

Sémantique dénotationnelle

... d'un langage impératif jouet

IMP: syntaxe

Commandes

$c ::=$	$x := e$	affectation
	skip	ne rien faire
	$c_1; c_2$	séquence
	if e then c_1 else c_2	conditionnelle
	while e do c	boucle while

Expressions

$e ::=$	x	variables
	\dot{n}	constante entière ($n \in \mathbb{Z}$)
	$e + e$	addition
	$\dot{-}e$	opposé

Sémantique dénotationnelle

$$\llbracket c \rrbracket : \rho \in \text{Env} \rightarrow \text{Env}_\perp$$

$$\text{Env} = \text{Var} \rightarrow \mathbb{Z}$$

$$\llbracket x := e \rrbracket \rho = \rho[x \mapsto \llbracket e \rrbracket \rho]$$

$$\llbracket \text{skip} \rrbracket \rho = \rho$$

$$\llbracket c_1; c_2 \rrbracket \rho = \begin{cases} \perp & \text{si } \llbracket c_1 \rrbracket \rho = \perp \\ \llbracket c_2 \rrbracket (\llbracket c_1 \rrbracket \rho) & \text{sinon} \end{cases}$$

$$\llbracket \text{if } e \text{ then } c_1 \text{ else } c_2 \rrbracket \rho = \begin{cases} \llbracket c_1 \rrbracket \rho & \text{si } \llbracket e \rrbracket \rho \neq 0 \\ \llbracket c_2 \rrbracket \rho & \text{si } \llbracket e \rrbracket \rho = 0 \end{cases}$$

$$\llbracket \text{while } e \text{ do } c \rrbracket \rho = \text{lfp}(F_{e,c})(\rho)$$

où:

$$F_{e,c}: [\text{Env} \rightarrow \text{Env}_\perp] \rightarrow [\text{Env} \rightarrow \text{Env}_\perp]$$

$$F_{e,c}(f)(\rho) = \begin{cases} \rho & \text{si } \llbracket e \rrbracket \rho = 0 \\ \perp & \text{si } \llbracket e \rrbracket \rho \neq 0 \text{ et } \llbracket c \rrbracket \rho = \perp \\ f(\llbracket c \rrbracket \rho) & \text{si } \llbracket e \rrbracket \rho \neq 0 \text{ et } \llbracket c \rrbracket \rho \neq \perp \end{cases}$$

Bonne définition

- **Thm:** $\llbracket c \rrbracket \rho$ est bien définie, et la fonctionnelle $F_{e,c}$ est continue.
- Preuve: par récurrence structurelle sur c .
Le point intéressant est la continuité de $F_{e,c}$.

Sémantique opérationnelle à grands pas (rappel)

$$\frac{}{\rho \vdash x := e \Rightarrow \rho[x \mapsto \llbracket e \rrbracket \rho]} (:=) \quad \frac{}{\rho \vdash \text{skip} \Rightarrow \rho} (\text{skip})$$

$$\frac{\rho \vdash c_1 \Rightarrow \rho' \quad \rho' \vdash c_2 \Rightarrow \rho''}{\rho \vdash c_1; c_2 \Rightarrow \rho''} (\text{Seq})$$

$$\frac{\rho \vdash c_1 \Rightarrow \rho'}{\rho \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 \Rightarrow \rho' \quad \text{si } \llbracket e \rrbracket \rho \neq 0} (\text{if}_1)$$

$$\frac{\rho \vdash c_2 \Rightarrow \rho'}{\rho \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 \Rightarrow \rho' \quad \text{si } \llbracket e \rrbracket \rho = 0} (\text{if}_2)$$

$$\frac{\rho \vdash c \Rightarrow \rho' \quad \rho' \vdash \text{while } e \text{ do } c \Rightarrow \rho''}{\rho \vdash \text{while } e \text{ do } c \Rightarrow \rho'' \quad \text{si } \llbracket e \rrbracket \rho \neq 0} (\text{while})$$

$$\frac{}{\rho \vdash \text{while } e \text{ do } c \Rightarrow \rho \quad \text{si } \llbracket e \rrbracket \rho = 0} (\text{while}_{fin})$$

Correction

- **Thm:** si $\rho \vdash c \Rightarrow \rho_\infty$ est dérivable [grand pas] alors $\llbracket c \rrbracket \rho = \rho_\infty$.
- Preuve: par récurrence structurelle sur la dérivation...
- **Note:** Ceci implique $\llbracket c \rrbracket \rho \neq \perp$.

Adéquation

- Dans l'autre sens:
- **Thm:** si $\llbracket c \rrbracket \rho = \rho_\infty \neq \perp$ alors $\rho \vdash c \Rightarrow \rho_\infty$ est dérivable [grand pas].
- Preuve: par récurrence structurelle sur c .
- L'unique difficulté est le cas des boucles `while...`

Correction + adéquation

- **Thm:** Les deux propriétés suivantes sont équivalentes:
 - $\rho \vdash c \Rightarrow \rho_\infty$ est dérivable [grand pas]
 - $(c . \varepsilon, \rho) \rightarrow^* (\varepsilon, \rho_\infty)$ [petits pas]
 - $\llbracket c \rrbracket \rho = \rho_\infty \neq \perp$ [dénotationnelle].