

Laboratorio Lenguajes y Compiladores

Mansilla, Kevin Gaston*

April 24, 2025

Implementación de la semántica denotacional para el lenguaje imperativo simple con fallas (LIS + Fallas).

I Semántica del lenguaje

Se agregan excepciones al lenguaje imperativo simple.

$$\langle comm \rangle ::= fail \mid catchin \langle comm \rangle with \langle comm \rangle$$

Sea $\Sigma' = \Sigma \cup \{abort\}$ con orden discreto. La guncion semántica ahora es:

$$\llbracket _ \rrbracket \in \langle comm \rangle \rightarrow \Sigma \rightarrow \Sigma'_\perp$$

Las ecuaciones semánticas son las siguiente:

$$\begin{aligned} \llbracket Skip \rrbracket_\sigma &= \sigma \\ \llbracket Fail \rrbracket_\sigma &= \langle abort, \sigma \rangle \\ \llbracket v := e \rrbracket_\sigma &= [\sigma \mid v : \llbracket e \rrbracket_\sigma] \\ \llbracket if \ b \ then \ c_0 \ else \ c_1 \rrbracket_\sigma &= \begin{cases} \llbracket c_0 \rrbracket_\sigma & \text{si } \llbracket b \rrbracket_\sigma \\ \llbracket c_1 \rrbracket_\sigma & \text{si } \neg \llbracket b \rrbracket_\sigma \end{cases} \end{aligned}$$

Operadores de transferencia de control. Dada $f \in \Sigma \rightarrow \Sigma'_\perp$, denotamos por f_* la siguiente funcion

$$\begin{aligned} f_* &\in \Sigma'_\perp \rightarrow \Sigma'_\perp \\ f_*x &= \begin{cases} f\sigma & \text{si } x = \sigma \in \Sigma \\ x & \text{si no} \end{cases} \end{aligned}$$

En este caso, la presencia de una situación obortiva determina que no se transfiere el control a f Servira para describir el significado de $c_0; c_1$ ya que si ocurre una situacion de excepción al ejecutar c_0 , en control no es transferido a c_1 .

Por otro lado, dado $f \in \Sigma \rightarrow \Sigma'_\perp$, denotamos f_+ a la siguiente extension de f a Σ'_\perp :

$$\begin{aligned} f_+ &\in \Sigma'_\perp \rightarrow \Sigma'_\perp \\ f_+x &= \begin{cases} f\sigma & \text{si } x = \langle abort, \sigma \rangle \in \{abort\} \times \Sigma \\ x & \text{si no} \end{cases} \end{aligned}$$

*kevin.mansilla@mi.unc.edu.ar

En una clara dualidad con la definición de f_* , la definición de f_+ determina lo contrario: se transfiere el control a f sólo en caso de excepción. Esto corresponderá a *catchin c with c'*.

Finalmente, dada $f \in \Sigma \rightarrow \Sigma$, denotamos por f_{\dagger} la siguiente extension de f a Σ'_{\perp} :

$$f_{\dagger} \in \Sigma'_{\perp} \rightarrow \Sigma'_{\perp}$$

$$f_{\dagger}x = \begin{cases} \langle abort, f\sigma \rangle & \text{si } x = \langle abort, \sigma \rangle \\ fx & \text{si } x \in \Sigma \\ \perp & \text{si no} \end{cases}$$

Nose que aui hay una transferencia de control a f en cualquier situacion (abortiva o no). Servirá para restaurar el valor de las variables locales. Entonces las demas funcione semanticas

$$\begin{aligned} \llbracket c_0; c_1 \rrbracket_{\sigma} &= \llbracket c_1 \rrbracket_* (\llbracket c_0 \rrbracket_{\sigma}) \\ \llbracket catchin c_0 with c_1 \rrbracket_{\sigma} &= \llbracket c_1 \rrbracket_+ (\llbracket c_0 \rrbracket_{\sigma}) \\ \llbracket newvar v := e in c \rrbracket_{\sigma} &= (\lambda \sigma' \in \Sigma. [\sigma' \mid v : \sigma v])_{\dagger} (\llbracket c \rrbracket [\sigma \mid v : \llbracket e \rrbracket_{\sigma}]) \\ \llbracket while b do c \rrbracket &= \sqcup_{i=0}^{\infty} F^i \perp_{\Sigma \rightarrow \Sigma_{\perp}} \\ &\text{donde} \\ Fw\sigma &= \begin{cases} w_*(\llbracket c \rrbracket_{\sigma}) & \text{si } \llbracket b \rrbracket_{\sigma} \\ \sigma & \text{si } \neg \llbracket b \rrbracket_{\sigma} \end{cases} \end{aligned}$$