Application: Garbage Gollection

Stefan Muller

CS 534: Types and Programming Languages, Spring 2024
Lecture 8

1 Variable Declarations

Previously, we didn't have any way for variables to come into being; they had to just be in the state when we started running the program. It's helpful to have a way to *declare* variables.

$$\text{Statements} \quad s \quad ::= \ldots \mid \mathsf{decl}\ x = e\ \mathsf{in}$$

$$\frac{x \notin \Gamma \qquad \Gamma \vdash e : \tau \qquad \Gamma, x : \tau \vdash s\ \mathsf{ok}}{\Gamma \vdash \mathsf{decl}\ x = e\ \mathsf{in}\ s\ \mathsf{ok}}\ (\text{T-DECL}) \qquad \frac{e \mapsto_\sigma e'}{\langle \mathsf{decl}\ x = e\ \mathsf{in}\ s, \sigma\rangle \mapsto \langle \mathsf{decl}\ x = e'\ \mathsf{in}\ s, \sigma\rangle}\ (\text{S-DECL1})$$

$$\frac{e\ \mathsf{val}}{\langle \mathsf{decl}\ x = e\ \mathsf{in}\ s, \sigma\rangle \mapsto \langle s, \sigma[x \mapsto e]\rangle}\ (\text{S-DECL2})$$

Let's look at the preservation case for (S-DECL2). Consider the following step:

$$\langle \mathsf{decl}\ x = \overline{10}\ \mathsf{in}\ x := x + \overline{1} \emptyset, \mapsto \rangle \langle x := x + \overline{1}, \{x = 10\}\rangle$$

We have $\cdot \vdash \mathsf{decl}\ x = \overline{10}\ \mathsf{in}\ x := x + \overline{1}\ \mathsf{ok}$ and $\cdot \vdash \emptyset$. By preservation, we should have $\cdot \vdash x := x + \overline{1}\ \mathsf{ok}$. But this isn't true. We need to add x to the context while the program is running!

Lemma 1 (Preservation)**.** *If* $\Gamma \vdash s\ \mathsf{ok}$ *and* $\Gamma \vdash \sigma$ *and* $\langle s, \sigma\rangle \mapsto \langle s', \sigma'\rangle$ *then* ***there exists*** Γ' *such that* $\Gamma' \vdash s'\ \mathsf{ok}$ *and* $\Gamma' \vdash \sigma'$

Proof. By induction on the derivation of $\langle s, \sigma\rangle \mapsto \langle s', \sigma'\rangle$. The existing cases don't need to change, they just set $\Gamma' = \Gamma$.

- (S-DECL1) Basically the same as the search rule for $x := e$.
- (S-DECL2). Then $s = \mathsf{decl}\ x = e\ \mathsf{in}\ s'$ and $\sigma' = \sigma[x \mapsto e]$. By inversion, $x \notin \Gamma$ and $\Gamma \vdash e : \tau$ and $\Gamma, x : \tau \vdash s'\ \mathsf{ok}$. Let $\Gamma' = \Gamma, x : \tau$. Because $x \notin \Gamma$, we can apply weakening to all of the values in σ' (including e) and we get that for all $y : \tau' \in \Gamma'$, $\Gamma' \vdash \sigma(y) : \tau'$.

□

2 Garbage

Consider the following program.

$$
\begin{array}{ll}
\mathsf{decl}\ n = \overline{10}\ \mathsf{in}\ ; & \sigma_1 = \emptyset \\
\mathsf{decl}\ x = \overline{0}\ \mathsf{in}\ ; & \\
\mathsf{decl}\ r = \overline{0}\ \mathsf{in}\ ; & \sigma_2 = \{n = 10, x = 0, r = 0\} \\
\mathsf{while}\ x < n\ \mathsf{do} & \\
\quad x := x + \overline{1}; \quad r := r + \overline{1} & \\
\mathsf{od}; & \\
\mathsf{decl}\ y = \overline{0}\ \mathsf{in}\ ; & \sigma_3 = \{n = 10, x = 10, r = 10, y = 0\} \\
\mathsf{while}\ y < n\ \mathsf{do} & \\
\quad y := y + \overline{1}; & \\
\quad r := r + \overline{2} & \\
\mathsf{od} &
\end{array}
$$

By the time we declare y, we know that x is never used again. We want to be able to get rid of it and free up the memory. In many languages, this is the purpose of garbage collection (GC).

We can model GC with a new judgment that either has the whole program take a step or applies GC to it.

$$
\frac{\langle s, \sigma\rangle \mapsto \langle s', \sigma'\rangle}{\langle s, \sigma\rangle \Rightarrow \langle s', \sigma'\rangle}\ (\text{S-STEP}) \qquad \frac{\sigma' = \{x = v \mid \sigma(x) = v \wedge x \in \mathit{Vars}(s)\}}{\langle s, \sigma\rangle \Rightarrow \langle s, \sigma'\rangle}\ (\text{S-GC})
$$

where

$$
\begin{array}{rcl}
\mathit{Vars}(x) & \triangleq & \{x\} \\
\mathit{Vars}(\overline{n}),\ \mathit{Vars}(\text{“}s\text{”}) & \triangleq & \emptyset \\
\mathit{Vars}(e_1 + e_2),\ \mathit{Vars}(e_1\ \hat{}\ e_2) & \triangleq & \mathit{Vars}(e_1) \cup \mathit{Vars}(e_2) \\
\mathit{Vars}(|e|) & \triangleq & \mathit{Vars}(e) \\
\mathit{Vars}(\mathsf{skip}) & \triangleq & \emptyset \\
\mathit{Vars}(x := e) & \triangleq & \mathit{Vars}(e) \cup \{x\} \\
\mathit{Vars}(\mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2\ \mathsf{fi}) & \triangleq & \mathit{Vars}(e) \cup \mathit{Vars}(s_1) \cup \mathit{Vars}(s_2) \\
\mathit{Vars}(\mathsf{while}\ e\ \mathsf{do}\ s\ \mathsf{od}) & \triangleq & \mathit{Vars}(e) \cup \mathit{Vars}(s) \\
\mathit{Vars}(s_1; s_2) & \triangleq & \mathit{Vars}(s_1) \cup \mathit{Vars}(s_2)
\end{array}
$$

Q: Why do we need a new judgment, rather than just making (S-GC) a new rule for the $\mapsto$ judgment?

$$
\begin{array}{ll}
 & \langle \mathsf{decl}\ x = \overline{1}\ \mathsf{in}\ ; \mathsf{decl}\ y = x + \overline{1}\ \mathsf{in}\ ; y := y + \overline{1}, \emptyset\rangle \\
\Rightarrow^2 & \langle \mathsf{decl}\ y = x + \overline{1}\ \mathsf{in}\ ; y := y + \overline{1}, \{x = 1\}\rangle \\
\Rightarrow^2 & \langle y := y + \overline{1}, \{x = 1, y = 2\}\rangle \\
\Rightarrow & \langle y := y + \overline{1}, \{y = 2\}\rangle \\
\Rightarrow & \langle \mathsf{skip}, \{y = 3\}\rangle
\end{array}
$$

Lemma 2. *1. If $\Gamma \vdash e : \tau$ and for all $x \in \mathit{Vars}(e)$, we have $\Gamma'(x) = \Gamma(x)$, then $\Gamma' \vdash e : \tau$.*

2. If $\Gamma \vdash s\ \mathsf{ok}$ and for all $x \in \mathit{Vars}(e)$, we have $\Gamma'(x) = \Gamma(x)$, then $\Gamma' \vdash s\ \mathsf{ok}$.

Proof. 1. By induction on the derivation of $\Gamma \vdash e : \tau$.

2. By induction on the derivation of $\Gamma \vdash s\ \mathsf{ok}$.

□

Lemma 3 (Preservation for $\Rightarrow$)**.** *If $\Gamma \vdash s\ \mathsf{ok}$ and $\Gamma \vdash \sigma$ and $\langle s, \sigma\rangle \Rightarrow \langle s', \sigma'\rangle$ then **there exists** Γ' such that $\Gamma' \vdash s'\ \mathsf{ok}$ and $\Gamma' \vdash \sigma'$*

Proof. By induction on the derivation of $\langle s, \sigma\rangle \Rightarrow \langle s', \sigma'\rangle$.

- (S-STEP). Then $\langle s, \sigma\rangle \mapsto \langle s', \sigma'\rangle$. The result follows by preservation for $\mapsto$.
- (S-GC). Then $s' = s$ and $\sigma' = \{x = v \mid \sigma(x) = v \wedge x \in \mathit{Vars}(s)\}$. Let $\Gamma' = \{x : \Gamma(x) \mid x \in \mathit{Vars}(s)\}$. By Lemma 2, $\Gamma' \vdash s\ \mathsf{ok}$. Let $x : \tau \in \Gamma'$. We have $x = \sigma(x) \in \sigma'$ and, by assumption, $\cdot \vdash \sigma(x) : \tau$.

□