots, X_m) \\ &\vdots \\ X_m &= g_m(X_1, \ldots, X_m) \end{aligned}$$

  tak, že se z pravých stran (výrazů $g_1, \ldots, g_m$) vytvoří funkce

$$\Gamma : C_1 \times \cdots \times C_m \to C_1 \times \cdots \times C_m$$

  kde $C_1, \ldots, C_m$ jsou domény objektů zastoupených pomocí proměnných $X_1, \ldots, X_m$.

- Pokud jsou $g_1, \ldots, g_m$ vytvořeny „bezpečným“ způsobem, je funkce Γ vždy spojitá a uvedený systém rovnic definuje m-tici objektů $\mu\Gamma$.

- Tímto způsobem lze zejména dát jasný smysl systému rekurzivně definovaných funkcí, tedy definovat denotační sémantiku funkcionálních programů.

Denotační sémantika while cyklu

- Označme $w \equiv \textbf{while } b \textbf{ do } c$.
- Platí $w \sim \textbf{if } b \textbf{ then } c; w \textbf{ else skip}$.
- Proto by mělo platit také $\mathcal{C}[\![w]\!] = \mathcal{C}[\![\textbf{if } b \textbf{ then } c; w \textbf{ else skip}]\!]$
- Tedy

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

- Tuto rovnost nelze chápat definitoricky, ale lze na ni nahlížet jako na „návod“, jak pro danou aproximaci $\mathcal{C}[\![w]\!]$ spočítat „lepší“ aproximaci.

Denotační sémantika while cyklu (2)

- Definujeme funkci $\Gamma : (\Sigma \to \Sigma) \to (\Sigma \to \Sigma)$ předpisem

$$\begin{array}{rcl} \Gamma(\varphi) & = & \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \textbf{true} \ \wedge \ (\sigma,\sigma') \in \varphi \circ \mathcal{C}[\![c]\!]\} \\ & \cup & \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]\sigma = \textbf{false}\} \end{array}$$

- Γ je totální a spojitá funkce na CPO $(\Sigma \to \Sigma, \subseteq)$, neboť

$$\Gamma(\varphi) \ = \ \bigcup_{(\sigma,\sigma') \in \varphi} \{\Gamma(\{(\sigma,\sigma')\})\}$$

  a lze tedy aplikovat větu 18.

- $\mathcal{C}[\![w]\!]$ by mělo být pevným bodem funkce Γ, tj. $\Gamma(\mathcal{C}[\![w]\!]) = \mathcal{C}[\![w]\!]$.

Denotační sémantika while cyklu (3)

- Γ může mít více pevných bodů; má-li však $C[\![w]\!]$ odpovídat intuitivnímu významu **while** cyklu, je třeba definovat

  $$C[\![w]\!] \;=\; \mu\Gamma$$

  Nejmenší pevný bod Γ existuje podle věty 19 a vypadá takto:

  $$\mu\Gamma \;=\; \bigcup_{i\in\mathbb{N}_0} \Gamma^i(\emptyset)$$

- Pozorování: $(\sigma,\sigma') \in \Gamma^i(\emptyset)$ právě když **while** b **do** c aktivovaný ve stavu σ skončí po nejvýše $i-1$ iteracích ve stavu σ'.

Denotační sémantika while cyklu – příklady

- **while** $X \leq 1$ **do** $X := X + 1$

$$
\begin{aligned}
\Gamma^0(\emptyset) &= \emptyset \\
\Gamma^1(\emptyset) &= \{(\sigma,\sigma) \mid \sigma(X) > 1\} \\
\Gamma^2(\emptyset) &= \Gamma^1(\emptyset) \cup \{(\sigma,\sigma[2/X]) \mid \sigma(X) = 1\} \\
\Gamma^3(\emptyset) &= \Gamma^2(\emptyset) \cup \{(\sigma,\sigma[2/X]) \mid \sigma(X) = 0\} \\
\Gamma^4(\emptyset) &= \Gamma^3(\emptyset) \cup \{(\sigma,\sigma[2/X]) \mid \sigma(X) = -1\} \\
&\vdots
\end{aligned}
$$

Obecně $\Gamma^{i+1}(\emptyset) = \Gamma^i(\emptyset) \cup \{(\sigma,\sigma[2/X]) \mid \sigma(X) = 2 - i\}$ pro každé $i \geq 1$.

- **while tt do** $X := X + 1$

$$
\begin{aligned}
\Gamma^0(\emptyset) &= \emptyset \\
\Gamma^1(\emptyset) &= \emptyset
\end{aligned}
$$

V tomto případě tedy $\mu\Gamma = \Gamma^0(\emptyset) = \emptyset$.

Denotační sémantika while cyklu – příklady (2)

- **while ff do** $X := X + 1$

$$\begin{aligned} \Gamma^0(\emptyset) &= \emptyset \\ \Gamma^1(\emptyset) &= \{(\sigma,\sigma) \mid \sigma \in \Sigma\} \\ \Gamma^2(\emptyset) &= \{(\sigma,\sigma) \mid \sigma \in \Sigma\} \end{aligned}$$

V tomto případě tedy $\mu\Gamma = \Gamma^1(\emptyset)$.

Ekvivalence operační a denotační sémantiky

Věta 20

1. *Pro každé* a, σ *a* n *platí:* $\langle a,\sigma\rangle \to n \iff \mathcal{A}[\![a]\!]\sigma = n$
2. *Pro každé* b, σ *a* t *platí:* $\langle b,\sigma\rangle \to t \iff \mathcal{B}[\![b]\!]\sigma = t$
3. *Pro každé* c *a* σ,σ' *platí:* $\langle c,\sigma\rangle \to \sigma' \iff C[\![c]\!]\sigma = \sigma'$

Důkaz. 1. a 2. indukcí ke struktuře a a b.

ad 3., „⇒“ Indukcí k výšce odvození $\langle c,\sigma\rangle \to \sigma'$. Uvážíme možné tvary c.

- $\langle \mathbf{skip},\sigma\rangle \to \sigma$. Platí $C[\![\mathbf{skip}]\!]\sigma = \sigma$ podle definice.
- $c \equiv X := a$. Pak kořen $\langle X := a,\sigma\rangle \to \sigma'$ má následníka $\langle a,\sigma\rangle \to n$ a platí $\sigma' = \sigma[n/X]$. Podle 1. $\mathcal{A}[\![a]\!]\sigma = n$ a $C[\![X := a]\!]\sigma = \sigma[n/X]$ dle definice.
- $c \equiv c_0; c_1$. Pak kořen $\langle c_0; c_1,\sigma\rangle \to \sigma'$ má následníky $\langle c_0,\sigma\rangle \to \sigma''$ a $\langle c_1,\sigma''\rangle \to \sigma'$. Podle I.P. $C[\![c_0]\!]\sigma = \sigma''$ a $C[\![c_1]\!]\sigma'' = \sigma'$, proto $(\sigma,\sigma') \in C[\![c_1]\!] \circ C[\![c_0]\!] = C[\![c_0; c_1]\!]$.

Ekvivalence operační a denotační sémantiky (2)

- $c \equiv$ **if** b **then** c_0 **else** c_1. Pak kořen $\langle$**if** b **then** c_0 **else** $c_1, \sigma\rangle \to \sigma'$ má buď následníky $\langle b, \sigma\rangle \to$ **true** a $\langle c_0, \sigma\rangle \to \sigma'$, nebo $\langle b, \sigma\rangle \to$ **false** a $\langle c_1, \sigma\rangle \to \sigma'$. V prvém případě $\mathcal{B}[\![b]\!]\sigma =$ **true** a $\mathcal{C}[\![c_0]\!]\sigma = \sigma'$ (podle I.P. a 2.), tedy $(\sigma, \sigma') \in \mathcal{C}[\![$**if** b **then** c_0 **else** $c_1]\!]$ podle definice. Druhý případ se dokáže podobně.

- $c \equiv$ **while** b **do** c. Pak kořen $\langle$**while** b **do** $c, \sigma\rangle \to \sigma'$ má buď
  - jediného následníka $\langle b, \sigma\rangle \to$ **false** a platí $\sigma' = \sigma$. Pak $\mathcal{B}[\![b]\!]\sigma =$ **false** podle 2., proto $(\sigma, \sigma) \in \Gamma(\emptyset) \subseteq \mu\Gamma$;
  - nebo tři následníky $\langle b, \sigma\rangle \to$ **true**, $\langle c, \sigma\rangle \to \sigma''$ a $\langle$**while** b **do** $c, \sigma''\rangle \to \sigma'$. Pak $\mathcal{B}[\![b]\!]\sigma =$ **true** podle 2. a $(\sigma, \sigma'') \in \mathcal{C}[\![c]\!]$, $(\sigma'', \sigma') \in \mathcal{C}[\![$**while** b **do** $c]\!]$ podle I.P. Podle definice $\mu\Gamma$ existuje $k \in \mathbb{N}_0$ takové, že $(\sigma'', \sigma') \in \Gamma^k(\emptyset)$. Dále podle definice Γ dostáváme, že $(\sigma, \sigma') \in \Gamma^{k+1}(\emptyset)$, tedy $(\sigma, \sigma') \in \mu\Gamma$.

Ekvivalence operační a denotační sémantiky (3)

„⇐“ Indukcí ke struktuře c.

- $c \equiv \textbf{skip}$. Platí $\mathcal{C}[\![\textbf{skip}]\!]\sigma = \sigma$ a $\langle \textbf{skip}, \sigma \rangle \to \sigma$ podle definice.
- $c \equiv X := a$. Platí $\mathcal{C}[\![X := a]\!]\sigma = \sigma[n/X]$ kde $\mathcal{A}[\![a]\!]\sigma = n$. Podle 1. $\langle a, \sigma \rangle \to n$, proto $\langle X := a, \sigma \rangle \to \sigma[n/X]$.
- $c \equiv c_0; c_1$. Jestliže $(\sigma, \sigma') \in \mathcal{C}[\![c_0; c_1]\!] = \mathcal{C}[\![c_1]\!] \circ \mathcal{C}[\![c_0]\!]$, existuje σ'' takové, že $(\sigma, \sigma'') \in \mathcal{C}[\![c_0]\!]$ a $(\sigma'', \sigma') \in \mathcal{C}[\![c_1]\!]$. Podle I.P. platí $\langle c_0, \sigma \rangle \to \sigma''$ a $\langle c_1, \sigma'' \rangle \to \sigma'$, tedy $\langle c_0; c_1, \sigma \rangle \to \sigma'$.
- $c \equiv \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1$. Jestliže $(\sigma, \sigma') \in \mathcal{C}[\![\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1]\!]$, jsou dvě možnosti:
  - $\mathcal{B}[\![b]\!]\sigma = \textbf{true}$ a $(\sigma, \sigma') \in \mathcal{C}[\![c_0]\!]$. Podle 2. a I.P. platí $\langle b, \sigma \rangle \to \textbf{true}$ a $\langle c_0, \sigma \rangle \to \sigma'$, proto $\langle \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1, \sigma \rangle \to \sigma'$.
  - Druhá možnost se ověří podobně.

Ekvivalence operační a denotační sémantiky (4)

- $c \equiv$ **while** b **do** c. Jestliže $(\sigma,\sigma') \in C[\![\textbf{while } b \textbf{ do } c]\!] = \mu\Gamma$, existuje $k \in \mathbb{N}_0$ takové, že $(\sigma,\sigma') \in \Gamma^k(\emptyset)$. Indukcí ke k dokážeme, že jestliže $(\sigma,\sigma') \in \Gamma^k(\emptyset)$, pak $\langle \textbf{while } b \textbf{ do } c, \sigma\rangle \to \sigma'$.

  $k = 0$. Jelikož $(\sigma,\sigma') \notin \Gamma^0(\emptyset) = \emptyset$, dokazovaná implikace platí.

  Indukční krok: Nechť tedy $(\sigma,\sigma') \in \Gamma^{k+1}(\emptyset)$. Podle definice Γ jsou dvě možnosti:

  - $\mathcal{B}[\![b]\!]\sigma =$ **true** a $(\sigma,\sigma') \in \Gamma^k(\emptyset) \circ C[\![c]\!]$. Podle 2. $\langle b,\sigma\rangle \to$ **true**. Navíc existuje σ'' takové, že $(\sigma,\sigma'') \in C[\![c]\!]$ a $(\sigma'',\sigma') \in \Gamma^k(\emptyset)$. Podle I.P. $\langle c,\sigma\rangle \to \sigma''$ a $\langle \textbf{while } b \textbf{ do } c, \sigma''\rangle \to \sigma'$, tedy $\langle \textbf{while } b \textbf{ do } c, \sigma\rangle \to \sigma'$.
  - $\mathcal{B}[\![b]\!]\sigma =$ **false** a $\sigma = \sigma'$. Pak $\langle b,\sigma\rangle \to$ **false** podle 2., proto $\langle \textbf{while } b \textbf{ do } c, \sigma\rangle \to \sigma$.

Axiomatická sémantika IMP

- Cílem je definovat odvozovací systém pro trojice tvaru

  $$\{A\} \quad c \quad \{B\}$$

  kde $c \in \mathbf{Com}$ a A, B jsou „tvrzení“. $\{A\} \quad c \quad \{B\}$ říká, že pro každý stav $\sigma \in \Sigma$ platí následující: Jestliže $\sigma \models A$, pak pokud c spuštěný ve stavu σ skončí ve stavu σ', platí $\sigma' \models B$.

  - $\{A\} \quad c \quad \{B\}$ je tedy tvrzení o částečné korektnosti programu c; neříká nic o tom, co platí, jestliže c ve stavu σ neskončí.
  - Platí tedy např. $\{\mathbf{true}\} \quad \mathbf{while\ tt\ do\ skip} \quad \{\mathbf{false}\}$

Axiomatická sémantika IMP (2)

- Pro zjednodušení notace zavedeme speciální „stav“ $\perp$:
  - $\Sigma_\perp \quad = \quad \Sigma \cup \{\perp\}$
  - Dále zavedeme funkci $C_\perp : \mathbf{Com} \to (\Sigma \to \Sigma_\perp)$:
    - $C_\perp[\![c]\!]\sigma = C[\![c]\!]\sigma$ pro každé $c \in \mathbf{Com}$ a $\sigma \in \Sigma$, kde $C[\![c]\!]\sigma$ je definováno;
    - $C_\perp[\![c]\!]\sigma = \perp$ pro každé $c \in \mathbf{Com}$ a $\sigma \in \Sigma$, kde $C[\![c]\!]\sigma$ je nedefinováno.
- Význam $\{A\} \quad c \quad \{B\}$ pak lze vyjádřit takto:

$$\{A\} \quad c \quad \{B\} \qquad \Longleftrightarrow \qquad \forall \sigma \in \Sigma : \quad \sigma \models A \quad \Rightarrow \quad C_\perp[\![c]\!]\sigma \models B$$

Syntaxe a sémantika tvrzení o programech

- Buď **IntVar** $= \{i, j, k, \ldots\}$ spočetná množina celočíselných proměnných.
- Rozšířené aritmetické výrazy **Aexpv**

$$a \quad ::= \quad n \mid X \mid i \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 * a_1$$

  kde $n \in$ **Num**, $X \in$ **Var** a $i \in$ **IntVar**.

- Tvrzení o programech („assertions“) **Assn**

$$A \quad ::= \quad \textbf{true} \mid \textbf{false} \mid a_0 = a_1 \mid a_0 \leq a_1 \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid \neg A \mid \forall i.A \mid \exists i.A$$

  kde $a_0, a_1 \in$ **Aexpv** a $i \in$ **IntVar**.

- Interpretace je funkce $I : \textbf{IntVar} \to \mathbb{Z}$. Množinu všech interpretací značíme $\mathcal{I}$.

Syntaxe a sémantika tvrzení o programech (2)

- Definujeme funkci $\mathcal{E} : \textbf{Aexpv} \to (\mathcal{I} \to (\Sigma \to \mathbb{Z}))$
  - $\mathcal{E}[\![n]\!]I\sigma = n$
  - $\mathcal{E}[\![X]\!]I\sigma = \sigma(X)$
  - $\mathcal{E}[\![i]\!]I\sigma = I(i)$
  - $\mathcal{E}[\![a_0 + a_1]\!]I\sigma = \mathcal{E}[\![a_0]\!]I\sigma \; + \; \mathcal{E}[\![a_1]\!]I\sigma$ (podobně pro „$-$“ a „$*$“)

Syntaxe a sémantika tvrzení o programech (3)

- A je splněno ve stavu $\sigma \in \Sigma_\bot$ za interpretace $I \in \mathcal{I}$ (psáno $\sigma \models^I A$)
  - $\bot \models^I A$ pro každé $A \in \mathbf{Assn}$; je-li $\sigma \neq \bot$, uplatníme následující pravidla:
  - $\sigma \models^I \mathbf{true}$
  - $\sigma \models^I a_0 = a_1 \iff \mathcal{E}[\![a_0]\!]I\sigma = \mathcal{E}[\![a_1]\!]I\sigma$
  - $\sigma \models^I a_0 \leq a_1 \iff \mathcal{E}[\![a_0]\!]I\sigma \leq \mathcal{E}[\![a_1]\!]I\sigma$
  - $\sigma \models^I A_0 \wedge A_1 \iff \sigma \models^I A_0 \ \wedge \ \sigma \models^I A_1$
  - $\sigma \models^I A_0 \vee A_1 \iff \sigma \models^I A_0 \ \vee \ \sigma \models^I A_1$
  - $\sigma \models^I \neg A \iff \sigma \not\models^I A$
  - $\sigma \models^I \forall i.A \iff \sigma \models^{I[n/i]} A$ pro každé $n \in \mathbb{Z}$
  - $\sigma \models^I \exists i.A \iff \sigma \models^{I[n/i]} A$ pro nějaké $n \in \mathbb{Z}$
- $A \in \mathbf{Assn}$ je platné, psáno $\models A$, pokud $\sigma \models^I A$ pro každé $\sigma \in \Sigma$ a $I \in \mathcal{I}$.

Syntaxe a sémantika tvrzení o programech (4)

Lema 21

- *Pro každé* $a \in$ **Aexp**, $\sigma \in \Sigma$ *a* $I \in \mathcal{I}$ *platí* $\mathcal{A}[\![a]\!]\sigma = \mathcal{E}[\![a]\!]I\sigma$.
- *Pro každé* $b \in$ **Bexp** *a* $\sigma \in \Sigma$ *platí*
  - $\mathcal{B}[\![b]\!]\sigma =$ **true** $\iff \sigma \models^I b$
  - $\mathcal{B}[\![b]\!]\sigma =$ **false** $\iff \sigma \not\models^I b$

Důkaz. Strukturální indukcí.

Tvrzení o částečné korektnosti programů

- Tvrzení o částečné korektnosti programu $c \in \mathbf{Com}$ je trojice tvaru

  $$\{A\}\ c\ \{B\}$$

  kde $A, B \in \mathbf{Assn}$.

- $\sigma \models^I \{A\}\ c\ \{B\} \iff (\sigma \models^I A \implies C_\perp[\![c]\!]\sigma \models^I B)$
- $\models \{A\}\ c\ \{B\} \iff \forall \sigma \in \Sigma\ \ \forall I \in \mathcal{I} :\ \sigma \models^I \{A\}\ c\ \{B\}$
- Tvrzení $\{A\}\ c\ \{B\}$ pro které platí $\models \{A\}\ c\ \{B\}$ nazýváme platné.
- Příklady:
  - $\models \{X \geq 1\}\ Y := X + 2\ \{Y \geq 3\}$
  - $\models \{i{=}X \wedge j{=}Y\}\ Z := X + Y + 5\ \{Z \leq i + j + 100\}$
  - $\models \{X{\geq}7\}$ **while** $X{>}3$ **do** $X := X - 1\ \{X{\geq}1\}$

Hoareův odvozovací systém

- Axiom pro **skip**: $\{A\}\ \mathbf{skip}\ \{A\}$
- Axiom pro přiřazení: $\{B[a/X]\}\ X := a\ \{B\}$
- Pravidlo pro sekvenční kompozici:

$$\frac{\{A\}\ c_0\ \{C\} \qquad \{C\}\ c_1\ \{B\}}{\{A\}\ c_0; c_1\ \{B\}}$$

- Pravidlo pro větvení:

$$\frac{\{A \wedge b\}\ c_0\ \{B\} \qquad \{A \wedge \neg b\}\ c_1\ \{B\}}{\{A\}\ \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1\ \{B\}}$$

- Pravidlo pro cyklus:

$$\frac{\{A \wedge b\}\ c\ \{A\}}{\{A\}\ \mathbf{while}\ b\ \mathbf{do}\ c\ \{A \wedge \neg b\}}$$

- Pravidlo důsledku:

$$\frac{\models (A \Rightarrow A') \qquad \{A'\}\ c\ \{B'\} \qquad \models (B' \Rightarrow B)}{\{A\}\ c\ \{B\}}$$

Hoareův odvozovací systém (2)

- Tvrzení $\{A\}\ c\ \{B\}$ je dokazatelné, psáno $\vdash \{A\}\ c\ \{B\}$, je-li dokazatelné v Hoarově odvozovacím systému.
- Tvrzení A pro které platí $\models \{A \wedge b\}\ c\ \{A\}$ se nazývá invariant cyklu **while** b **do** c.

Příklad: $\vdash \{X{=}5\}$ **if** $X = 5$ **then** $Y := X - 2$ **else** $X := Y + 5$ $\{X{=}5\}$

- $$\frac{\models (X{=}5 \wedge X{=}5) \Rightarrow X{=}5 \qquad \{X{=}5\}\quad Y := X - 2\quad \{X{=}5\} \qquad \models X{=}5 \Rightarrow X{=}5}{\{X{=}5 \wedge X{=}5\}\quad Y := X - 2\quad \{X{=}5\}}$$
- $$\frac{\models (X{=}5 \wedge \neg X{=}5) \Rightarrow Y + 5 = 5 \qquad \{Y + 5 = 5\}\quad X := Y + 5\quad \{X = 5\} \qquad \models X{=}5 \Rightarrow X{=}5}{\{X{=}5 \wedge \neg X{=}5\}\quad X := Y + 5\quad \{X{=}5\}}$$
- $$\frac{\{X{=}5 \wedge X{=}5\}\quad Y := X - 2\quad \{X{=}5\} \qquad \{X{=}5 \wedge \neg X{=}5\}\quad X := Y + 5\quad \{X{=}5\}}{\{X{=}5\}\quad \textbf{if } X = 5 \textbf{ then } Y := X - 2 \textbf{ else } X := Y + 5\quad \{X{=}5\}}$$

Hoareův odvozovací systém (3)

Příklad:

$\vdash \{X{+}Y{=}i \wedge Z{=}j\}$ **while** $X{<}0$ **do** $(X := X{+}1;\ Y := Y{-}1);\ Z := Z{-}(X{+}Y)\ \{Z = j{-}i\}$

- $$\frac{\{Z{-}(X{+}1{+}Y{-}1) = j{-}i\} \quad X := X{+}1 \quad \{Z{-}(X{+}Y{-}1) = j{-}i\} \qquad \{Z{-}(X{+}Y{-}1) = j{-}i\} \quad Y := Y{-}1 \quad \{Z{-}(X{+}Y) = j{-}i\}}{\alpha \equiv \quad \{Z{-}(X{+}1{+}Y{-}1) = j{-}i\} \quad X := X{+}1;\ Y := Y{-}1 \quad \{Z{-}(X{+}Y) = j{-}i\}}$$

- $$\frac{\models (Z{-}(X{+}Y) = j{-}i \wedge X{<}0) \Rightarrow (Z{-}(X{+}1{+}Y{-}1) = j{-}i) \quad \alpha \quad \models (Z{-}(X{+}Y) = j{-}i) \Rightarrow (Z{-}(X{+}Y) = j{-}i)}{\beta \equiv \quad \{Z{-}(X{+}Y) = j{-}i\} \quad \textbf{while } X{<}0 \textbf{ do } X := X{+}1;\ Y := Y{-}1 \quad \{Z{-}(X{+}Y) = j{-}i \wedge \neg X{<}0\}}$$

- $$\frac{\models (X{+}Y{=}i \wedge Z{=}j) \Rightarrow Z{-}(X{+}Y) = j{-}i \quad \beta \quad \models (Z{-}(X{+}Y) = j{-}i \wedge \neg X{<}0) \Rightarrow Z{-}(X{+}Y) = j{-}i}{\gamma \equiv \quad \{X{+}Y = i \wedge Z{=}j\} \quad \textbf{while } X{<}0 \textbf{ do } X := X{+}1;\ Y := Y{-}1 \quad \{Z{-}(X{+}Y) = j{-}i\}}$$

- $$\frac{\gamma \qquad \{Z{-}(X{+}Y) = j{-}i\} \quad Z := Z{-}(X{+}Y) \quad \{Z = j{-}i\}}{\{X{+}Y{=}i \wedge Z{=}j\} \quad \textbf{while } X{<}0 \textbf{ do } (X := X{+}1;\ Y := Y{-}1);\ Z := Z{-}(X{+}Y) \quad \{Z = j{-}i\}}$$

Cvičení

Dokažte následující tvrzení o částečné korektnosti:

- $\{X{=}3\}$ **while** $X{=}3$ **do** $Y := X{+}1$ $\{2{=}3\}$
- $\{X{\geq}7\}$ **while** $X{>}3$ **do** $X := X - 1$ $\{X{\geq}1\}$
- $\{X{=}4 \wedge Y{=}2\}$ **while** $X{>}3$ **do** $X := X - 1; Y := Y + 1$ $\{Y{\geq}0\}$
- $\{X{=}4 \wedge Y{=}2\}$ **while** $X{>}3$ **do** $X := X - 1; Y := Y + 1$ $\{Y{\geq}3\}$

Korektnost Hoareova odvozovacího systému

Lema 22

Nechť $I \in \mathcal{I}$, $X \in$ **Var**, $a_0, a_1 \in$ **Aexpv**, $a \in$ **Aexp** *a* $B \in$ **Assn**. *Pak pro každé* $\sigma \in \Sigma$ *platí:*

- $\mathcal{E}[\![a_0[a_1/X]]\!]I\sigma = \mathcal{E}[\![a_0]\!]I\sigma[\mathcal{E}[\![a_1]\!]I\sigma/X]$
- $\sigma \models^I B[a/X] \quad \Longleftrightarrow \quad \sigma[\mathcal{A}[\![a]\!]\sigma/X] \models^I B$

Důkaz. Indukcí ke struktuře a_0, resp. B.

Korektnost Hoareova odvozovacího systému (2)

Věta 23 (o korektnosti)

Jestliže $\vdash \{A\}\ c\ \{B\}$, *pak* $\models \{A\}\ c\ \{B\}$.

Důkaz. Indukcí k výšce odvozovacího stromu pro $\{A\}\ c\ \{B\}$.

Uvážíme, jaké pravidlo bylo použito pro odvození kořene:

- Axiom pro **skip**. Pak je kořen tvaru $\{A\}$ **skip** $\{A\}$. Toto tvrzení je zjevně platné.
- Axiom pro $X := a$. Pak je kořen tvaru $\{B[a/X]\}\ X := a\ \{B\}$. Nechť $I \in \mathcal{I}$ a $\sigma \in \Sigma$. Podle lematu 22 platí $\sigma \models^I B[a/X] \iff \sigma[\mathcal{A}[\![a]\!]\sigma/X] \models^I B$. Jelikož $\sigma[\mathcal{A}[\![a]\!]\sigma/X] = C[\![X := a]\!]\sigma$, dostáváme

  $$\sigma \models^I B[a/X] \quad \Rightarrow \quad C[\![X := a]\!]\sigma \models^I B,$$

  tedy $\models \{B[a/X]\}\ X := a\ \{B\}$.

Korektnost Hoareova odvozovacího systému (3)

- Pravidlo pro sekvenční kompozici. Pak je kořen tvaru $\{A\}\ c_0; c_1\ \{B\}$ a má následníky $\{A\}\ c_0\ \{C\}$ a $\{C\}\ c_1\ \{B\}$, což jsou podle I.P. platná tvrzení. Nechť $I \in \mathcal{I}$ a $\sigma \in \Sigma$. Předpokládejme, že $\sigma \models^I A$. Platí $C_\perp[\![c_0]\!]\sigma \models^I C$ (neboť $\models \{A\}\ c_0\ \{C\}$) a $C_\perp[\![c_1]\!](C_\perp[\![c_0]\!]\sigma) \models^I B$, protože $\models \{C\}\ c_1\ \{B\}$ a $C_\perp[\![c_0]\!]\sigma \models^I C$. Tedy $\models \{A\}\ c_0; c_1\ \{B\}$.

- Pravidlo pro větvení. Pak kořen $\{A\}$ **if** b **then** c_0 **else** c_1 $\{B\}$ má následníky $\{A \wedge b\}\ c_0\ \{B\}$ a $\{A \wedge \neg b\}\ c_1\ \{B\}$, což jsou podle I.P. platná tvrzení. Nechť $I \in \mathcal{I}$ a $\sigma \in \Sigma$. Předpokládejme, že $\sigma \models^I A$. Dále buď $\sigma \models^I b$, nebo $\sigma \models^I \neg b$. V prvém případě $\sigma \models^I A \wedge b$, tedy $C_\perp[\![c_0]\!]\sigma \models^I B$, neboť $\models \{A \wedge b\}\ c_0\ \{B\}$. V druhém případě $\sigma \models^I A \wedge \neg b$, tedy $C_\perp[\![c_1]\!]\sigma \models^I B$. Celkem $\models \{A\}$ **if** b **then** c_0 **else** c_1 $\{B\}$ (užitím lematu 21).

Korektnost Hoareova odvozovacího systému (4)

- Pravidlo pro cyklus. Pak kořen $\{A\}$ **while** b **do** c $\{A \wedge \neg b\}$ má následníka $\{A \wedge b\}$ c $\{A\}$, který je podle I.P. platným tvrzením. Nechť $I \in \mathcal{I}$ a $\sigma \in \Sigma$. Potřebujeme ukázat, že

$$\sigma \models^I A \quad \Rightarrow \quad C_\perp[\![\textbf{while } b \textbf{ do } c]\!]\sigma \models^I A \wedge \neg b$$

Mějme tedy σ a I takové, že $\sigma \models^I A$. Označme $\sigma' = C_\perp[\![\textbf{while } b \textbf{ do } c]\!]\sigma$. Pokud $\sigma' = \perp$, jsme hotovi. Jinak $\sigma' = C[\![\textbf{while } b \textbf{ do } c]\!]\sigma$ (podle definice $C_\perp$), proto $(\sigma, \sigma') \in \Gamma^j(\emptyset)$ pro nějaké $j \in \mathbb{N}_0$ (jelikož $C[\![\textbf{while } b \textbf{ do } c]\!] = \bigcup_{i=0}^{\infty} \Gamma^i(\emptyset)$). K tomu, že $\sigma' \models^I A \wedge \neg b$, stačí ukázat, že pro každé $j \in \mathbb{N}_0$ platí

$$D(j) \quad \equiv \quad \forall \sigma, \sigma' \in \Sigma, \forall I \in \mathcal{I} : \quad ((\sigma, \sigma') \in \Gamma^j(\emptyset) \wedge \sigma \models^I A) \quad \Rightarrow \quad \sigma' \models^I A \wedge \neg b$$

Indukcí vzhledem k j.

Korektnost Hoareova odvozovacího systému (5)

- $j = 0$. Jelikož $\Gamma^0(\emptyset) = \emptyset$, neplatí antecedent dokazované implikace.
- Indukční krok: Předpokládejme, že $D(j)$ platí. Dokážeme, že platí $D(j+1)$. Nechť tedy $(\sigma, \sigma') \in \Gamma^{j+1}(\emptyset)$ a $\sigma \models^I A$. Podle definice Γ jsou dvě možnosti:
  - $\mathcal{B}[\![b]\!]\sigma =$ **true** a $(\sigma, \sigma') \in \Gamma^j(\emptyset) \circ C[\![c]\!]$. Pak $\sigma \models^I b$ (užitím lematu 21), tedy $\sigma \models^I A \wedge b$. Dále existuje σ'' takové, že $(\sigma, \sigma'') \in C[\![c]\!]$ a $(\sigma'', \sigma') \in \Gamma^j(\emptyset)$. Jelikož $\models \{A \wedge b\}\ c\ \{A\}$, platí $\sigma'' \models^I A$. Nyní podle $D(j)$ dostáváme $\sigma' \models^I A \wedge \neg b$, tedy $D(j+1)$ platí.
  - $\mathcal{B}[\![b]\!]\sigma =$ **false** a $\sigma' = \sigma$. Platí $\sigma \models^I \neg b$ a tedy $\sigma \models^I A \wedge \neg b$, což bylo dokázat.

- Pravidlo důsledku. Pak kořen $\{A\}\ c\ \{B\}$ má následníky $\models (A \Rightarrow A')$, $\{A'\}\ c\ \{B'\}$ a $\models (B' \Rightarrow B)$. Podle I.P. je $\{A'\}\ c\ \{B'\}$ platné tvrzení. Nechť $I \in \mathcal{I}$ a $\sigma \in \Sigma$. Jestliže $\sigma \models^I A$, platí také $\sigma \models^I A'$, proto $C_\perp[\![c]\!]\sigma \models^I B'$ a tudíž i $C_\perp[\![c]\!]\sigma \models^I B$.

Nejslabší vstupní podmínka

- Nechť $c \in \mathbf{Com}$, $B \in \mathbf{Assn}$ a $I \in \mathcal{I}$. Nejslabší vstupní podmínka pro B vzhledem k c a I, označovaná $wp^I[\![c, B]\!]$, je definovaná takto:

$$wp^I[\![c, B]\!] \quad = \quad \{\sigma \in \Sigma \mid C_\perp[\![c]\!]\sigma \models^I B\}.$$

- Zavedeme značení $A^I = \{\sigma \in \Sigma \mid \sigma \models^I A\}$.
- Dané $A \in \mathbf{Assn}$ vyjadřuje nejslabší vstupní podmínku pro B a c, pokud pro každé $I \in \mathcal{I}$ platí $A^I = wp^I[\![c, B]\!]$.
- Platí $\models^I \{A\}\ c\ \{B\} \iff A^I \subseteq wp^I[\![c, B]\!]$.
- Předpokládejme, že $A_0 \in \mathbf{Assn}$ vyjadřuje nejslabší vstupní podmínku pro B a c. Pak výše uvedenou ekvivalenci lze přepsat na

$$\models^I \{A\}\ c\ \{B\} \iff A^I \subseteq A_0^I \iff \models^I (A \Rightarrow A_0)$$

  což platí pro libovolné I, tedy $\models \{A\}\ c\ \{B\} \iff \models (A \Rightarrow A_0)$ (odtud přívlastek „nejslabší“).

Nejslabší vstupní podmínka (2)

- Úplnost Hoareova odvozovacího systému dokážeme ve dvou krocích:
  - Pro každé $c \in \mathbf{Com}$ a $B \in \mathbf{Assn}$ existuje $A[\![c, B]\!] \in \mathbf{Assn}$, které vyjadřuje nejslabší vstupní podmínku pro B a c.
  - Pro každé $c \in \mathbf{Com}$ a $B \in \mathbf{Assn}$ platí $\vdash \{A[\![c, B]\!]\}\ c\ \{B\}$.

  Pokud $\models \{A\}\ c\ \{B\}$, platí $\models (A \Rightarrow A[\![c, B]\!])$, a jelikož $\vdash \{A[\![c, B]\!]\}\ c\ \{B\}$, dostáváme $\vdash \{A\}\ c\ \{B\}$ užitím pravidla důsledku.

Nejslabší vstupní podmínka. Příklady.

- $wp^I[\![\textbf{skip}, \textbf{false}]\!] = \emptyset$. Vyjádřitelné jako **false**.
- Obecně $wp^I[\![\textbf{skip}, B]\!] = B^I$.
- $wp^I[\![\textbf{while true do skip}, \textbf{true}]\!] = \Sigma$. Vyjádřitelné jako **true**.
- $wp^I[\![\textbf{while true do skip}, \textbf{false}]\!] = \Sigma$. Vyjádřitelné jako **true**.
- $wp^I[\![X := 2; Y := 4, X = 3]\!] = \emptyset$.
- $wp^I[\![X := X + 1; Y := Y - 1, X = 3 \wedge Y = 6]\!] = \{\sigma \in \Sigma \mid \sigma(X) = 2 \wedge \sigma(Y) = 7\}$. Vyjádřitelné jako $X = 2 \wedge Y = 7$.
- $wp^I[\![X := Y, i = X]\!] = \{\sigma \in \Sigma \mid \sigma(Y) = I(i)\}$. Vyjádřitelné jako $Y = i$.

Gödelův predikát β

- Definujeme 4-ární predikát β na nezáporných celých číslech předpisem

$$\beta(a,b,i,x) \iff x = a \mod (1 + b(1+i))$$

- Buď $\mathcal{S}$ nekonečná posloupnost nezáporných celých čísel, $a, b \in \mathbb{N}_0$. Řekneme, že $\mathcal{S}$ splňuje β (pro dané a a b), jestliže pro každé $i \in \mathbb{N}_0$ platí $\beta(a,b,i,\mathcal{S}(i))$.
- Pro každé $a, b \in \mathbb{N}_0$ existuje jediná posloupnost splňující β; tou je posloupnost $\mathcal{S}_{a,b}$ daná předpisem $\mathcal{S}_{a,b}(i) = a \mod (1 + b(1+i))$.
- Predikát β je vyjádřitelný v **Assn**, neboť $x = a \mod b$ lze napsat jako

$$a \geq 0 \quad \wedge \quad b \geq 0 \quad \wedge$$
$$\exists k : \quad (k \geq 0 \quad \wedge \quad k * b \leq a \quad \wedge \quad (k+1) * b > a \quad \wedge \quad x = a - (k * b))$$

Gödelův predikát β (2)

Věta 24

Pro každou konečnou posloupnost $n_0, \cdots, n_k$ nezáporných celých čísel existují $n, m \in \mathbb{N}_0$ taková, že $n_j = \mathcal{S}_{n,m}(j)$ pro každé $0 \leq j \leq k$. To znamená, že pro každé $0 \leq j \leq k$ platí $\beta(n, m, j, x) \iff x = n_j$.

Důkaz. (osnova)

- Nechť $m = (\max\{k, n_0, \cdots, n_k\})!$. Čísla

  $$p_i = 1 + m(1 + i), \quad 0 \leq i \leq k$$

  jsou navzájem nesoudělná a $n_i < p_i$ pro každé $0 \leq i \leq k$.

- Dále pro každé $0 \leq i \leq k$ definujeme $c_i = p_0 \cdot \cdots \cdot p_k / p_i$. Nyní pro každé $0 \leq i \leq k$ existuje přesně jedno d_i, $0 \leq d_i \leq p_i$, takové, že $(c_i \cdot d_i) \mod p_i = 1$

- Definujeme

  $$n = \sum_{i=0}^{k} c_i \cdot d_i \cdot n_i.$$

  Pro každé $0 \leq i \leq k$ platí $n_i = n \mod p_i$, což je tvrzení věty.

Rozšířený predikát $\beta^{\pm}$

- Cílem je vytvořit prostředek pro kódování konečných posloupností hodnot proměnných jazyka **IMP**, tj. konečných posloupností celých čísel.
- Celá čísla lze seřadit do posloupnosti $0, 0, 1, -1, 2, -2, 3, -3, \cdots$. Každé celé číslo je pak „kódováno“ pozicí v této posloupnosti, což je nezáporné celé číslo (0 má dokonce dva kódy, 0 a 1).
- Definujeme binární predikát $F(x, y)$ („x je kódem y“) na celých číslech předpisem

$$F(x,y) \quad \Longleftrightarrow \quad x \geq 0 \wedge \forall z : \ (x = 2{*}z \ \Rightarrow \ y = z) \wedge (x = 2{*}z{+}1 \ \Rightarrow \ y = -z)$$

- Nyní lze definovat 4-ární predikát $\beta^{\pm}$ na celých číslech (vyjádřitelný v **Assn**) předpisem

$$\beta^{\pm}(n,m,j,y) \quad \Longleftrightarrow \quad \exists x : \quad \beta(n,m,j,x) \quad \wedge \quad F(x,y)$$

- Analogicky jako pro β definujeme posloupnost splňující $\beta^{\pm}$ a posloupnost $\mathcal{S}^{\pm}_{n,m}$, která je jedinou posloupností splňující $\beta^{\pm}$ (pro dané $n, m \in \mathbb{N}_0$).

Rozšířený predikát $\beta^{\pm}$ (2)

Věta 25

Pro každou konečnou posloupnost $n_0, \cdots, n_k$ celých čísel existují $n, m \in \mathbb{N}_0$ taková, že $n_j = \mathcal{S}^{\pm}_{n,m}(j)$ pro každé $0 \leq j \leq k$. To znamená, že pro každé $0 \leq j \leq k$ platí $\beta^{\pm}(n, m, j, x) \iff x = n_j$.

Příklad užití predikátu $\beta^{\pm}$

- Lze v **Assn** vyjádřit $X \geq 0 \ \wedge \ Y = X!$?
- „Zakódujeme“ posloupnost $1, 1, 2, 6, 24, \cdots, X!$
  - $Y = X!$
  - $\exists \mathcal{S} : \ \mathcal{S}(0) = 1 \quad \wedge \quad \forall \ell : \ 1 \leq \ell \leq X \ \Rightarrow \ (\mathcal{S}(\ell) = \mathcal{S}(\ell - 1) * \ell) \quad \wedge \quad Y = \mathcal{S}(X)$
  - $\exists \mathcal{S} : \ \mathcal{S}(0) = 1$
    $\wedge \quad \forall \ell : \ 1 \leq \ell \leq X \ \Rightarrow \ (\forall u, v : \ u = \mathcal{S}(\ell) \ \wedge \ v = \mathcal{S}(\ell - 1) \ \Rightarrow \ u = v * \ell)$
    $\wedge \quad Y = \mathcal{S}(X)$
  - $\exists n \, \exists m : \ \beta^{\pm}(n, m, 0, 1)$
    $\wedge \quad \forall \ell : 1 \leq \ell \leq X \quad \Rightarrow \quad (\forall u, v : \beta^{\pm}(n, m, \ell, u) \wedge \beta^{\pm}(n, m, \ell - 1, v) \quad \Rightarrow \ u = v * \ell)$
    $\wedge \quad \beta^{\pm}(n, m, X, Y)$

Vyjádřitelnost nejslabší vstupní podmínky v Assn

Věta 26

*Pro každé $c \in$ **Com** a $B \in$ **Assn** existuje $A[\![c, B]\!] \in$ **Assn** takové, že pro každé $I \in \mathcal{I}$ platí $A[\![c, B]\!]^I = wp^I[\![c, B]\!]$.*

Důkaz. Je dobré si znovu uvědomit, že

$$A[\![c, B]\!]^I = wp^I[\![c, B]\!] \iff \forall \sigma \in \Sigma : (\sigma \models^I A[\![c, B]\!] \iff C_\perp[\![c]\!]\sigma \models^I B).$$

Důkaz je veden indukcí ke struktuře c.

- $c \equiv$ **skip**. Stačí položit $A[\![\textbf{skip}, B]\!] = B$. Pro každé $I \in \mathcal{I}$ a $\sigma \in \Sigma$ platí

$$\begin{aligned} & \sigma \in wp^I[\![\textbf{skip}, B]\!] \\ \iff & C_\perp[\![\textbf{skip}]\!]\sigma \models^I B \\ \iff & \sigma \models^I B \\ \iff & \sigma \models^I A[\![\textbf{skip}, B]\!]. \end{aligned}$$

Vyjádřitelnost nejslabší vstupní podmínky v Assn (2)

- $c \equiv X := a$. Definujeme $A[\![X := a, B]\!] = B[a/X]$. Pak pro každé $I \in \mathcal{I}$ a $\sigma \in \Sigma$ platí

  $\sigma \in wp^I[\![X := a, B]\!]$

  $\iff \sigma[\mathcal{A}[\![a]\!]\sigma/X] \models^I B$

  $\iff \sigma \models^I B[a/X]$ (užitím lematu 22)

  $\iff \sigma \models^I A[\![X := a, B]\!]$.

- $c \equiv c_0; c_1$. Definujeme $A[\![c_0; c_1, B]\!] = A[\![c_0, A[\![c_1, B]\!]]\!]$. Pro každé $I \in \mathcal{I}$ a $\sigma \in \Sigma$ platí

  $\sigma \in wp^I[\![c_0; c_1, B]\!]$

  $\iff C_\perp[\![c_0; c_1]\!]\sigma \models^I B$

  $\iff C[\![c_0]\!]\sigma \models^I A[\![c_1, B]\!]$ (podle I.P.)

  $\iff \sigma \models^I A[\![c_0, A[\![c_1, B]\!]]\!]$ (podle I.P.)

  $\iff \sigma \models^I A[\![c_0; c_1, B]\!]$.

Vyjádřitelnost nejslabší vstupní podmínky v Assn (3)

- $c \equiv \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1$. Definujeme $A[\![\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1, B]\!] = (b \wedge A[\![c_0, B]\!]) \vee (\neg b \wedge A[\![c_1, B]\!])$. Pak pro každé $I \in \mathcal{I}$ a $\sigma \in \Sigma$ platí

$$
\begin{aligned}
& \sigma \in wp^I[\![c, B]\!] \\
\iff\ & C_\perp[\![c]\!]\sigma \models^I B \\
\iff\ & (\mathcal{B}[\![b]\!] = \textbf{true} \wedge C_\perp[\![c_0]\!]\sigma \models^I B) \vee (\mathcal{B}[\![b]\!] = \textbf{false} \wedge C_\perp[\![c_1]\!]\sigma \models^I B) \\
\iff\ & (\sigma \models^I b \wedge \sigma \models^I A[\![c_0, B]\!]) \vee (\sigma \models^I \neg b \wedge \sigma \models^I A[\![c_1, B]\!]) \text{ podle I.P.} \\
\iff\ & \sigma \models^I (b \wedge A[\![c_0, B]\!]) \vee (\neg b \wedge A[\![c_1, B]\!]) \iff \sigma \models^I A[\![c, B]\!].
\end{aligned}
$$

Vyjádřitelnost nejslabší vstupní podmínky v Assn (4)

- $c \equiv \textbf{while } b \textbf{ do } c_0$. Platí $\sigma \in wp^I[\![\textbf{while } b \textbf{ do } c_0, B]\!] \iff$

$$\forall k\ \forall \sigma_0, \cdots, \sigma_k \in \Sigma\ :$$
$$\Big(\sigma = \sigma_0\ \wedge\ \forall i (0 \le i < k)\ : (\sigma_i \models^I b\ \wedge\ C[\![c_0]\!]\sigma_i = \sigma_{i+1})\Big)\ \Rightarrow\ \sigma_k \models^I b \vee B \tag{1}$$

Stačí tedy výše uvedené tvrzení „přeložit“ do **Assn**. Nechť $X_0, \ldots, X_{\ell-1}$ jsou všechny proměnné, které se vyskytují v b, c_0 a B. Symbolem $\vec{s}$ označíme ℓ-tici celočíselných proměnných $s(0), \ldots, s(\ell-1)$. Nyní lze (1) přepsat na

$$\forall k\ \forall \vec{s}_0, \cdots, \vec{s}_k\ :$$
$$\Big(\sigma \models^I \vec{X} = \vec{s}_0\ \wedge$$
$$\forall i (0 \le i < k)\ : (\models^I b[\vec{s}_i/\vec{X}] \wedge\ \models^I (A[\![c_0, \vec{X} = \vec{s}_{i+1}]\!]\ \wedge\ \neg A[\![c_0, \textbf{false}]\!])[\vec{s}_i/\vec{X}])\Big)$$
$$\Rightarrow\ \models^I (b \vee B)[\vec{s}_k/\vec{X}]$$

Tuto formuli lze dále přepsat na výraz **Assn** pomocí predikátu $\beta^{\pm}$ následujícím způsobem:

Vyjádřitelnost nejslabší vstupní podmínky v Assn (5)

- Řetěz $\forall \vec{s}_0, \cdots, \vec{s}_k$ přepíšeme na $\forall n, m$.
- $\sigma \models^I \vec{X} = \vec{s}_0$ přepíšeme na $\bigwedge_{j=0}^{\ell-1} \beta^{\pm}(n, m, j, X_j)$.
- $\models^I b[\vec{s}_i/\vec{X}]$ přepíšeme na $\forall u_0, \ldots, u_{\ell-1} : \bigwedge_{j=0}^{\ell-1} \beta^{\pm}(n, m, j, u_j) \Rightarrow b[\vec{u}/\vec{X}]$.
- $\models^I (A[\![c_0, \vec{X} = \vec{s}_{i+1}]\!] \wedge \neg A[\![c_0, \textbf{false}]\!])[\vec{s}_i/\vec{X}]$ přepíšeme na $\forall u_0, \ldots, u_{\ell-1}, v_0, \ldots, v_{\ell-1} : \Big(\bigwedge_{j=0}^{\ell-1} \beta^{\pm}(n, m, \ell{*}i + j, u_j) \wedge \beta^{\pm}(n, m, \ell{*}(i{+}1) + j, v_j)\Big) \Rightarrow (A[\![c_0, \vec{X} = \vec{s}_{i+1}]\!][\vec{v}/\vec{s}_{i+1}] \wedge \neg A[\![c_0, \textbf{false}]\!])[\vec{u}/\vec{X}]$.
- $\models^I (b \vee B)[\vec{s}_k/\vec{X}]$ přepíšeme na $\forall u_0, \ldots, u_{\ell-1} : \bigwedge_{j=0}^{\ell-1} \beta^{\pm}(n, m, (\ell{*}k) + j, u_j) \Rightarrow (b \vee B)[\vec{u}/\vec{X}]$.

Úplnost Hoareova odvozovacího systému

Věta 27

Nechť $c \in \mathbf{Com}$ *a* $B \in \mathbf{Assn}$. *Pak* $\vdash \{A[\![c, B]\!]\}\, c\, \{B\}$.

Důkaz. Indukcí ke struktuře c (využijeme poznatků z důkazu věty 26).

- $c \equiv \mathbf{skip}$. Pak $A[\![\mathbf{skip}, B]\!] \equiv B$, tedy $\vdash \{A[\![\mathbf{skip}, B]\!]\}\, \mathbf{skip}\, \{B\}$ užitím axiomu pro **skip**.
- $c \equiv X := a$. Pak $A[\![X := a, B]\!] \equiv B[a/X]$, tedy $\vdash \{A[\![X := a, B]\!]\}\, X := a\, \{B\}$ užitím axiomu pro přiřazení.
- $c \equiv c_0; c_1$. Pak $A[\![c_0; c_1, B]\!] \equiv A[\![c_0, A[\![c_1, B]\!]]\!]$. Podle I.P. $\vdash \{A[\![c_1, B]\!]\}\, c_1\, \{B\}$ a $\vdash \{A[\![c_0, A[\![c_1, B]\!]]\!]\}\, c_0\, \{A[\![c_1, B]\!]\}$. Podle pravidla pro sekvenční kompozici $\vdash \{A[\![c_0, A[\![c_1, B]\!]]\!]\}\, c_0; c_1\, \{B\}$.

Úplnost Hoareova odvozovacího systému (2)

- $c \equiv \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1$. Pak $A[\![c, B]\!] \equiv (b \wedge A[\![c_0, B]\!]) \vee (\neg b \wedge A[\![c_1, B]\!])$. Podle I.P. $\vdash \{A[\![c_0, B]\!]\}\, c_0\, \{B\}$ a $\vdash \{A[\![c_1, B]\!]\}\, c_1\, \{B\}$. Jelikož $\models (A[\![c, B]\!] \wedge b) \Rightarrow A[\![c_0, B]\!]$, užitím pravidla důsledku dostáváme $\vdash \{A[\![c, B]\!] \wedge b\}\, c_0\, \{B\}$. Podobně $\vdash \{A[\![c, B]\!] \wedge \neg b\}\, c_1\, \{B\}$, můžeme tedy použít pravidlo pro větvení čímž obdržíme $\vdash \{A[\![c, B]\!]\}\, c\, \{B\}$.

Úplnost Hoareova odvozovacího systému (3)

- $c \equiv$ **while** b **do** c_0. Dokážeme, že

  a) $\models \{A[\![c, B]\!] \wedge b\}\, c_0\, \{A[\![c, B]\!]\}$,

  b) $\models (A[\![c, B]\!] \wedge \neg b) \Rightarrow B$.

  Podle I.P. $\vdash \{A[\![c_0, A[\![c, B]\!]]\!]\}\, c_0\, \{A[\![c, B]\!]\}$. Podle a) pak $\vdash \{A[\![c, B]\!] \wedge b\}\, c_0\, \{A[\![c, B]\!]\}$ užitím pravidla důsledku. Dále podle pravidla pro **while** dostáváme $\vdash \{A[\![c, B]\!]\}\, c\, \{A[\![c, B]\!] \wedge \neg b\}$ a pomocí b) konečně $\vdash \{A[\![c, B]\!]\}\, c\, \{B\}$.

  ad a) Nechť $\sigma \in \Sigma$ a $I \in \mathcal{I}$. Jestliže $\sigma \models^I A[\![c, B]\!] \wedge b$, platí $\sigma \models^I A[\![c, B]\!]$ a $\sigma \models^I b$, tj. $C_\perp[\![c]\!]\sigma \models^I B$ a $\sigma \models^I b$. Denotační sémantika byla definována tak, že platí:

  $$C_\perp[\![c]\!] \quad = \quad C_\perp[\![\textbf{if } b \textbf{ then } c_0; c \textbf{ else skip}]\!]$$

  To znamená, že $C_\perp[\![c_0; c]\!]\sigma \models^I B$, tedy $C_\perp[\![c]\!](C_\perp[\![c_0]\!]\sigma) \models^I B$. Proto $C_\perp[\![c_0]\!]\sigma \models^I A[\![c, B]\!]$, tedy $\models \{A[\![c, B]\!] \wedge b\}\, c_0\, \{A[\![c, B]\!]\}$.

  ad b) Nechť $\sigma \in \Sigma$ a $I \in \mathcal{I}$. Jestliže $\sigma \models^I A[\![c, B]\!] \wedge \neg b$, pak $\sigma \models^I A[\![c, B]\!]$ a $\sigma \models^I \neg b$. Jelikož

  $$C_\perp[\![c]\!] \quad = \quad C_\perp[\![\textbf{if } b \textbf{ then } c_0; c \textbf{ else skip}]\!],$$

  dostáváme $C_\perp[\![c]\!]\sigma = \sigma$, proto $\sigma \models^I B$. Tedy $\models (A[\![c, B]\!] \wedge \neg b) \Rightarrow B$.

Úplnost Hoareova odvozovacího systému (4)

Věta 28 (o úplnosti)

Jestliže $\models \{A\}\ c\ \{B\}$, *pak* $\vdash \{A\}\ c\ \{B\}$.

Důkaz. Předpokládejme, že $\models \{A\}\ c\ \{B\}$. Podle věty 26 existuje $A[\![c,B]\!] \in \mathbf{Assn}$, které vyjadřuje nejslabší vstupní podmínku pro c a B. Platí tedy $\models (A \Rightarrow A[\![c,B]\!])$. Podle věty 27 platí $\vdash \{A[\![c,B]\!]\}\ c\ \{B\}$ a tedy $\vdash \{A\}\ c\ \{B\}$ užitím pravidla důsledku.

Ekvivalence axiomatické a denotační sémantiky

- Axiomatická sémantika přirozeným způsobem definuje sémantickou ekvivalenci $\simeq$ na příkazech:

$$c_0 \simeq c_1 \iff \forall A, B \in Assn : \quad (\models \{A\}\, c_0\, \{B\} \iff \models \{A\}\, c_1\, \{B\})$$

Věta 29

Pro každé $c_0, c_1 \in$ **Com** *platí:* $C[\![c_0]\!] = C[\![c_1]\!] \iff c_0 \simeq c_1$

Důkaz.

„$\Rightarrow$“ Buďte $\sigma \in \Sigma$ a $I \in \mathcal{I}$ takové, že $\sigma \models^I A$. Jelikož $C_\perp[\![c_0]\!]\sigma = C_\perp[\![c_1]\!]\sigma$, platí

$$C_\perp[\![c_0]\!]\sigma \models^I B \iff C_\perp[\![c_1]\!]\sigma \models^I B$$

„$\Leftarrow$“ Ukážeme, že pokud $C[\![c_0]\!] \neq C[\![c_1]\!]$, pak existuje $\sigma \in \Sigma$ a $A, B \in$ **Assn** takové, že

$$\models \{A\}\, c_0\, \{B\} \quad \not\iff \quad \models \{A\}\, c_1\, \{B\}$$

Ekvivalence axiomatické a denotační sémantiky (2)

Jestliže $C[\![c_0]\!] \neq C[\![c_1]\!]$, existuje $\sigma \in \Sigma$ takové, že $C_\perp[\![c_0]\!]\sigma \neq C_\perp[\![c_1]\!]\sigma$. Nechť $\mathcal{X}$ je množina všech proměnných, které se vyskytují v c_0 a c_1. Definujeme

$$A \quad \equiv \quad \bigwedge_{Y \in \mathcal{X}} Y = n_Y$$

kde n_Y je hodnota proměnné Y ve stavu σ. Dále označme $\sigma_0 = C_\perp[\![c_0]\!]\sigma$ a $\sigma_1 = C_\perp[\![c_1]\!]\sigma$. Rozlišíme tři možnosti:

- $\sigma_0 \neq \perp \neq \sigma_1$. Jelikož $\sigma_0 \neq \sigma_1$, existuje $X \in \mathcal{X}$ takové, že $\sigma_0(X) \neq \sigma_1(X)$. Definujeme $B \equiv X{=}m$, kde m je hodnota proměnné X ve stavu σ_0. Pak $\models \{A\}\, c_0\, \{B\}$, zatímco $\not\models \{A\}\, c_1\, \{B\}$ (neboť $\sigma \not\models^I \{A\}\, c_1\, \{B\}$ pro libovolné $I \in \mathcal{I}$).
- $\sigma_0 = \perp$. Pak $\sigma_1 \neq \perp$. Proto $\models \{A\}\, c_0\, \{\textbf{false}\}$, zatímco $\not\models \{A\}\, c_1\, \{\textbf{false}\}$.
- $\sigma_1 = \perp$. Podobně.

Operační sémantika paralelních programů

- Syntaxi jazyka **IMP** rozšíříme o paralelní operátor $\|$; paralelní příkazy **PCom** jsou definovány rovnicí

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

- SOS sémantiku II. typu rozšíříme o pravidla

$$\frac{\langle c_0, \sigma\rangle \mapsto \langle c_0', \sigma'\rangle}{\langle c_0 \| c_1, \sigma\rangle \mapsto \langle c_0' \| c_1, \sigma'\rangle} \qquad \frac{\langle c_1, \sigma\rangle \mapsto \langle c_1', \sigma'\rangle}{\langle c_0 \| c_1, \sigma\rangle \mapsto \langle c_0 \| c_1', \sigma'\rangle}$$

- Podle stávajících pravidel např.

$$\langle (X := 1 \| X := 2); X := 3, \sigma\rangle \mapsto \langle (\textbf{skip} \| X := 2); X := 3, \sigma[1/X]\rangle \mapsto \langle (\textbf{skip} \| \textbf{skip}); X := 3, \sigma[2/X]\rangle$$

Z konfigurace $\langle (\textbf{skip} \| \textbf{skip}); X := 3, \sigma[2/X]\rangle$ není možné odvodit žádný přechod, ačkoliv příkaz $X := 3$ je v této konfiguraci proveditelný (podle intuitivního chápání významu „;“ a „$\|$“).

Operační sémantika paralelních programů (2)

- Zavedeme predikát **IsSkip** předpisem

$$\begin{aligned}
\textbf{IsSkip}(\textbf{skip}) &= \textbf{true} \\
\textbf{IsSkip}(X := a) &= \textbf{false} \\
\textbf{IsSkip}(c_0; c_1) &= \textbf{IsSkip}(c_0) \wedge \textbf{IsSkip}(c_1) \\
\textbf{IsSkip}(\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1) &= \textbf{false} \\
\textbf{IsSkip}(\textbf{while } b \textbf{ do } c) &= \textbf{false} \\
\textbf{IsSkip}(c_0 \| c_1) &= \textbf{IsSkip}(c_0) \wedge \textbf{IsSkip}(c_1)
\end{aligned}$$

- Pravidlo $\langle \textbf{skip}; c, \sigma \rangle \mapsto \langle c, \sigma \rangle$ nahradíme pravidlem

$$\frac{}{\langle c_0; c_1, \sigma \rangle \mapsto \langle c_1, \sigma \rangle} \ \textbf{IsSkip}(c_0)$$

- Nyní je již odvoditelný přechod $\langle (\textbf{skip} \| \textbf{skip}); X := 3, \sigma[2/X] \rangle \mapsto \langle X := 3, \sigma[2/X] \rangle$
- Paralelní programy mohou být nedeterministické.

Verifikace paralelních a neukončených programů

- Ověření sémantické ekvivalence.
  - Specifikace i skutečná implementace se popíše ve vhodném „vyšším“ jazyce (CCS, Petriho sítě, apod.) s dobře definovanou operační sémantikou.
  - Dokáže se, že specifikace a implementace jsou ekvivalentní.
  - V tomto kontextu je formalizace pojmu sémantické ekvivalence netriviální problém (bisimulační ekvivalence apod.)
- Ověření platnosti formule vhodné logiky.
  - „Vhodnou“ logikou je v tomto případě obvykle nějaký typ modální (temporální) logiky.
  - temporální logiky lze klasifikovat z mnoha hledisek, např.
    - „state-based“ $\times$ „action-based“
    - „linear-time“ $\times$ „branching-time“
- Z praktického hlediska je hlavní omezující faktor velikost množiny konfigurací.

Logika LTL Syntaxe

- Buď $At = \{p, q, r, \dots\}$ spočetná množina atomických výroků.

  $$\varphi \quad ::= \quad \mathbf{true} \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathcal{X}\varphi \mid \varphi_1 \,\mathcal{U}\, \varphi_2$$

  Dále definujeme $\mathcal{F}\varphi \equiv \mathbf{true}\,\mathcal{U}\,\varphi$ a $\mathcal{G}\varphi \equiv \neg\mathcal{F}\neg\varphi$
- Buď φ LTL formule.
  - $At(\varphi)$ označuje množinu všech atomických výroků, které se vyskytují ve φ;
  - charakteristická abeceda formule φ je množina $\Sigma_\varphi = 2^{At(\varphi)}$;
  - Σ_φ^ω označuje množinu všech nekonečných slov nad abecedou Σ_φ;
  - nechť $w \in \Sigma_\varphi^\omega$. Symbol $w(i)$ označuje i-tý znak slova w; symbol w_i označuje i-tý sufix slova w pro každé $i \in \mathbb{N}_0$.

Příklad:

- $\varphi \;=\; (p \vee q)\,\mathcal{U}\,(p \wedge q)$
- $At(\varphi) \;=\; \{p, q\}$
- $\Sigma_\varphi \;=\; \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$
- je-li $w \;=\; \emptyset\,\{p\}\,\emptyset\,\{q\}\,\{q\}\,\{p, q\}\,\cdots$, platí $w(2) = \emptyset$, $w(3) = \{q\}$, $w_2 = \emptyset\,\{q\}\,\{q\}\,\{p, q\}\,\cdots$

Logika LTL Sémantika

- Platnost φ pro dané $w \in \Sigma_\varphi^\omega$ je definována indukcí ke struktuře φ:

$$
\begin{array}{lcl}
w \models \textbf{true} & & \\
w \models p & \iff & p \in w(0) \\
w \models \neg\varphi & \iff & w \not\models \varphi \\
w \models \varphi_1 \wedge \varphi_2 & \iff & w \models \varphi_1 \quad \wedge \quad w \models \varphi_2 \\
w \models \mathcal{X}\varphi & \iff & w_1 \models \varphi \\
w \models \varphi_1 \,\mathcal{U}\, \varphi_2 & \iff & \exists j: \quad w_j \models \varphi_2 \quad \wedge \quad \forall i < j: \; w_i \models \varphi_1
\end{array}
$$

- Charakteristický jazyk LTL formule φ je množina nekonečných slov $L_\varphi = \{w \in \Sigma_\varphi^\omega \mid w \models \varphi\}$

LTL jako jazyk vlastností neukončených programů

- Uvážíme „instanci“ LTL logiky, kde $At = \mathbf{Assn}$, tj.

  $$\varphi \quad ::= \quad \mathbf{true} \mid A \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathcal{X}\varphi \mid \varphi_1 \,\mathcal{U}\, \varphi_2$$

  kde $A \in \mathbf{Assn}$.

- Nechť $c \in \mathbf{PCom}$ a $\sigma \in \Sigma$. Běh programu c ze stavu σ je nekonečná posloupnost konfigurací

  $$\alpha \;=\; \langle c_0, \sigma_0\rangle \langle c_1, \sigma_1\rangle \langle c_2, \sigma_2\rangle \langle c_3, \sigma_3\rangle \cdots,$$

  kde $c_0 = c$, $\sigma_0 = \sigma$ a $\langle c_i, \sigma_i\rangle \mapsto \langle c_{i+1}, \sigma_{i+1}\rangle$ pro každé $i \in \mathbb{N}_0$.

- Buď φ LTL formule, I interpretace a α běh. Charakteristické slovo běhu α vzhledem k formuli φ a interpretaci I je slovo $\alpha^I_\varphi \in \Sigma^\omega_\varphi$, kde $\alpha^I_\varphi(i) = \{A \in \mathbf{Assn}(\varphi) \mid \sigma_i \models^I A\}$.

- Běh α splňuje LTL formuli φ při interpretaci I, psáno $\alpha \models^I \varphi$, jestliže $\alpha^I_\varphi \models \varphi$.

- Konfigurace $\langle c, \sigma\rangle$ splňuje LTL formuli φ při interpretaci I, psáno $\langle c, \sigma\rangle \models^I \varphi$, jestliže pro každý běh α začínající v $\langle c, \sigma\rangle$ platí $\alpha \models^I \varphi$.

LTL jako jazyk vlastností neukončených programů (2)

Příklady:

- $c \equiv$ **while tt do** $X := X + 1$. Pak $\langle c, \sigma \rangle \models^I \mathcal{F}(X > 5)$ pro každé σ a I.
- $c \equiv X := 2$. Pak $\langle c, \sigma \rangle \models^I \mathcal{G}(X = 3)$ pro každé σ a I, neboť neexistuje žádný běh začínající v $\langle c, \sigma \rangle$.
- $c \equiv X := 2 \parallel$ **while tt do skip**. Pak $\langle c, \sigma \rangle \not\models^I \mathcal{G}(X = 3)$ pro každé σ a I. Např. ale platí $\langle c, \sigma \rangle \models^I (X = 2) \Rightarrow \mathcal{G}(X = 2)$
- $c \equiv X := 2 \parallel X := 3 \parallel$ **while tt do skip**. Uvažme stav σ kde $\sigma(X) = 2$. Pak $\langle c, \sigma \rangle \not\models^I \mathcal{F}(X = 3)$ a $\langle c, \sigma \rangle \not\models^I \mathcal{G}(X \neq 3)$ pro každé I.

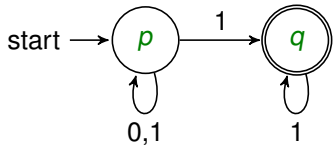
Problém: Jak efektivně ověřit, zda $\langle c, \sigma \rangle \models^I \varphi$?

ω-regulární jazyky a Büchiho automaty

- Büchiho automat je pětice $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, kde
  - Q je konečná množina stavů;
  - Σ je konečná abeceda;
  - $\delta \subseteq Q \times \Sigma \times Q$ je přechodová relace (místo $(p, a, q) \in \delta$ budeme psát $p \xrightarrow{a} q$);
  - $q_0 \in Q$ je počáteční stav;
  - $F \subseteq Q$ je množina koncových stavů.
- Výpočet automatu $\mathcal{A}$ na slově $w \in \Sigma^\omega$ je nekonečná posloupnost stavů $p_0\, p_1\, p_2 \cdots$ taková, že $p_0 = q_0$ a $p_i \xrightarrow{w(i)} p_{i+1}$ pro každé $i \in \mathbb{N}_0$. Výpočet je akceptující, jestliže se v něm některý koncový stav vyskytuje ∞-krát.
- Automat $\mathcal{A}$ akceptuje jazyk $L(\mathcal{A}) \subseteq \Sigma^\omega$ složený ze slov, pro která existuje akceptující výpočet.
- Buď Σ konečná abeceda. $L \subseteq \Sigma^\omega$ je ω-regulární, pokud existuje Büchiho automat $\mathcal{A}$ takový, že $L(\mathcal{A}) = L$.

ω-regulární jazyky a Büchiho automaty (2)

Příklad: Uvažme následující Büchiho automat $\mathcal{A}$:



Pak $L(\mathcal{A})$ obsahuje právě ta nekonečná slova nad abecedou 0, 1, ve kterých se 0 vyskytuje konečně-krát.

Vlastnosti Büchiho automatů a ω-regulárních jazyků

Věta 30

Nechť L_1, L_2 jsou ω-regulární jazyky nad abecedou Σ. Pak $L_1 \cup L_2$ a $L_1 \cap L_2$ jsou také ω-regulární jazyky.

Věta 31

Nechť $\mathcal{A}$ je Büchiho automat. Problém, zda $L(\mathcal{A}) = \emptyset$ je rozhodnutelný v polynomiálním čase).

Důkaz. Nechť $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$. Stačí si uvědomit, že $L(\mathcal{A}) \neq \emptyset$ právě když existuje $f \in F$ takový, že f je v (grafu) automatu $\mathcal{A}$ dosažitelný z q_0 a existuje cesta z f do f délky alespoň 1. Oba tyto (grafové) problémy jsou snadno rozhodnutelné v polynomiálním čase (dokonce v nedeterministickém logaritmickém prostoru).

Ověřování platnosti LTL formulí

- Buď $c \in \mathbf{PCom}$ a $\sigma \in \Sigma$ takové, že přechodový systém $(S, \{\tau\}, \mapsto)$, kde $S = \{\langle c', \sigma' \rangle \mid \langle c, \sigma \rangle \to^* \langle c', \sigma' \rangle\}$, má konečně mnoho konfigurací.
- Nechť φ je LTL formule a I interpretace. Definujeme Büchiho automat $\mathcal{A} = (S, \Sigma_\varphi, \delta, \langle c, \sigma \rangle, S)$, kde $\langle c', \sigma' \rangle \xrightarrow{M} \langle c'', \sigma'' \rangle$ právě když $\langle c', \sigma' \rangle \mapsto \langle c'', \sigma'' \rangle$ a $M = \{A \in \mathbf{Assn}(\varphi) \mid \sigma' \models^I \varphi\}$.
- Platí $\langle c, \sigma \rangle \models^I \varphi$ právě když $L(\mathcal{A}) \cap L(\mathcal{A}_{\neg\varphi}) = \emptyset$.

Ověřování platnosti LTL formulí (2)

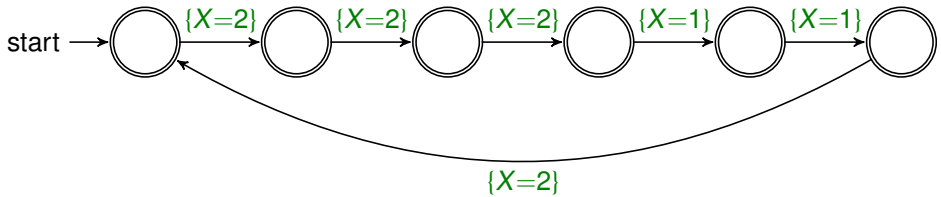
Příklad:

- Nechť $c \equiv$ **while tt do** $(X := 1; X := 2)$, $\varphi \equiv \mathcal{G}(X{=}1 \Rightarrow \mathcal{F}(X{=}2))$, $I \in \mathcal{I}$ a $\sigma \in \Sigma$ kde $\sigma(X) = 2$.
- Označíme-li $w \equiv$ **while tt do** $(X := 1; X := 2)$, platí

$$
\begin{aligned}
&\langle w, \sigma\rangle \quad \mapsto \\
&\langle \textbf{if tt then } (X := 1; X := 2; w) \textbf{ else skip}, \ \sigma\rangle \quad \mapsto \\
&\langle X := 1; X := 2; w, \ \sigma\rangle \quad \mapsto \\
&\langle \textbf{skip}; X := 2; w, \ \sigma[1/X]\rangle \quad \mapsto \\
&\langle X := 2; w, \ \sigma[1/X]\rangle \quad \mapsto \\
&\langle \textbf{skip}; w, \ \sigma\rangle \quad \mapsto \\
&\langle w, \ \sigma\rangle
\end{aligned}
$$

Ověřování platnosti LTL formulí (3)

Automat $\mathcal{A}$ tedy vypadá následovně:



- Dále $\neg\varphi \equiv \mathcal{F}(X{=}1 \wedge \mathcal{G}(\neg(X{=}2))$, jako $\mathcal{A}_{\neg\varphi}$ lze tedy použít (mírně modifikovaný) automat z předchozího příkladu.