Compiladores 2024-2
Facultad de Ciencias, UNAM
Práctica 6: Sistema de tipos.

Lourdes del Carmen Gonzáles Huesca Juan Alfonso Garduño Solís
Fausto David Hernández Jasso Juan Pablo Yamamoto Zazueta

"Un sistema de tipos es un método sintáctico para demostrar la ausencia de ciertos comportamientos al clasificar frases de acuerdo a los tipos de valores que calculan. Este método consiste en un conjunto de reglas para asociar tipos a contrucciones o expresiones del lenguaje."

Preeliminares.

El objetivo de esta práctica es que a partir de una tabla de símbolos de un programa seamos capaces de verificar que cada expresión que compone dicho programa este correctamente tipada/tipificada. Para lograrlo es necesario conocer qué variables y funciones están en el alcance, este es representado por el contexto de tipificado Γ, el cual mapea nombres x de variables a tipos σ.
Para nuestro caso tenemos que σ:

$$\begin{array}{lll}
\tau ::= \texttt{int} & T ::= \tau & \sigma ::= \texttt{var}\ \tau \\
\quad \mid \texttt{boolean} & \quad \mid \texttt{unit} & \quad \mid \texttt{fn}\ T \to T' \\
\quad \mid \texttt{int[]} & \quad \mid (\tau_1, ..., \tau_n) \text{ para } n \geq 2 & \\
\quad \mid \texttt{boolean[]} & &
\end{array}$$

Expresiones.

El juicio $\Gamma \vdash e : T$ es la regla para tipificar una expresión; significa que a partir del contexto Γ podemos deducir que e tiene tipo T. A continuacion utilizamos la metabariable x y f para representar identificadores arbitrarios de variables y funciones respectivamente, n como literal entera y `unit` como equivalente del `void` en Java para enunciar los jucios de tipificado de **expresiones** para nuestro lenguaje de programación.

$$\frac{}{\Gamma \vdash n : \texttt{int}} \qquad \frac{}{\Gamma \vdash true : \texttt{bool}} \qquad \frac{}{\Gamma \vdash false : \texttt{bool}}$$

$$\frac{\Gamma(x) = \texttt{var}\ \tau}{\Gamma \vdash x : \tau} \qquad \frac{\Gamma \vdash e : \texttt{int}}{\Gamma \vdash -e : \texttt{int}} \qquad \frac{\Gamma \vdash e : \texttt{bool}}{\Gamma \vdash !e : \texttt{bool}}$$

$$\frac{\Gamma \vdash e : \tau\texttt{[]}}{\Gamma \vdash \texttt{length}(e) : \texttt{int}} \qquad \frac{\Gamma \vdash e_1 : \tau \ \ ... \ \ \Gamma \vdash e_n : \tau \quad n \geq 0}{\Gamma \vdash \{e_1, ..., e_n\} : \tau\texttt{[]}} \qquad \frac{\Gamma \vdash e_1 : \tau\texttt{[]} \quad \Gamma \vdash e_2 : \texttt{int}}{\Gamma \vdash e_1[e_2] : \tau}$$

$$\frac{\Gamma \vdash e_1 : \texttt{int} \quad \Gamma \vdash e_2 : \texttt{int} \quad \oplus \in \{+, -, *, /, \%\}}{\Gamma \vdash e_1 \oplus e_2 : \texttt{int}} \qquad \frac{\Gamma \vdash e_1 : \texttt{int} \quad \Gamma \vdash e_2 : \texttt{int} \quad \ominus \in \{==, !=, <, <=, >, >=\}}{\Gamma \vdash e_1 \ominus e_2 : \texttt{bool}}$$

$$\frac{\Gamma \vdash e_1 : \texttt{bool} \quad \Gamma \vdash e_2 : \texttt{bool} \quad \oplus \in \{==, !=, \&, |\}}{\Gamma \vdash e_1 \ominus e_2 : \texttt{bool}} \qquad \frac{\Gamma(f) = \texttt{fn}\ \tau \to T' \quad T' \neq \mathit{unit} \quad \Gamma \vdash e : \tau}{\Gamma \vdash f(e) : T'}$$

$$\frac{\Gamma(f) = \texttt{fn}(\tau_1, ..., \tau_n) \to T' \quad T' \neq \texttt{unit} \quad \Gamma \vdash e_i : \tau_i{}^{(\forall i \in 1...n)} \quad n \geq 2}{\Gamma \vdash f(e_1, ..., e_n) : T'}$$

Sentencias.

Para la siguiente parte de nuestro sistema de tipos debemos tener en cuenta las sentencias del lenguaje que se van a complementar con las reglas para las expresiones.
Para un lenguaje como el nuestro, en un compilador de verdad el sistema de tipos para las sentencias es algo así como dinámico en el sentido de que el contexto se va construyendo en el mismo orden que sucede en el programa (pero aún en tiempo de compilación) pero para simplificar este paso en nuestro compilador solo tomaremos en cuenta las siguientes reglas que podemos deducir a partir de nuetra tabla de símbolos.
Agregamos la metavariable s para referirnos a una sentencia.

$$\frac{\Gamma(x) = \texttt{var}\ \tau \quad \Gamma \vdash e : \tau}{\Gamma \vdash x = e : \texttt{unit}}\ \textsc{Assign} \qquad \frac{\Gamma \vdash e : \texttt{bool} \quad \Gamma \vdash s : \texttt{unit}}{\Gamma \vdash \texttt{if}(e)\ \ s : \texttt{unit}}\ \textsc{If} \qquad \frac{\Gamma \vdash e : \texttt{bool} \quad \Gamma \vdash s : \texttt{unit}}{\Gamma \vdash \texttt{while}(e)\ \ s : \texttt{unit}}\ \textsc{While}$$

$$\frac{\Gamma \vdash e : \texttt{bool} \quad \Gamma \vdash s_1 : \texttt{unit} \quad \Gamma \vdash s_2 : \texttt{unit}}{\Gamma \vdash \texttt{if}(e)\ \ s_1\ \ \texttt{else}\ \ s_2\ \ : \texttt{unit}}\ \textsc{If-Else} \qquad \frac{\Gamma \vdash e_1 : \tau[] \quad \Gamma \vdash e_2 : \texttt{int} \quad \Gamma \vdash e3 : \tau}{\Gamma \vdash e_1[e_2] = e_3 : \mathit{unit}}\ \textsc{ArrAssign}$$

$$\frac{\Gamma(f) = \texttt{fn}\ \tau \to \texttt{unit} \quad \Gamma \vdash e : \tau}{\Gamma \vdash f(e) : \texttt{unit}}\ \textsc{PrCall}$$

$$\frac{\Gamma(f) = \texttt{fn}\ \ (\tau_1, ..., \tau_n) \to \texttt{unit} \quad \Gamma \vdash e_i : \tau_i{}^{(\forall i \in 1...n)} \quad n \geq 2}{\Gamma \vdash f(e_1, ..., e_n) : \texttt{unit}}\ \textsc{PrCallMulti}$$

Cuando decimos que una sentencia tiene tipo `unit` cuando no incumple con ninguna de las hipótesis.

Ejercicios.

- ***(10 puntos)*** Define con las reglas enunciadas, un proceso llamado `type-check` que se encargue de que dado un programa parseado y su respectiva tabla de símbolos, verifique que tipos de las expresiones y las sentencias sean correctos.