P. Thiemann, G. Radanne Summersemester 2017/18

**Essentials of Programming Languages**

https://proglang.informatik.uni-freiburg.de/teaching/konzepte/2018ss/

## Language 1 – WHILE

2018-04-18

# While

The WHILE language is composed of mutable variables and while loops:

$$
\begin{array}{lll}
\multicolumn{3}{l}{\text{Expressions}} \\
e & ::= & x \mid (+\ e\ e) \mid \ldots \\
\multicolumn{3}{l}{\text{Statements}} \\
s & ::= & (\texttt{set}\ x\ e) \qquad \text{assignment} \\
 & \mid & \texttt{skip} \qquad \text{empty statement} \\
 & \mid & (s\ s\ \ldots) \qquad \text{sequence} \\
 & \mid & (\texttt{if}\ e\ s\ s) \qquad \text{conditional} \\
 & \mid & (\texttt{while}\ e\ s) \qquad \text{repetition}
\end{array}
$$

**Exercise 1** (Big step)
Implement a big step semantics with environment for the WHILE language. You can reuse the code used for the *arith* language for expressions.

**Exercise 2** (Exceptions)
We extend the WHILE language with exceptions:

$$r ::= \texttt{div0} \mid \ldots s \qquad\qquad ::= \cdots \mid (\texttt{throw}\ r) \mid (\texttt{try}\ s\ (r\ s))$$

Implement the two big step judgements for WHILE with exceptions. Use property testing to check that all programs either return a value or an exception.

**Bonus** Add new exceptions and implement a multi-catch handler:

$$s ::= \cdots \mid (\texttt{try}\ s\ (r_0\ s_0)\ (r_1\ s_1) \ldots)$$

if the exception raised by $s$ is $r_0$, then $s_0$ is executed, and so on. Do you know any other similar language construct? What are the differences?

**Exercise 3** (For loops)
We consider adding for loops to our language:

$$s ::= \cdots \mid (\texttt{for}\ x\ (v\ \ldots)\ s)$$

where $x$ ranges over the values $v_i$ in $s$.

Implement for loops using `while` and `set`. In your semantics, what is the scope of $x$ ? In particular, is $x$ available outside of the loop ?

In programming languages that you know, what is the scope of the for-loop variable ? You can try and report your findings.