

Midterm

Name (please print): _____

Notes

- (1) This book is closed book.
- (2) Some students are taking the exam at different times. **Sign this statement:** "I have not communicated with other people about this exam, and I will not communicate with others who are still taking the exam." _____
- (3) There are 4 questions; they are worth equal credit.
- (4) Put your answers on these pages. If you want to hand in scratch work, you may; make sure your name is on each page.
- (5) If you need to make an assumption to clarify a problem, *write your assumption down*. Only reasonable assumptions get full credit.

1. Stages of compilation

a) Draw a rectangle, ellipse, and arrow picture showing the steps involved in compiling a program. Rectangles show data, ellipses show execution of code, and arrows are labelled with the type of the data passing between the rectangles and ellipses. Ignore compiler-construction tools.

b) What is the name of the scanner-generator program you used in Assignment 2?

c) Does your scanner use the symbol-table code you wrote in Assignment 1? _____

d) What is the expected complexity of symbol-table lookup? (Use big-O notation.)

2. Regular sets

a) Using the method shown in class, including all the λ transitions, convert this regular expression into a non-deterministic finite-state automaton (NFA): $a (b c \mid b d^*) e$

b) Convert the NFA you built in (a) into a deterministic finite-state automaton (DFA) using the subset construction.

c) Convert the DFA you built in (b) into an optimal DFA using the algorithm discussed in class.

d) List the sequence of 5 states your optimal DFA enters while accepting the string $abdde$.

3. Context-free grammars

Consider this grammar for "1-3 expressions". Here, the symbols (and) stand for themselves; they are not metacharacters.

$$\begin{aligned} S &\rightarrow i \\ S &\rightarrow (S) \\ S &\rightarrow (S S S) \end{aligned}$$

a) Is this grammar from right recursive, left recursive, both, not recursive, or otherwise recursive? (Circle one).

b) Show the parse tree, using this grammar, for this valid 1-3 expression:

((i) i ((i i i) i i))

c) This grammar is not LL(1). Why not?

4. Top-down parsing

Consider the grammar in the following chart

Production	First	Follow	Predict
1 $P \rightarrow DS Ss \$$			
2 $Ds \rightarrow D Ds$			
3 $Ds \rightarrow \lambda$			
4 $D \rightarrow f id$			
5 $D \rightarrow i id$			
6 $Ss \rightarrow S Ss$			
7 $Ss \rightarrow \lambda$			
8 $S \rightarrow id = E$			
9 $S \rightarrow p id$			
10 $E \rightarrow T + E$			
11 $E \rightarrow T$			
12 $T \rightarrow T * F$			
13 $T \rightarrow F$			
14 $F \rightarrow (E)$			
15 $F \rightarrow n$			

- Compute the First set **for each RHS** and the Follow set **for each nonterminal**. Place them in the list above.
- Compute the Predict set **for each rule** (for a top-down parser). Place them in the list above.
- Write code (don't worry about precise syntax) that implements the procedure in a recursive-descent parser for the nonterminal symbol Ds .

d) Give evidence **from your answer to (b)** that this grammar is not LL(1).