CS 352 – Compilers: Principles and Practice
Mid-term Examination, 10/18/05

Instructions: Read carefully through the whole exam first and plan your time. Note the relative weight of each question and part (as a percentage of the score for the whole exam). The total points is 100 (your grade will be the percentage of your answers that are correct).

This exam is **closed book, closed notes**. You may *not* refer to any book or other materials.

You have **75 minutes** to complete both (2) questions. Write your answers on this paper (use both sides if necessary).

Name:

Student Number:

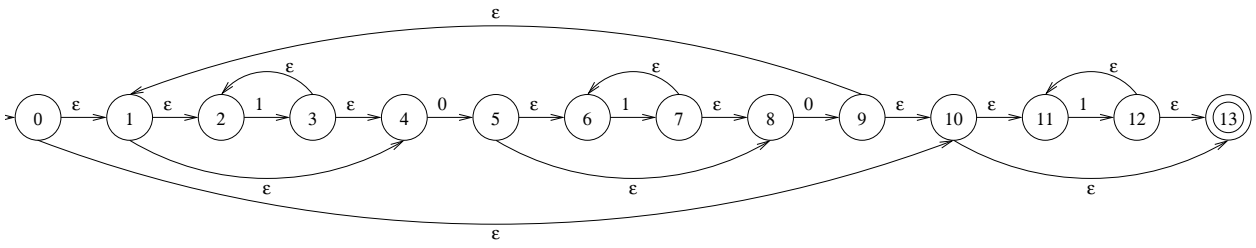
Signature

1. [Regular expressions, scanning; 40%]

   (a) (15%) As described in class, construct an NFA that recognizes the same language as defined by the following regular expression:

$$(1^*01^*0)^*1^*$$

   Answer:

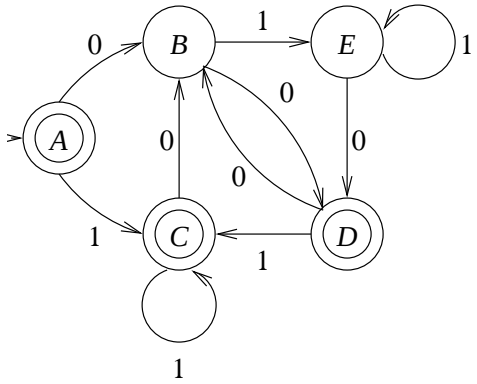


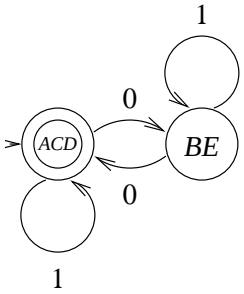
   (b) (25%) Using the subset construction, convert the NFA into a DFA. Optimize the resulting DFA by merging equivalent states (if any).

   Answer:

	0	1
$A = \{0,1,2,4,10,11,13\}$	B	C
$B = \{5,6,8\}$	D	E
$C = \{2,3,4,11,12,13\}$	B	C
$D = \{1,2,4,9,10,11,13\}$	B	C
$E = \{6,7,8\}$	D	E

	0	1
$ACD = \{0,1,2,3,4,9,10,11,12,13\}$	BE	ACD
$BE = \{5,6,7,8\}$	CD	BE





2. [Context Free Grammars, parsing; 60%] Consider the following simple grammar and the language it describes:

$$S \rightarrow CC$$
$$C \rightarrow Cc$$
$$C \rightarrow d$$

(a) (5%) In words, what *language* does this grammar describe?

Answer:

A *d* followed by a sequence of 0 or more *c*s, repeated twice.

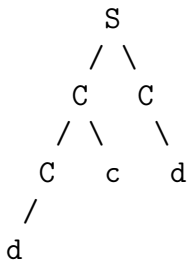
(b) (5%) Write a regular expression for this *language*.

Answer:

dc^*dc^*

(c) (5%) Draw the parse tree for the sentence *dcd* using this grammar.

Answer:



(d) (5%) Is this *language* LL(1)? Explain. [There is a simple argument.]

Answer:

Yes, the language is regular, and all regular languages are LL(1).

(e) (5%) Is this *grammar* LL(1)? Explain. [There is a simple argument.]

Answer:

No, the grammar is not LL(1) by inspection since it is left-recursive.

(f) (5%) Transform the grammar, as necessary, so that it can be parsed predictively by a top-down parser using 1 token of lookahead.

Answer:

$$S \rightarrow CC$$
$$C \rightarrow dC'$$
$$C' \rightarrow cC'$$
$$C' \rightarrow \varepsilon$$

(g) (10%) Derive an LL(1) parse table to parse this language.

Answer:

	FIRST	FOLLOW	c	d	\$
S	d	\$		$S \rightarrow CC$	
C	d	d\$		$C \rightarrow dC'$	
C'	$c\varepsilon$	d\$	$C' \rightarrow cC'$	$C' \rightarrow \varepsilon$	$C' \rightarrow \varepsilon$

(h) (10%) Show the steps of an LL(1) parser as it parses the input *dcd* using your parse table.

Answer:

Stack	Input
$S\$$	$dcd\$$
$CC\$$	$dcd\$$
$dC'C\$$	$dcd\$$
$C'C\$$	$cd\$$
$cC'C\$$	$cd\$$
$C'C\$$	$d\$$
$C\$$	$d\$$
$dC'\$$	$d\$$
$C'\$$	$\$$
$\$$	$\$$

(i) (10%) Consider the original grammar, now with numbered productions:

1	$S \rightarrow CC$
2	$C \rightarrow Cc$
3	$C \rightarrow d$

Here is its LR(1) parse table:

	c	d	\$	C
0		$s4$		1
1	$s3$	$s4$		2
2	$s3$		acc	
3	$r2$	$r2$	$r2$	
4	$r3$	$r3$	$r3$	

Show the steps of an LR parser as it uses this table to parse the input *dcd*.

Answer:

$0_{\$}$	$dcd\$$
$4_d\ 0_{\$}$	$cd\$$
$1_C\ 0_{\$}$	$cd\$$
$3_c\ 1_C\ 0_{\$}$	$d\$$
$1_C\ 0_{\$}$	$d\$$
$4_d\ 1_C\ 0_{\$}$	$\$$
$2_C\ 1_C\ 0_{\$}$	$\$$