

Sémantiky programovacích jazyků

Doporučená literatura

- Glynn Winskel: The Formal Semantics of Programming Languages
- Matthew Hennesy: The Semantics of Programming Languages
- PDF a PS soubor této prezentace je dostupný na <http://www.fi.muni.cz/usr/kucera/teach.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, *nelze „ladit“!*

$X := X + 1 \parallel X := X + 1$

- Návrh programovacích jazyků

Základní „styly“ sémantik programovacích jazyků

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

Abstraktní syntax programovacích jazyků

- Abstraktní syntaktická rovnice: rovnice tvaru

$$X ::= at_1 \mid \dots \mid at_n \mid op_1(\alpha_{(1,1)}, \dots, \alpha_{(1,n_1)}) \mid \dots \mid op_m(\alpha_{(m,1)}, \dots, \alpha_{(m,n_m)})$$

kde

- ★ at_1, \dots, at_n jsou **atomy**.
- ★ op_1, \dots, op_m jsou **operace** (které mohou mít i nulovou aritu – pak jde o **konstanty**).
- ★ $\alpha_{(i,j)}$ je buď X nebo atom (opakované výskyty jsou rozlišeny indexy).
- ★ Pro každý atom at_i je dána jeho **syntaktická doména** (množina) A_i .

- Příklad:

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

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

Syntaktické stromy

- Uvažme abstraktní syntaktickou rovnici

$$X ::= at_1 \mid \cdots \mid at_n \mid op_1(\alpha_{(1,1)}, \dots, \alpha_{(1,n_1)}) \mid \cdots \mid op_m(\alpha_{(m,1)}, \dots, \alpha_{(m,n_m)})$$

- Množina syntaktických stromů pro X je definována induktivně:

- ★ Je-li a prvek syntaktické domény některého z atomů at_1, \dots, at_n , je strom s jediným uzlem a syntaktický strom pro X výšky 0.

- ★ Je-li op konstanta, je strom s jediným uzlem op syntaktický strom pro X výšky 0.

- ★ Je-li op operace arity $n \geq 1$ s argumenty $\alpha_1, \dots, \alpha_n$, je strom s kořenem op a n následníky, kde i -tý následník je buď

- * kořen syntaktického stromu pro X , je-li $\alpha_i = X$,

- * prvek syntaktické domény atomu at , je-li $\alpha_i = at$.

také syntaktickým stromem pro X výšky $k + 1$, kde k je maximum z výšek následníků kořene (prvky syntaktických domén atomů mají výšku 0).

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

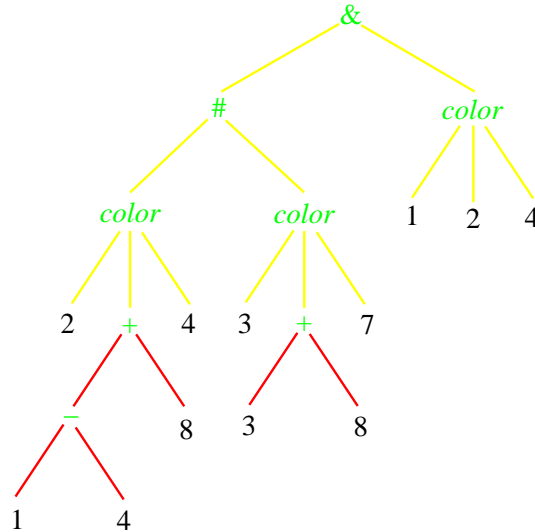
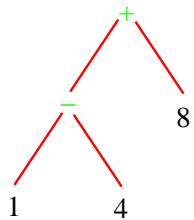
Příklad definice abstraktní syntaxe

- $X ::= \text{num} \mid X_0 + X_1 \mid X_0 - X_1$

Syntaktickou doménou atomu *num* jsou dekadické zápisy celých čísel.

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

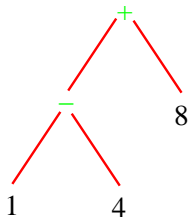
Syntaktickou doménou atomu *X* je množina všech syntaktických stromů pro *X*.



Konkrétní syntax programovacích jazyků

Určuje, jak jednoznačně zapisovat syntaktické stromy jako řetězce symbolů.

- Aritmetické výrazy



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**

- ★ závorky
- ★ klíčové slovo **fi**

Odvozovací systémy, důkazy a dokazatelná tvrzení

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

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

- **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**“ (tj. „obráceně“ než syntaktické stromy).

Odvozovací systém výrokové logiky

- **Abstraktní syntax** formulí výrokové logiky:

$$\varphi ::= \text{výrok} \mid \varphi_0 \rightarrow \varphi_1 \mid \neg \varphi$$

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

- **Schémata axiomů** odvozovacího systému výrokové logiky

$$\star \varphi \rightarrow (\psi \rightarrow \varphi)$$

$$\star (\varphi \rightarrow (\psi \rightarrow \xi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi))$$

$$\star (\neg \varphi \rightarrow \neg \psi) \rightarrow (\psi \rightarrow \varphi)$$

- **Odvozovací pravidlo** modus ponens

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi}$$

- **Příklad:** $A \rightarrow A$ je dokazatelná formule.

$$\frac{A \rightarrow (A \rightarrow A) \quad \frac{A \rightarrow ((A \rightarrow A) \rightarrow A) \quad (A \rightarrow ((A \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A))}{(A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)}}{A \rightarrow A}$$

Indukce k výšce stromu

- 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.
- **Tvrzení o indukci k výšce stromu:** Nechť je splněna následující podmínka:
 - ★ Pro každé $n \in \mathbb{N}_0$:

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} .
- **Strukturální indukce:** 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

Num = $\{0, 1, -1, 2, -2, \dots\}$

Bool = $\{\mathbf{tt}, \mathbf{ff}\}$

Var = $\{A, B, C, \dots\}$

- Aritmetické výrazy **Aexp**

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

kde $n \in \mathbf{Num}$ a $X \in \mathbf{Var}$.

- Pravdivostní výrazy **Bexp**

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

kde $t \in \mathbf{Bool}$ a $a_0, a_1 \in \mathbf{Aexp}$.

- Příkazy **Com**

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

kde $X \in \mathbf{Var}$, $a \in \mathbf{Aexp}$ a $b \in \mathbf{Bexp}$.

Operační sémantika

- Programům v daném jazyce je přiřazen **přechodový systém**, který popisuje výpočetní procesy jednotlivých programů.
- **Přechodový systém**: 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**
- Jednotlivé „typy“ operační sémantiky se liší definicí množiny konfigurací a přechodové relace
 - ★ **SMC stroj**: **S**tick – **M**emory – **C**ontrol stack
 - ★ **λ -kalkul**
 - ★ **SOS**: **S**trukturální **O**perační **S**émantika

SOS sémantika IMP prvního typu („big step“)

- **Stav** je zobrazení $\sigma : \mathbf{Var} \rightarrow \mathbb{Z}$, množina **všech** stavů se značí Σ .
- Cílem je definovat přechodový systém, kde
 - ★ množina konfigurací je Σ ,
 - ★ množina akcí je **Com**,
 - ★ přechodová relace odpovídá „výslednému efektu“ programů, tj. $\sigma \xrightarrow{c} \sigma'$ právě když výpočet programu c zahájený ve stavu σ skončí a přejde do stavu σ' .
- Za tímto účelem definujeme odvozovací systémy pro tři relace:
 - ★ $\rightarrow_A \subseteq \mathbf{Aexp} \times \Sigma \times \mathbb{Z}$; prvky zapisujeme ve tvaru $\langle a, \sigma \rangle \rightarrow_A n$.
 - ★ $\rightarrow_B \subseteq \mathbf{Bexp} \times \Sigma \times \mathbb{T}$; prvky zapisujeme ve tvaru $\langle b, \sigma \rangle \rightarrow_B t$.
 - ★ $\rightarrow_C \subseteq \mathbf{Com} \times \Sigma \times \Sigma$; prvky zapisujeme ve tvaru $\langle c, \sigma \rangle \rightarrow_C \sigma'$.Indexy A, B, C budou obvykle vynechány.
- Pak již lze **definovat**: $\sigma \xrightarrow{c} \sigma' \iff \langle c, \sigma \rangle \rightarrow \sigma'$

Aritmetické výrazy $\mathbf{A_{exp}}$

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

- $\langle \mathbf{n}, \sigma \rangle \rightarrow \mathbf{n}$
- $\langle \mathbf{X}, \sigma \rangle \rightarrow \sigma(\mathbf{X})$
- $$\frac{\langle \mathbf{a}_0, \sigma \rangle \rightarrow \mathbf{n}_0 \quad \langle \mathbf{a}_1, \sigma \rangle \rightarrow \mathbf{n}_1}{\langle \mathbf{a}_0 + \mathbf{a}_1, \sigma \rangle \rightarrow \mathbf{n}} \quad \mathbf{n} = \mathbf{n}_0 + \mathbf{n}_1$$
- $$\frac{\langle \mathbf{a}_0, \sigma \rangle \rightarrow \mathbf{n}_0 \quad \langle \mathbf{a}_1, \sigma \rangle \rightarrow \mathbf{n}_1}{\langle \mathbf{a}_0 - \mathbf{a}_1, \sigma \rangle \rightarrow \mathbf{n}} \quad \mathbf{n} = \mathbf{n}_0 - \mathbf{n}_1$$
- $$\frac{\langle \mathbf{a}_0, \sigma \rangle \rightarrow \mathbf{n}_0 \quad \langle \mathbf{a}_1, \sigma \rangle \rightarrow \mathbf{n}_1}{\langle \mathbf{a}_0 * \mathbf{a}_1, \sigma \rangle \rightarrow \mathbf{n}} \quad \mathbf{n} = \mathbf{n}_0 * \mathbf{n}_1$$

Pravdivostní výrazy Bexp

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

- $\langle tt, \sigma \rangle \rightarrow \mathbf{true}$
- $\langle ff, \sigma \rangle \rightarrow \mathbf{false}$
- $$\frac{\langle a_0, \sigma \rangle \rightarrow n_0 \quad \langle a_1, \sigma \rangle \rightarrow n_1}{\langle a_0 = a_1, \sigma \rangle \rightarrow \mathbf{true}} \quad n_0 = n_1$$
- $$\frac{\langle a_0, \sigma \rangle \rightarrow n_0 \quad \langle a_1, \sigma \rangle \rightarrow n_1}{\langle a_0 = a_1, \sigma \rangle \rightarrow \mathbf{false}} \quad n_0 \neq n_1$$
- $$\frac{\langle a_0, \sigma \rangle \rightarrow n_0 \quad \langle a_1, \sigma \rangle \rightarrow n_1}{\langle a_0 \leq a_1, \sigma \rangle \rightarrow \mathbf{true}} \quad n_0 \leq n_1$$
- $$\frac{\langle a_0, \sigma \rangle \rightarrow n_0 \quad \langle a_1, \sigma \rangle \rightarrow n_1}{\langle a_0 \leq a_1, \sigma \rangle \rightarrow \mathbf{false}} \quad n_0 > n_1$$
- $$\frac{\langle b, \sigma \rangle \rightarrow \mathbf{false}}{\langle \mathbf{not} \ b, \sigma \rangle \rightarrow \mathbf{true}}$$
- $$\frac{\langle b, \sigma \rangle \rightarrow \mathbf{true}}{\langle \mathbf{not} \ b, \sigma \rangle \rightarrow \mathbf{false}}$$
- $$\frac{\langle b_0, \sigma \rangle \rightarrow t_0 \quad \langle b_1, \sigma \rangle \rightarrow t_1}{\langle b_0 \ \mathbf{and} \ b_1, \sigma \rangle \rightarrow \mathbf{true}} \quad t_0 = \mathbf{true} \wedge t_1 = \mathbf{true}$$
- $$\frac{\langle b_0, \sigma \rangle \rightarrow t_0 \quad \langle b_1, \sigma \rangle \rightarrow t_1}{\langle b_0 \ \mathbf{and} \ b_1, \sigma \rangle \rightarrow \mathbf{false}} \quad t_0 = \mathbf{false} \vee t_1 = \mathbf{false}$$
- $$\frac{\langle b_0, \sigma \rangle \rightarrow t_0 \quad \langle b_1, \sigma \rangle \rightarrow t_1}{\langle b_0 \ \mathbf{or} \ b_1, \sigma \rangle \rightarrow \mathbf{true}} \quad t_0 = \mathbf{true} \vee t_1 = \mathbf{true}$$
- $$\frac{\langle b_0, \sigma \rangle \rightarrow t_0 \quad \langle b_1, \sigma \rangle \rightarrow t_1}{\langle b_0 \ \mathbf{or} \ b_1, \sigma \rangle \rightarrow \mathbf{false}} \quad t_0 = \mathbf{false} \wedge t_1 = \mathbf{false}$$

Příkazy Com

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

- $\langle \text{skip}, \sigma \rangle \rightarrow \sigma$

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

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

- $$\frac{\langle b, \sigma \rangle \rightarrow \text{true} \quad \langle c_0, \sigma \rangle \rightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'}$$

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

- $$\frac{\langle b, \sigma \rangle \rightarrow \text{false}}{\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma}$$

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

Příklad důkazového stromu v SOS sémantice 1. typu

Uvažme program **while** $A \leq 2$ **do** $A := A + C$

- Stav σ je definován takto: $\sigma(A) = 1$, $\sigma(C) = 2$, a pro $A \neq B \neq C$ je $\sigma(B) = 10$. Pak $\langle \text{while } A \leq 2 \text{ do } A := A + C, \sigma \rangle \rightarrow \sigma[3/A]$, neboť

$$\begin{array}{c}
 \frac{\langle A, \sigma \rangle \rightarrow 1 \quad \langle 2, \sigma \rangle \rightarrow 2}{\langle A \leq 2, \sigma \rangle \rightarrow \mathbf{true}} \quad \frac{\frac{\langle A, \sigma \rangle \rightarrow 1 \quad \langle C, \sigma \rangle \rightarrow 2}{\langle A + C, \sigma \rangle \rightarrow 3}}{\langle A := A + C, \sigma \rangle \rightarrow \sigma[3/A]} \quad \frac{\frac{\langle A, \sigma[3/A] \rangle \rightarrow 3 \quad \langle 2, \sigma[3/A] \rangle \rightarrow 2}{\langle A \leq 2, \sigma[3/A] \rangle \rightarrow \mathbf{false}}}{\langle \text{while } A \leq 2 \text{ do } A := A + C, \sigma[3/A] \rangle \rightarrow \sigma[3/A]} \\
 \hline
 \langle \text{while } A \leq 2 \text{ do } A := A + C, \sigma \rangle \rightarrow \sigma[3/A]
 \end{array}$$

- Stav $\sigma' = \sigma[0/C]$. Důkazový strom s kořenem tvaru $\langle \text{while } A \leq 2 \text{ do } A := A + C, \sigma' \rangle \rightarrow \sigma''$ sestrojit **nelze** (pro žádné σ'').

SOS sémantika prvního typu je deterministická

Věta 1.

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

Důkaz. 1. a 2. indukcí ke struktuře a a b , 3. indukcí k výšce odvození $\langle c, \sigma \rangle \rightarrow \sigma'$.

ad 1.

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

ad 3. Indukcí k výšce odvození ukážeme, že pokud pro dané c a σ existuje (nějaké) σ' takové, že $\langle c, \sigma \rangle \rightarrow \sigma'$, je toto σ' určeno jednoznačně.

Nechť $\langle c, \sigma \rangle \rightarrow \sigma'$ je kořen důkazového stromu výšky $n \in \mathbb{N}_0$. Uvážíme možné tvary c .

- $\langle \text{skip}, \sigma \rangle \rightarrow \sigma$.
- $c \equiv X := a$. Pak kořen $\langle X := a, \sigma \rangle \rightarrow \sigma'$ má následníka $\langle a, \sigma \rangle \rightarrow n$ 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 $\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$ má následníky $\langle c_0, \sigma \rangle \rightarrow \sigma''$ a $\langle c_1, \sigma'' \rangle \rightarrow \sigma'$. Podle indukčního předpokladu je σ'' i σ' určeno jednoznačně.
- $c \equiv \text{if } b \text{ then } c_0 \text{ else } c_1$. Pak kořen $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$ má buď následníky $\langle b, \sigma \rangle \rightarrow \text{true}$ a $\langle c_0, \sigma \rangle \rightarrow \sigma'$, nebo $\langle b, \sigma \rangle \rightarrow \text{false}$ a $\langle c_1, \sigma \rangle \rightarrow \sigma'$. Podle 2. nastává právě jedna z těchto možností, proto je σ' určeno jednoznačně.
- $c \equiv \text{while } b \text{ do } c$. Pak kořen $\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$ má buď jediného následníka $\langle b, \sigma \rangle \rightarrow \text{false}$ a $\sigma' = \sigma$, nebo tři následníky $\langle b, \sigma \rangle \rightarrow \text{true}$, $\langle c, \sigma \rangle \rightarrow \sigma''$ a $\langle \text{while } b \text{ do } c, \sigma'' \rangle \rightarrow \sigma'$. 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**

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

- Pravdivostní výrazy **Bexp**

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

- Příkazy **Com**

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

Příklad ekvivaletních programů

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

- $\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma' \Rightarrow \langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \rightarrow \sigma'$

Jsou dvě možnosti:

$$\star \frac{\overline{\dots} \quad \langle b, \sigma \rangle \rightarrow \text{false}}{\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma} \quad \text{Pak ale také} \quad \frac{\overline{\dots} \quad \langle b, \sigma \rangle \rightarrow \text{false} \quad \langle \text{skip}, \sigma \rangle \rightarrow \sigma}{\langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \rightarrow \sigma}$$

$$\star \frac{\overline{\dots} \quad \langle b, \sigma \rangle \rightarrow \text{true} \quad \overline{\dots} \quad \langle c, \sigma \rangle \rightarrow \sigma'' \quad \overline{\dots} \quad \langle \text{while } b \text{ do } c, \sigma'' \rangle \rightarrow \sigma'}{\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'} \quad \text{Pak ale také}$$

$$\frac{\overline{\dots} \quad \langle b, \sigma \rangle \rightarrow \text{true} \quad \overline{\dots} \quad \langle c, \sigma \rangle \rightarrow \sigma'' \quad \overline{\dots} \quad \langle \text{while } b \text{ do } c, \sigma'' \rangle \rightarrow \sigma'}{\langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \rightarrow \sigma'}$$

- 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 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 $\mathbf{A_{exp}}$

- $\langle n, \sigma \rangle$ – 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:

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

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

- $\langle \text{skip}, \sigma \rangle$ – konečná konfigurace

- $\langle X := n, \sigma \rangle \mapsto \langle \text{skip}, \sigma[n/X] \rangle$

$$\frac{\langle a, \sigma \rangle \mapsto \langle a', \sigma \rangle}{\langle X := a, \sigma \rangle \mapsto \langle X := a', \sigma \rangle}$$

- $\langle \text{skip}; c, \sigma \rangle \mapsto \langle c, \sigma \rangle$

$$\frac{\langle c_0, \sigma \rangle \mapsto \langle c'_0, \sigma' \rangle}{\langle c_0; c_1, \sigma \rangle \mapsto \langle c'_0; c_1, \sigma' \rangle}$$

- $\langle \text{if tt then } c_0 \text{ else } c_1, \sigma \rangle \mapsto \langle c_0, \sigma \rangle$

$$\langle \text{if ff then } c_0 \text{ else } c_1, \sigma \rangle \mapsto \langle c_1, \sigma \rangle$$

- $$\frac{\langle b, \sigma \rangle \mapsto \langle b', \sigma \rangle}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto \langle \text{if } b' \text{ then } c_0 \text{ else } c_1, \sigma \rangle}$$

- $\langle \text{while } b \text{ do } c, \sigma \rangle \mapsto \langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle$

Sémantická ekvivalence výrazů a příkazů (II)

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

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

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

- Aritmetické výrazy **Aexp**

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

- Pravdivostní výrazy **Bexp**

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

- Příkazy **Com**

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

Ekvivalence SOS sémantik 1. a 2. typu

Lema 2.

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 $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto^k \langle \text{if } b' \text{ then } c_0 \text{ else } c_1, \sigma \rangle$

Důkaz. Indukcí ke k .

ad 1., první implikace.

- **Báze ($k = 0$):** $\langle a_0, \sigma \rangle \mapsto^0 \langle a'_0, \sigma \rangle \iff a'_0 = a_0 \iff \langle a_0 + a_1, \sigma \rangle \mapsto^0 \langle a'_0 + a_1, \sigma \rangle$
- **Indukční krok:**
 $\langle a_0, \sigma \rangle \mapsto^{k+1} \langle a'_0, \sigma \rangle \Rightarrow \langle a_0, \sigma \rangle \mapsto \langle a''_0, \sigma \rangle$ a $\langle a''_0, \sigma \rangle \mapsto^k \langle a'_0, \sigma \rangle \Rightarrow$
 $\langle a_0, \sigma \rangle \mapsto \langle a''_0, \sigma \rangle$ a $\langle a''_0 + a_1, \sigma \rangle \mapsto^k \langle a'_0 + a_1, \sigma \rangle$ (podle I.P.) \Rightarrow
 $\langle a_0 + a_1, \sigma \rangle \mapsto \langle a''_0 + a_1, \sigma \rangle$ a $\langle a''_0 + a_1, \sigma \rangle \mapsto^k \langle a'_0 + a_1, \sigma \rangle \Rightarrow$
 $\langle a_0 + a_1, \sigma \rangle \mapsto^{k+1} \langle a'_0 + a_1, \sigma \rangle$

ad 2.,3.,4. Podobně.



Lema 3.

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 $\langle X := a, \sigma \rangle \mapsto^k \langle \text{skip}, \sigma' \rangle$, pak $\langle X := a, \sigma \rangle \mapsto^{k-1} \langle X := n, \sigma \rangle \mapsto \langle \text{skip}, \sigma[n/X] \rangle$, kde $\sigma' = \sigma[n/X]$ a $\langle a, \sigma \rangle \mapsto^{k-1} \langle n, \sigma \rangle$.
3. Jestliže $\langle c_0; c_1, \sigma \rangle \mapsto^k \langle \text{skip}, \sigma' \rangle$, pak $\langle c_0; c_1, \sigma \rangle \mapsto^l \langle \text{skip}; c_1, \sigma'' \rangle \mapsto \langle c_1, \sigma'' \rangle \mapsto^m \langle \text{skip}, \sigma' \rangle$, kde $l, m < k$ a $\langle c_0, \sigma \rangle \mapsto^l \langle \text{skip}, \sigma'' \rangle$.
4. Jestliže $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto^k \langle \text{skip}, \sigma' \rangle$, pak platí jedna z následujících možností:
 - ★ $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto^l \langle \text{if } tt \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto \langle c_0, \sigma \rangle \mapsto^m \langle \text{skip}, \sigma' \rangle$, kde $l, m < k$ a $\langle b, \sigma \rangle \mapsto^l \langle tt, \sigma \rangle$.
 - ★ $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto^l \langle \text{if } ff \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto \langle c_1, \sigma \rangle \mapsto^m \langle \text{skip}, \sigma' \rangle$, kde $l, m < k$ a $\langle b, \sigma \rangle \mapsto^l \langle ff, \sigma \rangle$.

Důkaz. Indukcí ke k .

ad 1. Báze indukce ($k = 0$)

- Jelikož $\langle a_0 \odot a_1, \sigma \rangle \not\mapsto^0 \langle n, \sigma \rangle$, implikace platí.

Indukční krok:

- Jestliže $a_0 \notin \mathbf{Num}$, pak $\langle a_0 \odot a_1, \sigma \rangle \mapsto \langle a'_0 \odot a_1, \sigma \rangle \mapsto^{k-1} \langle n, \sigma \rangle$, kde $\langle a_0, \sigma \rangle \mapsto \langle a'_0, \sigma \rangle$. Podle I.P. platí $\langle a'_0 \odot a_1, \sigma \rangle \mapsto^l \langle n_0 \odot a_1, \sigma \rangle \mapsto^m \langle n_0 \odot n_1, \sigma \rangle \mapsto \langle n, \sigma \rangle$, kde $l, m < k-1$, $\langle a'_0, \sigma \rangle \mapsto^l \langle n_0, \sigma \rangle$ a $\langle a_1, \sigma \rangle \mapsto^m \langle n_1, \sigma \rangle$. Proto také $\langle a_0 \odot a_1, \sigma \rangle \mapsto^{l+1} \langle n_0 \odot a_1, \sigma \rangle \mapsto^m \langle n_0 \odot n_1, \sigma \rangle \mapsto \langle n, \sigma \rangle$, kde $l+1, m < k$ a $\langle a_0, \sigma \rangle \mapsto^{l+1} \langle n_0, \sigma \rangle$.
- Jestliže $a_0 \equiv n_0$ a $a_1 \notin \mathbf{Num}$, pak $\langle n_0 \odot a_1, \sigma \rangle \mapsto \langle n_0 \odot a'_1, \sigma \rangle \mapsto^{k-1} \langle n, \sigma \rangle$, kde $\langle a_1, \sigma \rangle \mapsto \langle a'_1, \sigma \rangle$. Podle I.P. platí $\langle n_0 \odot a'_1, \sigma \rangle \mapsto^l \langle n_0 \odot a'_1, \sigma \rangle \mapsto^m \langle n_0 \odot n_1, \sigma \rangle \mapsto \langle n, \sigma \rangle$, kde $l, m < k-1$ (v tomto případě $l = 0$) a $\langle a'_1, \sigma \rangle \mapsto^m \langle n_1, \sigma \rangle$. Zbytek důkazu je podobný jako výše.
- Jestliže $a_0 \equiv n_0$ a $a_1 \equiv n_1$, stačí položit $l = m = 0$.

□

Věta 4.

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

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

ad 1.

- $a \equiv n$. Pak $\langle n, \sigma \rangle \rightarrow n$ a $\langle n, \sigma \rangle \mapsto^0 \langle n, \sigma \rangle$ dle definice.
- $a \equiv X$. Pak $\langle X, \sigma \rangle \rightarrow \sigma(X)$ a $\langle X, \sigma \rangle \mapsto \langle \sigma(X), \sigma \rangle$ dle definice.

- $a \equiv a_0 + a_1$. Podle I.P. platí

$$\star \langle a_0, \sigma \rangle \rightarrow n_0 \iff \langle a_0, \sigma \rangle \mapsto^* \langle n_0, \sigma \rangle$$

$$\star \langle a_1, \sigma \rangle \rightarrow n_1 \iff \langle a_1, \sigma \rangle \mapsto^* \langle n_1, \sigma \rangle$$

Dále

$$\langle a_0 + a_1, \sigma \rangle \rightarrow n \iff$$

$$\langle a_0, \sigma \rangle \rightarrow n_0 \text{ a } \langle a_1, \sigma \rangle \rightarrow n_1 \text{ kde } n = n_0 + n_1 \iff$$

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

$$\langle a_0 + a_1, \sigma \rangle \mapsto^* \langle n_0 + a_1 \rangle \mapsto^* \langle n_0 + n_1, \sigma \rangle \mapsto \langle n, \sigma \rangle \text{ kde } n = n_0 + n_1 \text{ (lema 2 (1))} \iff$$

$$\langle a_0 + a_1, \sigma \rangle \mapsto^* \langle n, \sigma \rangle \text{ (podle lematu 3 (1))}$$

- $a \equiv a_0 - a_1$. Podobně.

- $a \equiv a_0 * a_1$. Podobně.

ad 3.

(\Rightarrow) Indukcí k výšce odvození $\langle c, \sigma \rangle \rightarrow \sigma'$. Uvážíme možné tvary c .

- $\langle \text{skip}, \sigma \rangle \rightarrow \sigma$. Platí $\langle \text{skip}, \sigma \rangle \mapsto^0 \langle \text{skip}, \sigma \rangle$

- $c \equiv X := a$. Pak kořen $\langle X := a, \sigma \rangle \rightarrow \sigma'$ má následníka $\langle a, \sigma \rangle \rightarrow n$ a platí $\sigma' = \sigma[n/X]$. Podle 1. $\langle a, \sigma \rangle \mapsto^* \langle n, \sigma \rangle$, proto $\langle X := a, \sigma \rangle \mapsto^* \langle X := n, \sigma \rangle \mapsto^* \langle \text{skip}, \sigma[n/X] \rangle$ podle lematu 2 (2).

- $c \equiv c_0; c_1$. Pak kořen $\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$ má následníky $\langle c_0, \sigma \rangle \rightarrow \sigma''$ a $\langle c_1, \sigma'' \rangle \rightarrow \sigma'$. Podle indukčního předpokladu $\langle c_0, \sigma \rangle \mapsto^* \langle \text{skip}, \sigma'' \rangle$ a $\langle c_1, \sigma'' \rangle \mapsto^* \langle \text{skip}, \sigma' \rangle$. Podle **lematu 2 (3)** platí $\langle c_0; c_1, \sigma \rangle \mapsto^* \langle \text{skip}; c_1, \sigma'' \rangle \mapsto \langle c_1, \sigma'' \rangle \mapsto^* \langle \text{skip}, \sigma' \rangle$.
- $c \equiv \text{if } b \text{ then } c_0 \text{ else } c_1$. Pak kořen $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$ má buď následníky $\langle b, \sigma \rangle \rightarrow \text{true}$ a $\langle c_0, \sigma \rangle \rightarrow \sigma'$, nebo $\langle b, \sigma \rangle \rightarrow \text{false}$ a $\langle c_1, \sigma \rangle \rightarrow \sigma'$. V prvním případě $\langle b, \sigma \rangle \mapsto^* \langle \text{tt}, \sigma \rangle$ a $\langle c_0, \sigma \rangle \mapsto^* \langle \text{skip}, \sigma' \rangle$ (podle I.P. a 2.), tedy $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto^* \langle \text{if tt then } c_0 \text{ else } c_1, \sigma \rangle \mapsto \langle c_0, \sigma \rangle \mapsto^* \langle \text{skip}, \sigma' \rangle$ podle **lematu 2 (4)**. Druhý případ se dokáže podobně.
- $c \equiv \text{while } b \text{ do } c$. Pak kořen $\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$ má buď jediného následníka $\langle b, \sigma \rangle \rightarrow \text{false}$ a $\sigma' = \sigma$, nebo tři následníky $\langle b, \sigma \rangle \rightarrow \text{true}$, $\langle c, \sigma \rangle \rightarrow \sigma''$ a $\langle \text{while } b \text{ do } c, \sigma'' \rangle \rightarrow \sigma'$. V prvním případě $\langle b, \sigma \rangle \mapsto^* \langle \text{ff}, \sigma \rangle$ podle 2., proto $\langle \text{while } b \text{ do } c, \sigma \rangle \mapsto \langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \mapsto^* \langle \text{if ff then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \mapsto \langle \text{skip}, \sigma \rangle$ podle **lematu 2 (4)**. V druhém případě $\langle b, \sigma \rangle \mapsto^* \langle \text{tt}, \sigma \rangle$, $\langle c, \sigma \rangle \mapsto^* \langle \text{skip}, \sigma'' \rangle$ a $\langle \text{while } b \text{ do } c, \sigma'' \rangle \mapsto^* \langle \text{skip}, \sigma' \rangle$ podle 2. a I.P. Proto také $\langle \text{while } b \text{ do } c, \sigma \rangle \mapsto \langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \mapsto^* \langle \text{if tt then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \mapsto \langle c; \text{while } b \text{ do } c, \sigma \rangle \mapsto^* \langle \text{while } b \text{ do } c, \sigma'' \rangle \mapsto^* \langle \text{skip}, \sigma' \rangle$ podle **lematu 2**.

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

Báze indukce:

- $\langle \text{skip}, \sigma \rangle \mapsto^0 \langle \text{skip}, \sigma \rangle$. Platí $\langle \text{skip}, \sigma \rangle \rightarrow \sigma$.

Indukční krok: Nechť $\langle c, \sigma \rangle \mapsto^k \langle \text{skip}, \sigma' \rangle$ kde $k \geq 1$. Uvážíme možné tvary c .

- $c \equiv X := a$. Pak $\langle X := a, \sigma \rangle \mapsto^{k-1} \langle X := n, \sigma \rangle \mapsto \langle \text{skip}, \sigma[n/X] \rangle$, kde $\langle a, \sigma \rangle \mapsto^{k-1} \langle n, \sigma \rangle$ (podle **lematu 3 (2)**). Proto také $\langle a, \sigma \rangle \rightarrow n$ podle 1. Tedy $\langle X := a, \sigma \rangle \rightarrow \sigma[n/X]$.
- $c \equiv c_0; c_1$. Pak $\langle c_0; c_1, \sigma \rangle \mapsto^l \langle \text{skip}; c_1, \sigma'' \rangle \mapsto \langle c_1, \sigma'' \rangle \mapsto^m \langle \text{skip}, \sigma' \rangle$ kde $l, m < k$ a $\langle c_0, \sigma \rangle \mapsto^l \langle \text{skip}, \sigma'' \rangle$ podle **lematu 3 (3)**. Podle I.P. $\langle c_0, \sigma \rangle \rightarrow \sigma''$ a $\langle c_1, \sigma'' \rangle \rightarrow \sigma'$, tedy $\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$.
- $c \equiv \text{if } b \text{ then } c_0 \text{ else } c_1$. Podle **lematu 3 (4)** jsou dvě možnosti:
 - ★ $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto^l \langle \text{if } tt \text{ then } c_0 \text{ else } c_1, \sigma \rangle \mapsto \langle c_0, \sigma \rangle \mapsto^m \langle \text{skip}, \sigma' \rangle$, kde $l, m < k$ a $\langle b, \sigma \rangle \mapsto^l \langle tt, \sigma \rangle$ (podle **lematu 3 (4)**). Dále podle 2. a I.P. platí $\langle b, \sigma \rangle \rightarrow \text{true}$ a $\langle c_0, \sigma \rangle \rightarrow \sigma'$, tedy také $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$.
 - ★ Druhá možnost se ověří podobně.
- $c \equiv \text{while } b \text{ do } c$. Pak $\langle \text{while } b \text{ do } c, \sigma \rangle \mapsto \langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle$. Podle **lematu 3 (4)** jsou dvě možnosti:
 - ★ $\langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \mapsto^l \langle \text{if } tt \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \mapsto \langle c; \text{while } b \text{ do } c, \sigma \rangle \mapsto^m \langle \text{skip}, \sigma' \rangle$, kde $l, m < k$ a $\langle b, \sigma \rangle \mapsto^l \langle tt, \sigma \rangle$. Opětovným použitím **lematu 3 (3)** dostáváme $\langle c; \text{while } b \text{ do } c, \sigma \rangle \mapsto^{l'} \langle \text{skip}; \text{while } b \text{ do } c, \sigma'' \rangle \mapsto \langle \text{while } b \text{ do } c, \sigma'' \rangle \mapsto^{m'} \langle \text{skip}, \sigma' \rangle$, kde $l', m' < m < k$ a $\langle c, \sigma \rangle \mapsto^{l'} \langle \text{skip}, \sigma'' \rangle$. Podle 2. a I.P. dostáváme $\langle b, \sigma \rangle \rightarrow \text{true}$, $\langle c, \sigma \rangle \rightarrow \sigma''$ a $\langle \text{while } b \text{ do } c, \sigma'' \rangle \rightarrow \sigma'$, proto $\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$.

★ $\langle \text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \mapsto^l \langle \text{if ff then } (c; \text{while } b \text{ do } c) \text{ else skip}, \sigma \rangle \mapsto \langle \text{skip}, \sigma' \rangle \mapsto^m \langle \text{skip}, \sigma' \rangle$, kde $l, m < k$ a $\langle b, \sigma \rangle \mapsto^l \langle \text{ff}, \sigma \rangle$ (v tomto případě je $m = 0$ a $\sigma' = \sigma$). Podle 2. platí $\langle b, \sigma \rangle \rightarrow \text{false}$ a tedy $\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma$.

Denotační sémantika IMP

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

Argumenty funkcí $\mathcal{A}, \mathcal{B}, \mathcal{C}$ se píší do „sémantických“ závorek $\llbracket \cdot \rrbracket$

Aritmetické výrazy \mathcal{Aexp}

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

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

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

Úplné částečné uspořádání (CPO)

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

$$d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq d_3 \dots$$

prvků z D má v D supréмум.

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.
- $(\Sigma \rightarrow \Sigma, \subseteq)$ je CPO.
 - ★ $f \subseteq g$ je-li g „více definovaná“ než f .

Monotónní a spojitě funkce

- Necht' (D, \sqsubseteq) , (E, \preceq) jsou CPO, $f : D \rightarrow E$ totální funkce. f je **monotónní**, jestliže pro každé $a, b \in D$ platí:

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

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

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

Příklad:

- Každá funkce z diskrétního CPO je spojitá.

Postačující podmínka spojitosti

Věta 5. Bud' M množina, $f : 2^M \rightarrow 2^M$ 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

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

- **Spojitost:** Bud' $A_1 \subseteq A_2 \subseteq A_3 \subseteq A_4 \dots$ nekonečný řetěz podmnožin M . Pak

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



Věta o pevném bodě

Věta 6. *Bud' (D, \sqsubseteq) CPO mající nejmenší prvek \perp a Γ spojitá funkce na D . Položme*

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

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

Důkaz.

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

$$\Gamma(\mu\Gamma) = \Gamma(\bigsqcup_{i \in \mathbb{N}_0} \Gamma^i(\perp)) = \bigsqcup_{i \in \mathbb{N}_0} \Gamma^{i+1}(\perp) = \bigsqcup_{i \in \mathbb{N}_0} \Gamma^i(\perp) = \mu\Gamma$$

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

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

□

Denotační sémantika **while** cyklu

- Označme $w \equiv \text{while } b \text{ do } c.$
- Platí $w \sim \text{if } b \text{ then } c; w \text{ else skip}$ (viz strana 22).
- Proto by mělo platit také $\mathcal{C}[[w]] = \mathcal{C}[[\text{if } b \text{ then } c; w \text{ else skip}]]$
- Tedy

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

- Tuto rovnost nelze chápat definitoricky, ale lze na ni nahlížet jako na „návod“, jak pro danou aproximaci $\mathcal{C}[[w]]$ spočítat „lepší“ aproximaci.
- Definujeme funkci $\Gamma : (\Sigma \rightarrow \Sigma) \rightarrow (\Sigma \rightarrow \Sigma)$ předpisem

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

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

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

a lze tedy aplikovat **větu 5**.

- $\mathcal{C}[[w]]$ by mělo být **pevným bodem** funkce Γ , tj. $\Gamma(\mathcal{C}[[w]]) = \mathcal{C}[[w]]$.
- Γ může mít více pevných bodů; má-li však $\mathcal{C}[[w]]$ odpovídat intuitivnímu významu **while** cyklu, je třeba **definovat**

$$\mathcal{C}[[w]] = \mu\Gamma$$

Nejmenší pevný bod Γ existuje podle **věty 6** 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** $X := X + 1$

$$\Gamma^0(\emptyset) = \emptyset$$

$$\Gamma^1(\emptyset) = \{(\sigma, \sigma) \mid \sigma(X) > 1\}$$

$$\Gamma^2(\emptyset) = \Gamma^1(\emptyset) \cup \{(\sigma, \sigma[2/X]) \mid \sigma(X) = 1\}$$

$$\Gamma^3(\emptyset) = \Gamma^2(\emptyset) \cup \{(\sigma, \sigma[2/X]) \mid \sigma(X) = 0\}$$

$$\Gamma^4(\emptyset) = \Gamma^3(\emptyset) \cup \{(\sigma, \sigma[2/X]) \mid \sigma(X) = -1\}$$

\vdots

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

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

$$\Gamma^0(\emptyset) = \emptyset$$

$$\Gamma^1(\emptyset) = \emptyset$$

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

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

$$\Gamma^0(\emptyset) = \emptyset$$

$$\Gamma^1(\emptyset) = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$$

$$\Gamma^2(\emptyset) = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$$

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

Ekvivalence operační a denotační sémantiky

Věta 7.

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

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

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

- $\langle \text{skip}, \sigma \rangle \rightarrow \sigma$. Platí $\mathcal{C}[[\text{skip}]]\sigma = \sigma$ podle definice.
- $c \equiv X := a$. Pak kořen $\langle X := a, \sigma \rangle \rightarrow \sigma'$ má následníka $\langle a, \sigma \rangle \rightarrow n$ a platí $\sigma' = \sigma[n/X]$. Podle 1. $\mathcal{A}[[a]]\sigma = n$ a $\mathcal{C}[[X := a]]\sigma = \sigma[n/X]$ dle definice.
- $c \equiv c_0; c_1$. Pak kořen $\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$ má následníky $\langle c_0, \sigma \rangle \rightarrow \sigma''$ a $\langle c_1, \sigma'' \rangle \rightarrow \sigma'$. Podle I.P. $\mathcal{C}[[c_0]]\sigma = \sigma''$ a $\mathcal{C}[[c_1]]\sigma'' = \sigma'$, proto $(\sigma, \sigma') \in \mathcal{C}[[c_1]] \circ \mathcal{C}[[c_0]] = \mathcal{C}[[c_0; c_1]]$.
- $c \equiv \text{if } b \text{ then } c_0 \text{ else } c_1$. Pak kořen $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$ má buď následníky $\langle b, \sigma \rangle \rightarrow \text{true}$ a $\langle c_0, \sigma \rangle \rightarrow \sigma'$, nebo $\langle b, \sigma \rangle \rightarrow \text{false}$ a $\langle c_1, \sigma \rangle \rightarrow \sigma'$. V prvním případě $\mathcal{B}[[b]]\sigma = \text{true}$ a $\mathcal{C}[[c_0]]\sigma = \sigma'$ (podle I.P. a 2.), tedy $(\sigma, \sigma') \in \mathcal{C}[[\text{if } b \text{ then } c_0 \text{ else } c_1]]$ podle definice. Druhý případ se dokáže podobně.
- $c \equiv \text{while } b \text{ do } c$. Pak kořen $\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$ má buď

- ★ jediného následníka $\langle b, \sigma \rangle \rightarrow \mathbf{false}$ a platí $\sigma' = \sigma$. Pak $\mathcal{B}[b]\sigma = \mathbf{false}$ podle 2., proto $(\sigma, \sigma) \in \Gamma(\emptyset) \subseteq \mu\Gamma$;
- ★ nebo tři následníky $\langle b, \sigma \rangle \rightarrow \mathbf{true}$, $\langle c, \sigma \rangle \rightarrow \sigma''$ a $\langle \mathbf{while\ } b \mathbf{ do\ } c, \sigma'' \rangle \rightarrow \sigma'$. Pak $\mathcal{B}[b]\sigma = \mathbf{true}$ podle 2. a $(\sigma, \sigma'') \in \mathcal{C}[c]$, $(\sigma'', \sigma') \in \mathcal{C}[\mathbf{while\ } b \mathbf{ do\ } c]$ 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 $(\sigma, \sigma') \in \mu\Gamma$.

„ \Leftarrow “ Indukcí ke struktuře c .

- $c \equiv \mathbf{skip}$. Platí $\mathcal{C}[\mathbf{skip}]\sigma = \sigma$ a $\langle \mathbf{skip}, \sigma \rangle \rightarrow \sigma$ podle definice.
- $c \equiv X := a$. Platí $\mathcal{C}[X := a]\sigma = \sigma[n/X]$ kde $\mathcal{A}[a]\sigma = n$. Podle 1. $\langle a, \sigma \rangle \rightarrow n$, proto $\langle X := a, \sigma \rangle \rightarrow \sigma[n/X]$.
- $c \equiv c_0; c_1$. 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 $\langle c_0, \sigma \rangle \rightarrow \sigma''$ a $\langle c_1, \sigma'' \rangle \rightarrow \sigma'$, tedy $\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$.
- $c \equiv \mathbf{if\ } b \mathbf{ then\ } c_0 \mathbf{ else\ } c_1$. Jestliže $(\sigma, \sigma') \in \mathcal{C}[\mathbf{if\ } b \mathbf{ then\ } c_0 \mathbf{ else\ } c_1]$, jsou dvě možnosti:
 - ★ $\mathcal{B}[b]\sigma = \mathbf{true}$ a $(\sigma, \sigma') \in \mathcal{C}[c_0]$. Podle 2. a I.P. platí $\langle b, \sigma \rangle \rightarrow \mathbf{true}$ a $\langle c_0, \sigma \rangle \rightarrow \sigma'$, proto $\langle \mathbf{if\ } b \mathbf{ then\ } c_0 \mathbf{ else\ } c_1, \sigma \rangle \rightarrow \sigma'$.
 - ★ Druhá možnost se ověří podobně.
- $c \equiv \mathbf{while\ } b \mathbf{ do\ } c$. Jestliže $(\sigma, \sigma') \in \mathcal{C}[\mathbf{while\ } b \mathbf{ do\ } c] = \mu\Gamma$, 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 $\langle \mathbf{while\ } b \mathbf{ do\ } c, \sigma \rangle \rightarrow \sigma'$.
 $k = 0$. Jelikož $(\sigma, \sigma') \notin \Gamma^0(\emptyset) = \emptyset$, dokazovaná implikace platí.

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

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



Axiomatická sémantika IMP

- Uvažme program $P \equiv \text{while } X < 0 \text{ do } (X := X + 1; Y := Y - 1); Z := Z - (X + Y)$
- Jestliže před spuštěním P platí $X + Y = i$ a $Z = j$, pak po dokončení P platí $Z = j - i$.
- Jak to dokázat?
 - ★ pomocí operační nebo denotační sémantiky – problematické.
 - ★ Hoare: do P doplníme „tvrzení“ (pravdivá v „místě“ jejich výskytu) a podáme odvozovací systém, který umožní jejich platnost dokázat.

- **Příklad:**

$\{X + Y = i \wedge Z = j\}$

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

$\{Z - (X + Y) = j - i\}$

$Z := Z - (X + Y)$

$\{Z = j - i\}$

- 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. $\{\mathbf{true}\} \quad \mathbf{while \ tt \ do \ skip} \quad \{\mathbf{false}\}$

- Pro zjednodušení notace zavedeme speciální „stav“ \perp :

★ $\Sigma_{\perp} = \Sigma \cup \{\perp\}$

★ Dále zavedeme funkci $\mathcal{C}_{\perp} : \mathbf{Com} \rightarrow (\Sigma_{\perp} \rightarrow \Sigma_{\perp})$:

* $\mathcal{C}_{\perp}[[c]]\perp = \perp$ pro každé $c \in \mathbf{Com}$;

* $\mathcal{C}_{\perp}[[c]]\sigma = \mathcal{C}[[c]]\sigma$ pro každé $c \in \mathbf{Com}$ a $\sigma \in \Sigma$, kde $\mathcal{C}[[c]]\sigma$ je definováno;

* $\mathcal{C}_{\perp}[[c]]\sigma = \perp$ pro každé $c \in \mathbf{Com}$ a $\sigma \in \Sigma$, kde $\mathcal{C}[[c]]\sigma$ je nedefinováno.

- Význam $\{A\} \quad c \quad \{B\}$ pak lze vyjádřit takto:

$$\{A\} \quad c \quad \{B\} \quad \Longleftrightarrow \quad \forall \sigma \in \Sigma : \quad \sigma \models A \quad \Rightarrow \quad \mathcal{C}_{\perp}[[c]]\sigma \models B$$

Syntaxe a sémantika tvrzení o programech

- Buď **IntVar** = {i, j, k, ...} spočetná množina celočíselných proměnných.

- Rozšířené aritmetické výrazy **Aexpv**

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

kde $n \in \mathbf{Num}$, $X \in \mathbf{Var}$ a $i \in \mathbf{IntVar}$.

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

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

kde $a_0, a_1 \in \mathbf{Aexpv}$ a $i \in \mathbf{IntVar}$.

- **Interpretace** je funkce $I : \mathbf{IntVar} \rightarrow \mathbb{Z}$. Množinu všech interpretací značíme \mathcal{I} .

- Definujeme funkci $\mathcal{E} : \mathbf{Aexpv} \rightarrow (\mathcal{I} \rightarrow (\Sigma \rightarrow \mathbb{Z}))$
 - ★ $\mathcal{E}[\![n]\!]I\sigma = n$
 - ★ $\mathcal{E}[\![X]\!]I\sigma = \sigma(X)$
 - ★ $\mathcal{E}[\![i]\!]I\sigma = I(i)$
 - ★ $\mathcal{E}[\![a_0 + a_1]\!]I\sigma = \mathcal{E}[\![a_0]\!]I\sigma + \mathcal{E}[\![a_1]\!]I\sigma$ (podobně pro „−“ a „*“)
- A je splněno ve stavu $\sigma \in \Sigma_\perp$ za interpretace $I \in \mathcal{I}$ (psáno $\sigma \models^I A$)
 - ★ $\perp \models^I A$ pro každé $A \in \mathbf{Assn}$; je-li $\sigma \neq \perp$, 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 $I \in \mathcal{I}$.

Tvrzení o částečné korektnosti programů

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

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

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

- $\sigma \models^I \{A\} c \{B\} \iff (\sigma \models^I A \implies \mathcal{C}_\perp \llbracket c \rrbracket \sigma \models^I B)$
- $\models^I \{A\} c \{B\} \iff \forall \sigma \in \Sigma_\perp : \sigma \models^I \{A\} c \{B\}$
 - ★ Platí $\perp \models^I \{A\} c \{B\}$ pro každé $I \in \mathcal{I}$, $A, B \in \mathbf{Assn}$, $c \in \mathbf{Com}$.
 - ★ Proto $\models^I \{A\} c \{B\} \iff \forall \sigma \in \Sigma : \sigma \models^I \{A\} c \{B\}$ (lze využít v důkazech).
- $\models \{A\} c \{B\} \iff \forall I \in \mathcal{I} : \models^I \{A\} c \{B\}$
- Tvrzení $\{A\} c \{B\}$ pro které platí $\models \{A\} c \{B\}$ nazýváme **platné**.

Hoareův odvozovací systém pro tvrzení o částečné korektnosti

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

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

- Pravidlo pro větvení:

$$\frac{\{A \wedge b\} c_0 \{B\} \quad \{A \wedge \neg b\} c_1 \{B\}}{\{A\} \text{ if } b \text{ then } c_0 \text{ else } c_1 \{B\}}$$

- Pravidlo pro cyklus:

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

- Pravidlo důsledku:

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

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

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

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

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

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

Korektnost Hoareova odvozovacího systému

Lema 9. Necht' $I \in \mathcal{I}$, $X \in \mathbf{Var}$, $a_0, a_1 \in \mathbf{Aexpv}$, $a \in \mathbf{Aexp}$ a $B \in \mathbf{Assn}$. Pak pro každé $\sigma \in \Sigma$ platí:

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

Důkaz. Indukcí ke struktuře a_0 , resp. B . □

Věta 10 (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 $\{A\} \mathbf{skip} \{A\}$. Toto tvrzení je zjevně platné.
- **Axiom pro $X := a$.** Pak je kořen tvaru $\{B[a/X]\} X := a \{B\}$. Necht' $I \in \mathcal{I}$ a $\sigma \in \Sigma$. Podle lematu 9 platí $\sigma \models^I B[a/X] \iff \sigma[\mathcal{A}[[a]]\sigma/X] \models^I B$. Jelikož $\sigma[\mathcal{A}[[a]]\sigma/X] = \mathcal{C}[[X := a]]\sigma$, dostáváme

$$\sigma \models^I B[a/X] \Rightarrow \mathcal{C}[[X := a]]\sigma \models^I B,$$

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

- **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 $\mathcal{C}_\perp[\![c_0]\!]\sigma \models^I C$ (neboť $\models \{A\} c_0 \{C\}$) a $\mathcal{C}_\perp[\![c_1]\!](\mathcal{C}_\perp[\![c_0]\!]\sigma) \models^I B$, protože $\models \{C\} c_1 \{B\}$ a $\mathcal{C}_\perp[\![c_0]\!]\sigma \models^I C$. Tedy $\models \{A\} c_0; c_1 \{B\}$.
- **Pravidlo pro větvení.** Pak kořen $\{A\} \text{if } b \text{ then } c_0 \text{ else } c_1 \{B\}$ 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 prvním případě $\sigma \models^I A \wedge b$, tedy $\mathcal{C}_\perp[\![c_0]\!]\sigma \models^I B$, neboť $\models \{A \wedge b\} c_0 \{B\}$. V druhém případě $\sigma \models^I A \wedge \neg b$, tedy $\mathcal{C}_\perp[\![c_1]\!]\sigma \models^I B$. Celkem $\models \{A\} \text{if } b \text{ then } c_0 \text{ else } c_1 \{B\}$ (užitím **lematu 8**).
- **Pravidlo pro cyklus.** Pak kořen $\{A\} \text{while } b \text{ do } c \{A \wedge \neg b\}$ 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

$$\sigma \models^I A \quad \Rightarrow \quad \mathcal{C}_\perp[\![\text{while } b \text{ do } c]\!]\sigma \models^I A \wedge \neg b$$

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

★ $j = 0$. 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:
 - * $\mathcal{B}[b]\sigma = \mathbf{true}$ a $(\sigma, \sigma') \in \Gamma^j(\emptyset) \circ \mathcal{C}[c]$. Pak $\sigma \models^I b$ (užitím **lematu 8**), tedy $\sigma \models^I A \wedge b$. Dále existuje σ'' takové, že $(\sigma, \sigma'') \in \mathcal{C}[c]$ 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.
 - * $\mathcal{B}[b]\sigma = \mathbf{false}$ 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 $\mathcal{C}_\perp[c]\sigma \models^I B'$ a tudíž i $\mathcal{C}_\perp[c]\sigma \models^I B$.

□

Nejslabší vstupní podmínka

- Chceme-li odvodit tvrzení $\{A\} c_0; c_1 \{B\}$, lze to podle pravidla pro sekvenční kompozici provést tak, že pro vhodně zvolené $C \in \mathbf{Assn}$ odvodíme tvrzení $\{A\} c_0 \{C\}$ a $\{C\} c_1 \{B\}$.
- Dokážeme, že takové C existuje pro libovolné A, B, c_0, c_1 ; tímto C bude tzv. **nejslabší vstupní podmínka** pro c_1 a B (vyjádřená v \mathbf{Assn}).
- Necht' $c \in \mathbf{Com}$, $B \in \mathbf{Assn}$ a $I \in \mathcal{I}$. Nejslabší vstupní podmínka pro B vzhledem k c a I , označovaná $wp^I[[c, B]]$, je definovaná takto:

$$wp^I[[c, B]] = \{\sigma \in \Sigma_{\perp} \mid \mathcal{C}_{\perp}[[c]]\sigma \models^I B\}.$$

- Zavedeme značení $A^I = \{\sigma \in \Sigma_{\perp} \mid \sigma \models^I A\}$.
- 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 – příklady

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

Gödelův predikát β

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

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

- 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 $\mathcal{S}_{a,b}(i) = a \bmod(1 + b(1 + i))$.
- Predikát β je vyjádřitelný v **Assn**, neboť $x = a \bmod b$ lze napsat jako

$$a \geq 0 \quad \wedge \quad b \geq 0 \quad \wedge$$

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

Věta 11. Pro každou konečnou posloupnost n_0, \dots, 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)

- Necht'

$$m = (\max\{k, n_0, \dots, n_k\})!$$

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

$$c_i = p_0 \cdot \dots \cdot p_k / p_i.$$

Nyní pro každé $0 \leq i \leq k$ existuje přesně jedno d_i , $0 \leq d_i \leq p_i$, takové, že $(c_i \cdot d_i) \bmod p_i = 1$

- Definujeme

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

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



Rozšířený predikát β^\pm

- Cílem je vytvořit prostředek pro kódování konečných posloupností hodnot proměnných jazyka **IMP**, tj. konečných posloupností **celých** čísel.
- Celá čísla lze seřadit do posloupnosti $0, 0, 1, -1, 2, -2, 3, -3, \dots$. 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

$$F(x, y) \iff x \geq 0 \wedge \exists z : (x = 2 * z \Rightarrow y = z) \wedge (x = 2 * z + 1 \Rightarrow y = -z)$$

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

$$\beta^\pm(n, m, j, y) \iff \exists x : \beta(n, m, j, x) \wedge F(x, y)$$

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

Věta 12. Pro každou konečnou posloupnost n_0, \dots, n_k celých čísel existují $n, m \in \mathbb{N}_0$ taková, že $n_j = \mathcal{S}_{n,m}^\pm(j)$ 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 β^\pm

- Lze v **Assn** vyjádřit $X \geq 0 \wedge Y = X!$?
- „Zakódujeme“ posloupnost $1, 1, 2, 6, 24, \dots, X!$

★ $Y = X!$

★ $\exists \mathcal{S} .$

$$\begin{aligned} & \mathcal{S}(0) = 1 \\ & \wedge \forall l . 1 \leq l \leq X \Rightarrow (\mathcal{S}(l) = \mathcal{S}(l-1) * l) \\ & \wedge Y = \mathcal{S}(X) \end{aligned}$$

★ $\exists \mathcal{S} .$

$$\begin{aligned} & \mathcal{S}(0) = 1 \\ & \wedge \forall l . 1 \leq l \leq X \Rightarrow (\forall u . \forall v . u = \mathcal{S}(l) \wedge v = \mathcal{S}(l-1) \Rightarrow u = v * l) \\ & \wedge Y = \mathcal{S}(X) \end{aligned}$$

★ $\exists n . \exists m . \beta^\pm(n, m, 0, 1)$

$$\begin{aligned} & \wedge \forall l . 1 \leq l \leq X \Rightarrow (\forall u . \forall v . \beta^\pm(n, m, l, u) \wedge \beta^\pm(n, m, l-1, v) \Rightarrow \\ & \quad u = v * l) \\ & \wedge \beta^\pm(n, m, X, Y) \end{aligned}$$

Vyjádřitelnost nejslabší vstupní podmínky v \mathbf{Assn}

Věta 13. Pro každé $c \in \mathbf{Com}$ a $B \in \mathbf{Assn}$ existuje $A[[c, B]] \in \mathbf{Assn}$ takové, že pro každé $I \in \mathcal{I}$ platí $A[[c, B]]^I = wp^I[[c, B]]$.

Důkaz. Je dobré si znovu uvědomit, že

$$A[[c, B]]^I = wp^I[[c, B]] \iff \forall \sigma \in \Sigma : (\sigma \models^I A[[c, B]] \iff \mathcal{C}_\perp[[c]]\sigma \models^I B).$$

Důkaz je veden indukcí ke struktuře c .

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

$$\begin{aligned} \sigma &\in wp^I[[\mathbf{skip}, B]] \\ &\iff \mathcal{C}_\perp[[\mathbf{skip}]]\sigma \models^I B \\ &\iff \sigma \models^I B \\ &\iff \sigma \models^I A[[\mathbf{skip}, B]]. \end{aligned}$$

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

$$\begin{aligned} \sigma &\in wp^I[[X := a, B]] \\ &\iff \sigma[\mathcal{A}[[a]]\sigma/X] \models^I B \end{aligned}$$

$$\Longleftrightarrow \sigma \models^I B[a/X] \text{ (užitím lematu 9)}$$

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

- $c \equiv c_0; c_1$. 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

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

$$\Longleftrightarrow \mathcal{C}_\perp[c_0; c_1]\sigma \models^I B$$

$$\Longleftrightarrow \mathcal{C}[c_0]\sigma \models^I A[c_1, B] \text{ (podle I.P.)}$$

$$\Longleftrightarrow \sigma \models^I A[c_0, A[c_1, B]] \text{ (podle I.P.)}$$

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

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

$$\sigma \in wp^I[c, B]$$

$$\Longleftrightarrow \mathcal{C}_\perp[c]\sigma \models^I B$$

$$\Longleftrightarrow (\mathcal{B}[b] = \text{true} \wedge \mathcal{C}_\perp[c_0]\sigma \models^I B) \vee (\mathcal{B}[b] = \text{false} \wedge \mathcal{C}_\perp[c_1]\sigma \models^I B)$$

$$\Longleftrightarrow (\sigma \models^I b \wedge \sigma \models^I A[c_0, B]) \vee (\sigma \models^I \neg b \wedge \sigma \models^I A[c_1, B]) \text{ podle I.P.}$$

$$\Longleftrightarrow \sigma \models^I (b \wedge A[c_0, B]) \vee (\neg b \wedge A[c_1, B]) \Longleftrightarrow \sigma \models^I A[c, B].$$

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

$\forall k \forall \sigma_0, \dots, \sigma_k \in \Sigma :$

$$(\sigma = \sigma_0 \quad \wedge \quad \forall i(0 \leq i < k) : (\sigma_i \models^I b \wedge C[[c_0]]\sigma_i = \sigma_{i+1})) \Rightarrow (\sigma_k \models^I b \vee B)(1)$$

Stačí tedy výše uvedené tvrzení „přeložit“ do **Assn**. K tomu využijeme následující pozorování:
 Buď $A \in \mathbf{Assn}$ takové, že všechny proměnné vyskytující se v A jsou mezi X_1, \dots, X_l (místo X_1, \dots, X_l budeme psát také \vec{X}). Pro každé $\sigma \in \Sigma$ lze nyní definovat l -tici \vec{s} , kde $\vec{s}(i) = \sigma(X_i)$.
 Pro každé $I \in \mathcal{I}$ platí

$$\sigma \models^I A \iff \models^I A[\vec{s}/\vec{X}] \quad (*)$$

což lze snadno ukázat strukturální indukcí. Nechť X_1, \dots, X_l jsou všechny proměnné, které se vyskytují v c a B . Nyní lze (1) přepsat na

$\forall k \forall \vec{s}_0, \dots, \vec{s}_k \in \mathbb{Z} :$

$$\begin{aligned} & (\sigma \models^I \vec{X} = \vec{s}_0 \quad \wedge \\ & \forall i(0 \leq i < k) : (\models^I b[\vec{s}_i/\vec{X}] \wedge \models^I (A[[c_0], \vec{X} = \vec{s}_{i+1}] \wedge \neg A[[c_0], \mathbf{false}])[\vec{s}_i/\vec{X}])) \\ & \Rightarrow \models^I (b \vee B)[\vec{s}_k/\vec{X}] \end{aligned} \quad (2)$$

Ukážeme, že (1) a (2) jsou ekvivalentní. K tomu je (zejména) třeba dokázat, že pokud σ_i resp. σ_{i+1} mají na \vec{X} hodnoty \vec{s}_i resp. \vec{s}_{i+1} a jinde jsou stejné, platí

$$C[[c_0]]\sigma_i = \sigma_{i+1} \iff \models^I (A[[c_0], \vec{X} = \vec{s}_{i+1}] \wedge \neg A[[c_0], \mathbf{false}])[\vec{s}_i/\vec{X}]$$

pro každé $I \in \mathcal{I}$. Snadno se vidí, že

$$\mathcal{C}[\![c_0]\!]\sigma_i = \sigma_{i+1} \iff \sigma_i \in wp^I[\![c_0, \vec{X} = \vec{s}_{i+1}]\!] \quad \wedge \quad \mathcal{C}[\![c_0]\!]\sigma_i \text{ je definováno.}$$

Podle I.P. $\sigma_i \in wp^I[\![c_0, \vec{X} = \vec{s}_{i+1}]\!] \iff \sigma_i \models^I A[\![c_0, \vec{X} = \vec{s}_{i+1}]\!]$. Dále $\mathcal{C}[\![c_0]\!]\sigma_i$ je definováno právě když $\sigma_i \models^I \neg A[\![c_0, \mathbf{false}]\!]$. Nyní již stačí aplikovat (*) a dostaneme tvar použitý v (2). Formulí (2) lze nyní transformovat na výraz jazyka **Assn** pomocí predikátu β^\pm . Nejprve „zakódujeme“ jednotlivá \vec{s}_i jako dvojice nezáporných celých čísel:

$$\begin{aligned} & \forall k, n_0, m_0, \dots, n_k, m_k \geq 0 : \\ & \quad (\bigwedge_{j=0}^l \beta^\pm(n_0, m_0, j, X_j) \quad \wedge \\ & \quad \forall i(0 \leq i < k) : ((\forall x_0, \dots, x_l : \bigwedge_{j=0}^l \beta^\pm(n_i, m_i, j, x_j) \Rightarrow b[\vec{x}/\vec{X}]) \quad \wedge \\ & \quad (\forall x_0, \dots, x_l, y_0, \dots, y_l : (\bigwedge_{j=0}^l \beta^\pm(n_i, m_i, j, x_j) \wedge \beta^\pm(n_{i+1}, m_{i+1}, j, y_j)) \\ & \quad \Rightarrow (A[\![c_0, \vec{X} = \vec{y}]\!] \wedge \neg A[\![c_0, \mathbf{false}]\!])[\vec{x}/\vec{X}])))) \\ & \Rightarrow (\forall x_0, \dots, x_l : \bigwedge_{j=0}^l \beta^\pm(n_k, m_k, j, x_j) \Rightarrow (b \vee B)[\vec{x}/\vec{X}]) \end{aligned} \quad (3)$$

Zbývá zakódovat posloupnost $n_0, m_0, \dots, n_k, m_k$ do jediné dvojice čísel. Získané tvrzení bude kvantifikováno takto:

$$\forall k, n, m \geq 0 :$$

Dále provedeme tato nahrazení:

★ $\bigwedge_{j=0}^l \beta^\pm(n_0, m_0, j, X_j)$ nahradíme výrazem

$$\forall p, q : (\beta^{\pm}(n, m, 0, p) \wedge \beta^{\pm}(n, m, 1, q)) \Rightarrow \bigwedge_{j=0}^l \beta^{\pm}(p, q, j, X_j)$$

★ $\forall x_0, \dots, x_l : \bigwedge_{j=0}^l \beta^{\pm}(n_i, m_i, j, x_j) \Rightarrow b[\vec{x}/\vec{X}]$ nahradíme výrazem

$$\forall p, q : (\beta^{\pm}(n, m, 2 * i, p) \wedge \beta^{\pm}(n, m, 2 * i + 1, q)) \Rightarrow (\forall x_0, \dots, x_l : \bigwedge_{j=0}^l \beta^{\pm}(p, q, j, x_j) \Rightarrow b[\vec{x}/\vec{X}])$$

★ $(\forall x_0, \dots, x_l, y_0, \dots, y_l : (\bigwedge_{j=0}^l \beta^{\pm}(n_i, m_i, j, x_j) \wedge \beta^{\pm}(n_{i+1}, m_{i+1}, j, y_j)))$ nahradíme výrazem

$$\forall p, q, u, v : (\beta^{\pm}(n, m, 2 * i, p) \wedge \beta^{\pm}(n, m, 2 * i + 1, q) \wedge \beta^{\pm}(n, m, 2 * i + 2, u) \wedge \beta^{\pm}(n, m, 2 * i + 3, v)) \Rightarrow (\forall x_0, \dots, x_l, y_0, \dots, y_l : (\bigwedge_{j=0}^l \beta^{\pm}(p, q, j, x_j) \wedge \beta^{\pm}(u, v, j, y_j)))$$

★ $\forall x_0, \dots, x_l : \bigwedge_{j=0}^l \beta^{\pm}(n_k, m_k, j, x_j) \Rightarrow (b \vee B)[\vec{x}/\vec{X}]$ nahradíme výrazem

$$\forall p, q : (\beta^{\pm}(n, m, 2 * k, p) \wedge \beta^{\pm}(n, m, 2 * k + 1, q)) \Rightarrow (\forall x_0, \dots, x_l : \bigwedge_{j=0}^l \beta^{\pm}(p, q, j, x_j) \Rightarrow (b \vee B)[\vec{x}/\vec{X}]).$$

Získaná formule již patří do **Assn**.

□

Úplnost Hoareova odvozovacího systému

Lema 14. *Necht' $c \in \mathbf{Com}$ a $B \in \mathbf{Assn}$. Dále necht' $A[[c, B]] \in \mathbf{Assn}$ vyjadřuje nejslabší vstupní podmínku pro c a B . Pak $\vdash \{A[[c, B]]\} c \{B\}$.*

Důkaz. Indukcí ke struktuře c (využijeme poznatků z důkazu **věty 13**).

- $c \equiv \mathbf{skip}$. $A[[\mathbf{skip}, B]]$ je ekvivalentní B , zejména tedy $\models A[[\mathbf{skip}, B]] \Rightarrow B$. Proto $\vdash \{A[[\mathbf{skip}, B]]\} \mathbf{skip} \{B\}$ (užitím axiomu pro \mathbf{skip} a pravidla důsledku).
- $c \equiv X := a$. Pak $A[[X := a, B]]$ je ekvivalentní $B[a/X]$. Užitím axiomu pro přiřazení a pravidla důsledku dostáváme $\vdash \{A[[X := a, B]]\} X := a \{B\}$.
- $c \equiv c_0; c_1$. Pak $A[[c_0; c_1, B]]$ je ekvivalentní $A[[c_0, A[[c_1, B]]]]$, kde $A[[c_1, B]]$ vyjadřuje nejslabší vstupní podmínku pro c_1 a B , a $A[[c_0, A[[c_1, B]]]]$ vyjadřuje nejslabší vstupní podmínku pro c_0 a $A[[c_1, B]]$. 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 $\vdash \{A[[c_0, A[[c_1, B]]]]\} c_0; c_1 \{B\}$, proto také $\vdash \{A[[c_0; c_1, B]]\} c_0; c_1 \{B\}$ užitím pravidla důsledku.
- $c \equiv \mathbf{if } b \mathbf{ then } c_0 \mathbf{ else } c_1$. Pak $A[[c, B]]$ je ekvivalentní $(b \wedge A[[c_0, B]]) \vee (\neg b \wedge A[[c_1, B]])$, kde $A[[c_0, B]]$ resp. $A[[c_1, B]]$ vyjadřuje nejslabší vstupní podmínku pro c_0 resp. c_1 a 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 $\vdash \{A[[c, B]]\} c \{B\}$.

• $c \equiv \text{while } b \text{ do } c_0$. Dokážeme, že

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

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

Vyjadřuje-li $A[c_0, A[c, B]]$ nejslabší vstupní podmínku pro c_0 a $A[c, B]$, pak podle I.P. $\vdash \{A[c_0, A[c, B]]\} c_0 \{A[c, B]\}$. Podle a) pak platí $\models (A[c, B] \wedge b) \Rightarrow A[c_0, A[c, B]]$, proto $\vdash \{A[c, B] \wedge b\} c_0 \{A[c, B]\}$ užitím pravidla důsledku. Dále podle pravidla pro **while** dostáváme $\vdash \{A[c, B]\} c \{A[c, B] \wedge \neg b\}$ a pomocí b) a pravidla důsledku 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. $\mathcal{C}_\perp[c]\sigma \models^I B$ a $\sigma \models^I b$. Denotační sémantika byla definována tak, že platí:

$$\mathcal{C}_\perp[c] = \mathcal{C}_\perp[\text{if } b \text{ then } c_0; c \text{ else skip}]$$

To znamená, že $\mathcal{C}_\perp[c_0; c]\sigma \models^I B$, tedy $\mathcal{C}_\perp[c](\mathcal{C}_\perp[c_0]\sigma) \models^I B$. Proto $\mathcal{C}_\perp[c_0]\sigma \models^I A[c, B]$, 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

$$\mathcal{C}_\perp[c] = \mathcal{C}_\perp[\text{if } b \text{ then } c_0; c \text{ else skip}],$$

dostáváme $\mathcal{C}_\perp[c]\sigma = \sigma$, proto $\sigma \models^I B$. Tedy $\models (A[c, B] \wedge \neg b) \Rightarrow B$.

□

Věta 15 (o úplnosti). Jestliže $\models \{A\} c \{B\}$, pak $\vdash \{A\} c \{B\}$.

Důkaz. Předpokládejme, že $\models \{A\} c \{B\}$. Podle **věty 13** 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 **lematu 14** platí $\vdash \{A[[c, B]]\} c \{B\}$ a tedy $\vdash \{A\} c \{B\}$ užitím pravidla důsledku. \square

Ekvivalence axiomatické a denotační sémantiky

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

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

Věta 16. Pro každé $c_0, c_1 \in \mathbf{Com}$ platí: $\mathcal{C}[\![c_0]\!] = \mathcal{C}[\![c_1]\!]$ $\iff c_0 \simeq c_1$

Důkaz.

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

$$\mathcal{C}_\perp[\![c_0]\!]\sigma \models^I B \iff \mathcal{C}_\perp[\![c_1]\!]\sigma \models^I B$$

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

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

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

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

kde n_Y je hodnota proměnné Y ve stavu σ . Dále označme $\sigma_0 = \mathcal{C}_\perp[\![c_0]\!]\sigma$ a $\sigma_1 = \mathcal{C}_\perp[\![c_1]\!]\sigma$. 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 $\models \{A\} c_0 \{\mathbf{false}\}$, zatímco $\not\models \{A\} c_1 \{\mathbf{false}\}$.
- $\sigma_1 = \perp$. Podobně.



Označované příkazy

- Platnost tvrzení $\{A\}c\{B\}$ o částečné korektnosti lze (efektivně) redukovat na platnost tvrzení $A \Rightarrow A[[c, B]]$, kde $A[[c, B]]$ vyjadřuje nejslabší vstupní podmínku pro c a B .
- Pokud bychom měli k dispozici program, který rozhoduje platnost tvrzení z **Assn**, bylo by možné algoritmicky ověřit i platnost tvrzení o částečné korektnosti.
- Takový program z principiálních důvodů neexistuje; existují ale „dokazovače vět“ (theorem provers), které umožňují efektivně dokázat platnost určitých podtříd platných tvrzení **Assn**.
- Tyto nástroje mohou být velmi užitečné, pokud se kombinují s lidskou inteligencí.
- Myšlenka: místo konstrukce nejslabší vstupní podmínky doplníme prvky **Assn** přímo do příkazu (programu), čímž vznikne **označovaný příkaz**. Pak algoritmicky sestavíme množinu tvrzení z **Assn** (tzv. **verifikačních podmínek**), jejichž platnost je postačující podmínkou pro korektnost daného označování.

- **Označované příkazy ACom**

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

kde $X \in \mathbf{Var}$, $a \in \mathbf{Aexp}$, $b \in \mathbf{Bexp}$, $D \in \mathbf{Assn}$ a \bar{c} je označovaný příkaz který není přiřazením.

- **Označované tvrzení o částečné korektnosti** je trojice tvaru $\{A\}c\{B\}$, kde $A, B \in \mathbf{Assn}$ a $c \in \mathbf{ACom}$. Toto tvrzení je **platné**, je-li platné tvrzení $\{A\}c'\{B\}$ kde $c' \in \mathbf{Com}$ vznikne z c vynecháním všech prvků **Assn**.

Verifikační podmínky

- Označkový **while**-cyklus

$$\{A\} \text{ while } b \text{ do } \{D\}c \{B\} \quad (*)$$

obsahuje $D \in \text{Assn}$, které by mělo být invariantem, což znamená že $\{D \wedge b\}c\{D\}$ je platné tvrzení. Pokud už víme, že D je invariant, stačí pro platnost $(*)$ dokázat platnost tvrzení $A \Rightarrow D$ a $D \wedge \neg b \Rightarrow B$.

- Množina **verifikačních podmínek** pro označkové tvrzení $\{A\}c\{B\}$, psáno $\mathcal{V}(\{A\}c\{B\})$, je definována induktivně takto:

$$\mathcal{V}(\{A\} \text{ skip } \{B\}) = \{A \Rightarrow B\}$$

$$\mathcal{V}(\{A\} X := a \{B\}) = \{A \Rightarrow B[a/X]\}$$

$$\mathcal{V}(\{A\} c_0; X := a \{B\}) = \mathcal{V}(\{A\} c_0 \{B[a/X]\})$$

$$\mathcal{V}(\{A\} c_0; \{D\}\bar{c} \{B\}) = \mathcal{V}(\{A\} c_0 \{D\}) \cup \mathcal{V}(\{D\} c_1 \{B\}) \text{ kde } \bar{c} \text{ není přiřazení}$$

$$\mathcal{V}(\{A\} \text{ if } b \text{ then } c_0 \text{ else } c_1 \{B\}) = \mathcal{V}(\{A \wedge b\} c_0 \{B\}) \cup \mathcal{V}(\{A \wedge \neg b\} c_1 \{B\})$$

$$\mathcal{V}(\{A\} \text{ while } b \text{ do } \{D\}c \{B\}) = \mathcal{V}(\{D \wedge b\}c\{D\}) \cup \{A \Rightarrow D\} \cup \{D \wedge \neg b \Rightarrow B\}$$

Věta 17. Necht' $\{A\} c \{B\}$ je označované tvrzení o částečné korektnosti. Jestliže všechny prvky $\mathcal{V}(\{A\} c \{B\})$ jsou platná tvrzení, je i $\{A\} c \{B\}$ platné.

Důkaz. Indukcí ke struktuře c . □

Příklad:

- $\{\text{true}\} \text{ while false do } \{\text{false}\} \text{ skip } \{\text{true}\}$
- Toto označované tvrzení o částečné korektnosti je **platné**.
- $\mathcal{V}(\{\text{true}\} \text{ while false do } \{\text{false}\} \text{ skip } \{\text{true}\}) =$
 $\mathcal{V}(\{\text{false} \wedge \text{false}\} \text{ skip } \{\text{false}\}) \cup \{\text{true} \Rightarrow \text{false}\} \cup \{(\text{false} \wedge \neg \text{false}) \Rightarrow \text{true}\}$
- $\text{true} \Rightarrow \text{false} \in \mathcal{V}(\{\text{true}\} \text{ while false do } \{\text{false}\} \text{ skip } \{\text{true}\})$ platné **není**; platnost prvků $\mathcal{V}(\{A\} c \{B\})$ je tedy pouze **postačující**, nikoliv **nutná** podmínka platnosti $\{A\} c \{B\}$.

Operační sémantika paralelních programů

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

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

- SOS sémantiku II. typu rozšíříme o pravidla

$$\frac{\langle c_0, \sigma \rangle \mapsto \langle c'_0, \sigma' \rangle}{\langle c_0 \parallel c_1, \sigma \rangle \mapsto \langle c'_0 \parallel c_1, \sigma' \rangle} \qquad \frac{\langle c_1, \sigma \rangle \mapsto \langle c'_1, \sigma' \rangle}{\langle c_0 \parallel c_1, \sigma \rangle \mapsto \langle c_0 \parallel c'_1, \sigma' \rangle}$$

- Podle stávajících pravidel např.

$$\begin{aligned} \langle (X := 1 \parallel X := 2); X := 3, \sigma \rangle &\mapsto \\ \langle (\text{skip} \parallel X := 2); X := 3, \sigma[1/X] \rangle &\mapsto \\ \langle (\text{skip} \parallel \text{skip}); X := 3, \sigma[2/X] \rangle \end{aligned}$$

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

- Zavedeme predikát **IsSkip** předpisem

$$\mathbf{IsSkip}(\mathbf{skip}) = \mathbf{true}$$

$$\mathbf{IsSkip}(X := a) = \mathbf{false}$$

$$\mathbf{IsSkip}(c_0; c_1) = \mathbf{IsSkip}(c_0) \wedge \mathbf{IsSkip}(c_1)$$

$$\mathbf{IsSkip}(\mathbf{if } b \mathbf{ then } c_0 \mathbf{ else } c_1) = \mathbf{false}$$

$$\mathbf{IsSkip}(\mathbf{while } b \mathbf{ do } c) = \mathbf{false}$$

$$\mathbf{IsSkip}(c_0 \parallel c_1) = \mathbf{IsSkip}(c_0) \wedge \mathbf{IsSkip}(c_1)$$

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

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

- Nyní je již odvoditelný přechod $\langle (\mathbf{skip} \parallel \mathbf{skip}); X := 3, \sigma[2/X] \rangle \mapsto \langle X := 3, \sigma[2/X] \rangle$
- 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.

„Obecná“ definice LTL

Syntaxe LTL

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

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

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 $w = \emptyset\{p\}\emptyset\{q\}\{q\}\{p, q\} \dots$, platí $w(2) = \emptyset$, $w(3) = \{q\}$, $w_2 = \emptyset\{q\}\{q\}\{p, q\} \dots$

Sémantika LTL

- Buď φ LTL formule. **Platnost** φ pro dané $w \in \Sigma_\varphi^\omega$ je definována indukcí ke struktuře φ :

$$w \models \mathbf{true}$$

$$w \models p \iff p \in w(0)$$

$$w \models \neg \varphi \iff w \not\models \varphi$$

$$w \models \varphi_1 \wedge \varphi_2 \iff w \models \varphi_1 \quad \wedge \quad w \models \varphi_2$$

$$w \models \mathcal{X}\varphi \iff w_1 \models \varphi$$

$$w \models \varphi_1 \mathcal{U} \varphi_2 \iff \exists j : w_j \models \varphi_2 \quad \wedge \quad \forall i < j : w_i \models \varphi_1$$

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

LTL jako jazyk vlastností paralelních a neukončených programů

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

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

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 $\alpha_\varphi^I \in \Sigma_\varphi^\omega$, kde $\alpha_\varphi^I(i) = \{A \in \mathbf{Assn}(\varphi) \mid \sigma_i \models^I A\}$.
- Běh α splňuje** LTL formuli φ při interpretaci I , psáno $\alpha \models^I \varphi$, jestliže $\alpha_\varphi^I \models \varphi$.
- 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 $\alpha \models^I \varphi$.
- Množina vlastností **běhů** vyjádřitelných v LTL je uzavřená na negaci. Množina vlastností **konfigurací** vyjádřitelných v LTL na negaci uzavřená **není** (v důsledku „vestavěné“ univerzální kvantifikace přes běhy)!

Příklady:

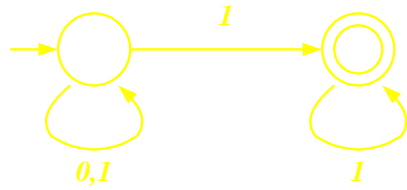
- $c \equiv \text{while tt do } X := X + 1.$ Pak $\langle c, \sigma \rangle \models^I \mathcal{F}(X > 5)$ pro každé σ a I .
- $c \equiv X := 2.$ Pak $\langle c, \sigma \rangle \models^I \mathcal{G}(X = 3)$ pro každé σ a I , neboť neexistuje žádný běh začínající v $\langle c, \sigma \rangle$.
- $c \equiv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

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 18. *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.*

Důkaz. Nechť $L_1 = L(\mathcal{A}_1)$ a $L_2 = L(\mathcal{A}_2)$, kde $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ a $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$. Bez újmy na obecnosti můžeme předpokládat, že $Q_1 \cap Q_2 = \emptyset$. Zřejmě $L_1 \cup L_2 = L(\mathcal{A}^\cup)$, kde

$$\mathcal{A}^\cup = (Q_1 \cup Q_2 \cup \{q_0\}, \Sigma, \delta_1 \cup \delta_2 \cup \{(q_0, a, q) \mid q_1 \xrightarrow{a} q \vee q_2 \xrightarrow{a} q\}, q_0, F_1 \cup F_2)$$

kde $q_0 \notin Q_1 \cup Q_2$. Dále $L_1 \cap L_2 = L(\mathcal{A}^\cap)$, kde

$$\mathcal{A}^\cap = (Q_1 \times Q_2 \times \{1, 2\}, \Sigma, \delta^\cap, (q_1, q_2, 1), F_1 \times Q_2 \times \{1\})$$

a δ^\cap je určena předpisem

- $(p, q, 1) \xrightarrow{a} (p', q', 1)$, jestliže $p \xrightarrow{a} p', q \xrightarrow{a} q'$ a $p \notin F_1$;
- $(p, q, 1) \xrightarrow{a} (p', q', 2)$, jestliže $p \xrightarrow{a} p', q \xrightarrow{a} q'$ a $p \in F_1$;
- $(p, q, 2) \xrightarrow{a} (p', q', 2)$, jestliže $p \xrightarrow{a} p', q \xrightarrow{a} q'$ a $q \notin F_2$;
- $(p, q, 2) \xrightarrow{a} (p', q', 1)$, jestliže $p \xrightarrow{a} p', q \xrightarrow{a} q'$ a $q \in F_2$.

□

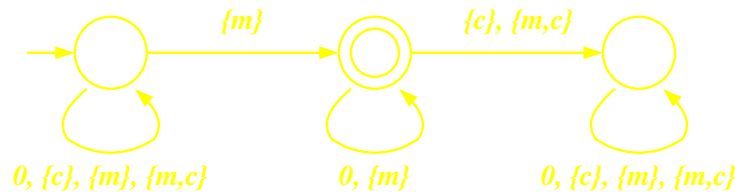
Věta 19. *Nechť \mathcal{A} je Büchiho automat. Problém, zda $L(\mathcal{A}) = \emptyset$ je rozhodnutelný (v polynomiálním čase).*

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). \square

Vztah LTL a Büchiho automatů

Věta 20. Bud' φ LTL formule. Jazyk L_φ je ω -regulární a lze algoritmicky sestavit Büchiho automat \mathcal{A}_φ velikosti $\mathcal{O}(2^{|\varphi|})$, který akceptuje L_φ .

Příklad: Uvažme formuli $\varphi = \mathcal{F}(m \wedge \mathcal{G}\neg c)$. Pak jazyk L_φ (nad abecedou $\Sigma_\varphi = \{\emptyset, \{m\}, \{c\}, \{m, c\}\}$) je akceptován Büchiho automatem



Ověřování platnosti LTL formulí

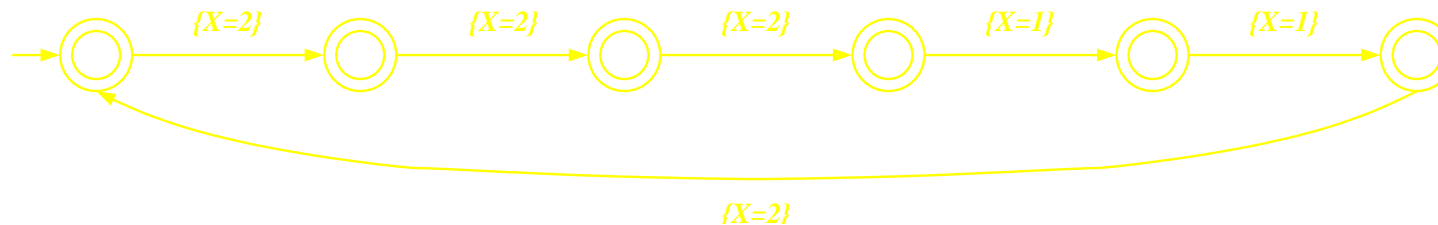
- Bud' $c \in \mathbf{PCom}$ a $\sigma \in \Sigma$ takové, že přechodový systém $(S, \{\tau\}, \mapsto)$, kde $S = \{\langle c', \sigma' \rangle \mid \langle c, \sigma \rangle \rightarrow^* \langle c', \sigma' \rangle\}$, má **konečně** mnoho konfigurací.
- Necht' φ je LTL formule a I interpretace. Definujeme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

Příklad:

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

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

Automat \mathcal{A} tedy vypadá následovně:



- Dále $\neg\varphi \equiv \mathcal{F}(X=1 \wedge \mathcal{G}(\neg(X=2)))$, lze tedy použít (mírně modifikovaný) automat z předchozího příkladu.
- Nyní stačí sestavit automat \mathcal{A}^\cap z důkazu **věty 18** a ověřit, zda $L(\mathcal{A}^\cap) = \emptyset$ (což lze rovněž provést algoritmicky podle **věty 19**).