Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.827   Multithreaded Parallelism: Languages and Compilers

Please Remember: The Midterm Exam will happen in class on Lecture 12 date. You
are allowed to use lecture notes (this is an open-notes test), but the problem sets and the textbook
are **not permitted**.

Topics

We have listed these topics below and enumerated the important subtopics that you should know.
The stars next to the Hindley-Milner type rules and inference algorithm indicate that we are not
requiring you to be intimately familiar with the rules or to mechanically apply them. You are,
however, required to be able to identify the type of any expression you are given or indicate that
the expression is type incorrect. We recommend you to look at problems in the pH Book in order
to get the feel for the quiz. Those include the problems in Chapters 2, 3, 4, and Appendix A.

1. **Lambda-calculus.** (pH Book: Chapter 2, Appendix A)

    - free and bound variables in an expression
    - beta-substitution
    - alpha and eta rules
    - reducing a lambda-calculus expression
    - renaming variable to avoid capture
    - recursion, fixed points, the Y combinator
    - confluence (Church-Rosser) property

2. **Interpreters.** (pH Book: Chapter 4)

    - normal form, head normal form, weak normal form
    - reduction strategies
        - applicative order
        - normal order
    - normalizing property of an interpreter or reduction strategy
    - call-by-name interpreter, call-by-value interpreter

3. **Lambda-calculus with constants and let-blocks.** (pH Book: Chapter 4)

    - projection functions and constructors

- new reduction rules
  - alpha and beta rules
  - instantation of let-bound variables
  - liftening and flattening rules for let-blocks

4. **Types.** (pH Book: Chapter 3, Appendix A)

- polymorphism
  - in let-blocks versus lambda-abstractions
- type inference/reconstruction (*)
  - Hindley-Milner typing rules and inference algorithm (*)
  - generalization (*)
  - unification (*)
- types versus type schemes

5. **Overloading.** (pH Book: Chapter 3)

- contrast with polymorphism
- type classes
- ambiguity in overloading resolution