COP 4020

PROGRAMMING LANGUAGE CONCEPTS

Instructor: Manuel E. Bermudez

SAMPLE MIDTERM EXAM

Closed book, notes, 50 min.

Problem 1 ______________(40p.)
Problem 2 ______________(60p.)

SCORE ______________(100p.)

NAME ____________________________

NOTE: Turn in your work on this exam only.
PROBLEM 1. (40 points)

Convert the following regular expression to a NFA, then transform the NFA to a right-linear regular grammar, and finally transform the grammar to the original regular expression. Show your work.

\[(a^*b + c^*)^* + (de+f)^*\]
WORKSPACE
PROBLEM 2. (60 points)

Below is a string-to-tree transduction grammar for regular expressions.

Expression  ->  Term (‘|’ Term)+  =>  "|"
            ->  Term;
Term       ->  List List+       =>  "cat"
            ->  List;
List       ->  Factor ‘list’ List =>  "list"
            ->  Factor;
Factor     ->  Primary ‘*’       =>  "*"
            ->  Primary ‘+’       =>  "+
            ->  Primary ‘?’       =>  "?”
            ->  Primary;
Primary    ->  ‘<id>’            =>  "<id>"
            ->  ‘( Expression ’)’;

Note that the alternation operator is |, and + is the unary postfix "one or more" operator.

a)  (20 pts.) Draw the Derivation Tree and the Abstract Syntax Tree for the expression (a|b*)* list c a?. Hint: consider abbreviating Expression, Term, List, Factor, and Primary, as E, T, L, F, and P, respectively.
b) (40 pts.) Write the skeleton of the corresponding recursive descent parser, including `BuildTree('node', N)` statements that will build the Abstract Syntax Tree in bottom-up fashion.
WORKSPACE