Problem Set 9: The Metacircular Evaluator

Out: Tuesday, 21 November 2006
Due: Thursday, 5 December 2006

Overview: In this problem set, you’ll make modifications to the metacircular evaluator, changing UML Scheme.

To run the metacircular evaluator, download mceval.ss from the course web site. Use the MzScheme language in DrScheme. After executing the code, evaluate (driver-loop). This will run the interactions for your metacircular UML Scheme world. Remember that everything you type in (except for variables and numbers) should be prefixed by “uml:” – if you forget, the system will have an error. If you type in an error, the driver-loop exits and you lose the state you had in the metacircular world.

You only need to turn in the portions of the code that you change.

Warm-up:
Run the metacircular evaluator and evaluate some UML Scheme expressions. Nothing to turn in for this part.

Problem 1:
Exercise 4.4 (“or” only) on p. 374. Remember to tag your new “or” with “uml:”.

Problem 2:
Create the uml:if-not syntax in the metacircular evaluator:
(uml:if-not <pred> uml:then <consequent> uml:else <alternative>)
If the predicate is false, the consequent should be evaluated. If the predicate is true, the alternative should be evaluated.

Problem 3:
Add uml:list to the metacircular evaluator. It should be a derived expression, meaning that you should convert the uml:list expression into a series of uml:cons expressions, with a nil at the end (nil, like true and false, are added to the global environment when setup-environment is evaluated and its value defined as the-global-environment). Hint: remember that uml:cond was also a derived expression; it was converted to a sequence of uml:if statements.

Problem 4:
Exercise 4.10 on p. 376. Choose your own way to change UML Scheme’s syntax. You can add a new statement, change the syntax of existing statements, make Scheme be postfix instead of prefix, etc. For this problem, explain what you did as well as turning in the code.