Prof. Dr. Peter Thiemann
Matthias Keil Summer Term 2015

Essentials of Programming Languages

`http://proglang.informatik.uni-freiburg.de/teaching/konzepte/2015/`

Exercise Sheet 4

4.1 Typed-Let

In the lecture, types and type environments were introduced.

$$\tau ::= \texttt{Int} \mid \texttt{Bool}$$
$$\Gamma ::= \emptyset \mid \Gamma, x : \tau$$

Write down the typing rules for the *Let* language from the previous exercise.

$$\begin{aligned}
\langle \mathit{val} \rangle \;&::= n \\
&\mid \texttt{true} \mid \texttt{false} \\[1em]
\langle \mathit{exp} \rangle \;&::= \langle \mathit{val} \rangle \\
&\mid \langle \mathit{exp} \rangle + \langle \mathit{exp} \rangle \mid \langle \mathit{exp} \rangle - \langle \mathit{exp} \rangle \mid \langle \mathit{exp} \rangle * \langle \mathit{exp} \rangle \mid \langle \mathit{exp} \rangle \,/\, \langle \mathit{exp} \rangle \\
&\mid \langle \mathit{exp} \rangle < \langle \mathit{exp} \rangle \mid \langle \mathit{exp} \rangle \wedge \langle \mathit{exp} \rangle \mid \langle \mathit{exp} \rangle \vee \langle \mathit{exp} \rangle \mid \neg \langle \mathit{exp} \rangle \\
&\mid \texttt{if } \langle \mathit{exp} \rangle \texttt{ then } \langle \mathit{exp} \rangle \texttt{ else } \langle \mathit{exp} \rangle \\
&\mid \langle \text{ident} \rangle \mid \texttt{let } \langle \text{ident} \rangle = \langle \mathit{exp} \rangle \texttt{ in } \langle \mathit{exp} \rangle
\end{aligned}$$

4.2 Error Let

In the *Let* language, type errors may interrupt the program execution. The following definition shows the specification of *Error-Let* as an extension of *Let*.

$$\langle \mathit{val} \rangle \;::=\; \cdots \mid \texttt{err}$$

The error value **err** is a special value arising from run-time type errors. An evaluation of an expression results in the error value if the expression is applied to values which are not in the domain of the expression. *For example:*

$$\begin{aligned}
1 + \mathit{true} &= \texttt{err} \\
1 \,/\, 0 &= \texttt{err}
\end{aligned}$$

- For each expression specify a suitable domain.
- Write down a set of small-step evaluation rules to specify error handling.
- Implement error handling in your interpreter.

Submission

Deadline The submission deadline is **22.05.2015, 12:00 (noon)**. Late submissions will not be accepted. Submit your solution to the subversion repository.

Submission Your solution will consist of one *folder* (**exercise4**) for each exercise sheet. Submit one *pdf* file (**⟨name⟩_exercise4_⟨nr⟩.pdf**) and one *rkt* file (**⟨name⟩_exercise4_⟨nr⟩.rkt**) per exercise.

Your solution may be either in German or in English. Clear and understandable style is required. You are strongly encouraged to test your solution. Your code must compile without errors (which did not necessarily mean that everything has to work). Provide your source code with comments to understand the intention.