

---

## Essentials of Programming Languages

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

---

### Exercise Sheet 3

#### 3.1 Variable Bindings

In a language without variables you cannot store intermediate results. Thus, we extend our language with variables and variable bindings:

$$\langle exp \rangle ::= \dots \mid \langle ident \rangle \mid \mathbf{let} \langle ident \rangle = \langle exp \rangle \mathbf{in} \langle exp \rangle$$

As you have seen in the lecture, we need a binding environment if we want to evaluate expressions containing variables. Implement variables and variable bindings in your interpreter. Choose a call-by-value behavior.

*Example:*

$$\mathbf{let} \ x = 1 + 2 \ \mathbf{in} \ (\mathbf{let} \ x = x + 2 \ \mathbf{in} \ x) = 5$$

#### 3.2 Free and Bound Variables

The notation of free and bound variables for expressions

$$e ::= x \mid \mathbf{let} \ x = e \ \mathbf{in} \ e$$

was introduced.

- Find an expression  $e$  such that  $FV(e) \cap BV(e) \neq \emptyset$  holds.
- For each of the following expressions, build a derivation tree and specify for each step the set of free and bound variables.

- $(x + y)$
- $\mathbf{let} \ x = (x + 1) \ \mathbf{in} \ (x + 1)$
- $\mathbf{let} \ x = 1 \ \mathbf{in} \ (x + 1)$
- $\mathbf{let} \ x = 1 \ \mathbf{in} \ (\mathbf{let} \ y = 1 \ \mathbf{in} \ (x + y))$
- $\mathbf{let} \ x = y \ \mathbf{in} \ (\mathbf{let} \ y = z \ \mathbf{in} \ (y + z))$
- $\mathbf{if} \ (x < y) \ \mathbf{then} \ (\mathbf{let} \ z = x \ \mathbf{in} \ z) \ \mathbf{else} \ (\mathbf{let} \ z = y \ \mathbf{in} \ z)$

### Submission

**Deadline** The submission deadline is **15.05.2015, 12:00 (noon)**. Late submissions will not be accepted. Submit your solution to the subversion repository.

**Submission** Your solution will consist of one *folder* (**exercise3**) for each exercise sheet. Submit one *pdf* file (**<name>\_exercise3\_<nr>.pdf**) and one *rkt* file (**<name>\_exercise3\_<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.