

CSE 341
FINAL Review Sheet
Open Book, Open Notes

Lisp

1. Be able to construct lists and dotted pairs and extract elements and substructures using primitive functions such as *cons*, *car*, *cdr*, *first*, *rest*, *append*, *eq*, *eql*, *equal*, *cond*, *list*, *quote*, *atom*, *null*. Be able to draw internal representations of such structures.
2. Be able to write recursive functions that can handle arbitrary lists or arbitrary S-expressions.
3. Be able to write or understand functions that use *let* or *let** and the more modern control structures such as *if*, *when*, *unless*, *loop*, *dotimes*, and *dolist*.
4. Be able to work with functions that have other functions as arguments, particularly *mapcar*, *eval*, *apply*, and *funcall*, with and without *lambda expressions*.
5. Be able to program in a pure functional form.
6. Be able to intelligently discuss the Lisp language, the Lisp interpreter, and functional programming.

Prolog

1. Be able to write Prolog facts and rules from English statements or questions using both constants and variables.
2. Be able to show how Prolog searches for the answer to a given query on a given database.
3. Be able to write recursive rules.
4. Be able to write rules that compute results.
5. Be able to write rules that handle lists.
6. Be able to use built-in predicates, built-in arithmetic, and *is*.
7. Be able to use cuts to prune the search or to explain how they work.
8. 9. Be able to construct Prolog structures (facts, rules, queries) dynamically and to use *=..*, *assert*, *call*, *read*, and *write* as needed.

General Concepts

1. Be able to discuss or use the following concepts, either generally or with respect to Java, Lisp, and Prolog.
 - (a) syntax (alphabets, tokens, rules, BNF, derivations)
 - (b) semantics (just know what it is, not formal semantic specs)
 - (c) translation (interpreters vs compilers)
 - (d) bindings and binding times
 - (e) variables (name, scope, type, memory, value)
 - (f) references
 - (g) routines and parameter passage methods
 - (h) runtime environments

2. Be able to compare Java, Lisp, and Prolog with respect to the following:
 - (a) characterizing features
 - (b) syntax
 - (c) structural organization of a program
 - (d) data types and structures provided
 - (e) control structures
 - (f) memory management
 - (g) binding constructs
 - (h) execution environments