Structure of Programming Languages I: Denotational Semantics

Assignments

Wolfgang Schreiner
Wolfgang.Schreiner@risc.uni-linz.ac.at
http://www.risc.uni-linz.ac.at/people/schreine
Research Institute for Symbolic Computation (RISC-Linz)
Johannes Kepler University, Linz, Austria

November 19, 1997

Please return the completed results by February 1. Put them in an envelope, address it to me, and put it into some university mail box. If there are problems, feel free to contact me per email. Do not forget to denote your name, Matrikelnummer and Studienkennzahl (SKZ).

1. Remove the command [[**read** I]] from the language presented in "Imperative Languages II" and add an *expression*

   ```
   read
   ```

   that reads the next element from the input stream (and advances the stream). Rewrite the semantics of expressions to match the new requirements and prove that the semantics of

   ```
   I := read
   ```

   is the same as that of the original command. Furthermore, add the assignment statement $[[I{:=}E]]$ to the domain of expressions such that its value is the value of the identifier after the assigment is performend and rewrite the semantics of expressions accordingly.

2. Extend the language presented in "A Block-Structured Language" with pointers: extend *Denotable-value* by a variant *Ptrlocn* = *Location* (locations that hold pointers); augment the syntax of the language and give the semantics of pointer declaration, dereferencing, assignment, and dynamic storage allocation. How does the integration of pointers change the stack-based storage model?