

## Design and Selection of Programming Languages

28 October 2005

### Exercise 7.1

Use the interaction framework from Exercise Sheet 4 and extend the parsers from the lecture to be able to produce `ghci`-like interaction:

```
let k = 3 * 4
> k = 12
let n = 3 * (k + 2)
> n = 42
n - k
> 30
n + m
> error: Variable "m" undefined!
```

#### Hints:

- Determine the input grammar for this system.
- Define a datatype to represent the ASTs of these inputs.
- Define a parsing function for these inputs.
- The current version of *SimpleLexer* on the course page can be used, but is of course not ideal. Starting from a variant of the lexer from Exercise 6.4 would be better.
- Determine an appropriate state datatype for this interaction.
- You will need an evaluation function for expressions with variables; this will of course need a variable assignment.
- Define the *Transition* function for this interaction.
- Define a *main* computation (using *runprocess* from Sheet 4) and compile your program.  
Test both the interpreted and the compiled versions.