CMSC 22610 **Implementation** **Handout 5**
Winter 2004 **of** **February 17**
Computer Languages

Mini-Lua variable-binding semantics

This handout formalizes the rules for variable binding in Mini-Lua programs. We use a stripped-down abstract syntax for Mini-Lua programs, which only includes variable and function definitions, blocks, and variable uses (we use s for statements, e, for expressions, and f and x for variables).

$$
\begin{array}{rcl}
s & ::= & s_1 \texttt{;}\ s_2 \\
& | & x \texttt{ = } e \\
& | & \texttt{local}\ x \texttt{ = } e \\
& | & \texttt{do}\ s\ \texttt{end} \\
& | & \texttt{function}\ f\ \texttt{(}x_1, \ldots, x_n\texttt{)}\ s \\
& | & \texttt{local function}\ f\ \texttt{(}x_1, \ldots, x_n\texttt{)}\ s \\
& | & \texttt{if}\ e\ \texttt{then}\ e_1\ \texttt{else}\ e_2 \\
\\
e & ::= & x
\end{array}
$$

Our "typing" judgements work on sets of globals (G) and environments (E), which are defined as follows:

$$
\begin{array}{rcl}
G & \in & 2^{\mathrm{Var}} \\
E & \in & \mathrm{Env} \quad = \quad \mathrm{Var} \xrightarrow{\text{fin}} \{\mathbf{local}, \mathbf{glob}\}
\end{array}
$$

The judgement forms are $E \vdash s : \langle E', G\rangle$, which means that under environment E, the statement s defines the envirnoment E' and the set of globals G, and $E \vdash e\ \mathbf{Ok}$, which means that the variables used in e are defined in E.

For statement sequencing, we use the environment from the first statement to check the second and union the set of globals.

$$\frac{E_0 \vdash s_1 : \langle E_1, G_1\rangle \quad E_1 \vdash s_2 : \langle E_2, G_2\rangle}{E_0 \vdash s_1 \texttt{;}\ s_2 : \langle E_2, G_1 \cup G_2\rangle}$$

A definition of a global variable extends the environment, assuming that the right-hand side is okay, as well as adding to the set of globals.

$$\frac{E \vdash e\ \mathbf{Ok}}{E \vdash x \texttt{ = } e : \langle E \pm \{x \mapsto \mathbf{glob}\}, \{x\}\rangle}$$

A definition of a local variable also extends the environment, assuming that the right-hand side is okay.

$$\frac{E \vdash e\ \mathbf{Ok}}{E \vdash \texttt{local}\ x \texttt{ = } e : \langle E \pm \{x \mapsto \mathbf{local}\}, \{\}\rangle}$$

A block localizes the environment (*i.e.*, definitions do not escape), but note that the set of defined globals does escape.

$$\frac{E \vdash s : \langle E', G \rangle}{E \vdash \textbf{do}\ s\ \textbf{end} : \langle E, G \rangle}$$

Like a block, a function definition localizes the environment generated by its body. Note that the body is checked in an environment that includes the function name itself.

$$\frac{E' = E \pm \{f \mapsto \mathbf{glob}\} \quad E'' = E' \pm \{x_1 \mapsto \mathbf{local}, \ldots, x_n \mapsto \mathbf{local}\} \quad E'' \vdash s : \langle E''', G \rangle}{E \vdash \texttt{function}\ f\ \texttt{(}x_1, \ldots, x_n\texttt{)}\ s : \langle E', G \rangle}$$

Local functions are similar to global functions.

$$\frac{E' = E \pm \{f \mapsto \mathbf{local}\} \quad E'' = E' \pm \{x_1 \mapsto \mathbf{local}, \ldots, x_n \mapsto \mathbf{local}\} \quad E'' \vdash s : \langle E''', G \rangle}{E \vdash \texttt{local function}\ f\ \texttt{(}x_1, \ldots, x_n\texttt{)}\ s : \langle E', G \rangle}$$

The conditional statement also localizes any definitions in its arms.

$$\frac{E \vdash e\ \mathbf{Ok} \quad E \vdash e_1 : \langle E_1, G_1 \rangle \quad E \vdash e_2 : \langle E_2, G_2 \rangle}{E \vdash \textbf{if}\ e\ \textbf{then}\ e_1\ \textbf{else}\ e_2 : \langle E, G_1 \cup G_2 \rangle}$$

Lastly, an expression is okay if its variables have been defined.

$$\frac{x \in \mathrm{dom}(E)}{E \vdash x\ \mathbf{Ok}}$$