# CS143 Midterm Exam SOLUTIONS (April 30, 2015)

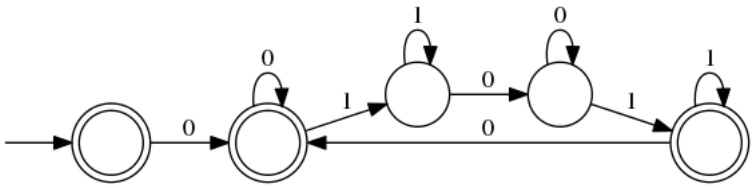
1. (10 points)

   For the alphabet $\{0, 1\}$, give a regular expression and draw a DFA for the set of all strings in which each 1 is preceded by one or more 0's (but not necessarily *immediately preceded*), and the number of "01" substrings appearing in the string is even (note: 01110111000 is in this language).

   **Regular expression**

   $(0^+1^+0^+1^+)^*0^*$

   **DFA**




2. (5 points) Draw a line through each useless production in the following grammar (hint: find the useful productions and symbols first.) **Note:** "Useless" has a specific mathematical definition that was presented in the lecture and notes.

~~$S \rightarrow AB$~~
$S \rightarrow S$
$S \rightarrow BB$
~~$A \rightarrow BC$~~
~~$A \rightarrow aAe$~~
~~$B \rightarrow AB$~~
$B \rightarrow aBb$
$B \rightarrow \epsilon$
~~$C \rightarrow AB$~~

3. (15 points)

Compute the FNE, Follow, and First sets for the nonterminals of the context-free grammar below, and write them in the spaces provided. Also, answer the questions about the grammar.

$$
\begin{array}{lcl}
S & \rightarrow & ABA \\
A & \rightarrow & aA \\
A & \rightarrow & \epsilon \\
B & \rightarrow & b \\
B & \rightarrow & \epsilon
\end{array}
$$

FNE

| | |
|---|---|
| $S$ | $a, b$ |
| $A$ | $a$ |
| $B$ | $b$ |

Follow

| | |
|---|---|
| $S$ | \$ |
| $A$ | $a, b, \$$ |
| $B$ | $a, \$$ |

First

| | |
|---|---|
| $S$ | $\epsilon, a, b$ |
| $A$ | $\epsilon, a$ |
| $B$ | $\epsilon, b$ |

Is the language of this grammar regular (circle one)? (Yes) No

Is the language of this grammar finite (circle one)? Yes (No)

Is the grammar LL(1) [answer without building the LL(1) parse table]? Yes (No)
(explain briefly below)

**Answer:** Consider the parse tree for the single string "a". We don't know if the first or second $A$ should expand to the 'a', so the grammar is ambiguous. Ambiguous grammars cannot be LL(1).

*Note:* Many students claimed that because the grammar is regular that it was LL(1). For any regular language there exists a grammar that is LL(1), but that doesn't mean that *every* grammar for that language is LL(1).

4. (10 points) Left factor and eliminate immediate left recursion in the following CFG:

$$
\begin{array}{lcl}
S & \rightarrow & Sa \\
S & \rightarrow & bS \\
S & \rightarrow & Sc \\
S & \rightarrow & bbA \\
A & \rightarrow & f
\end{array}
$$

**Left Factored:**

$$
\begin{array}{lcl}
S & \rightarrow & SX \\
X & \rightarrow & a \\
X & \rightarrow & c \\
S & \rightarrow & bY \\
Y & \rightarrow & S \\
Y & \rightarrow & bA \\
A & \rightarrow & f
\end{array}
$$

**No Left Recursion:**

$$
\begin{array}{lcl}
S & \rightarrow & BbbfD \\
B & \rightarrow & bB \\
B & \rightarrow & \epsilon \\
D & \rightarrow & cD \\
D & \rightarrow & aD \\
D & \rightarrow & \epsilon
\end{array}
$$

5. (10 points) Below is an SLR(1) parse table (ACTION and GOTO tables) for a context-free grammar. Next to it is a table giving the length of the right-hand side for each production in the grammar.

| | *ACTION* | | | | *GOTO* | | |
|---|---|---|---|---|---|---|---|
| | $a$ | $b$ | $d$ | \$ | $S$ | $A$ | $B$ |
| 0 | $s7$ | | $s8$ | | 1 | 2 | |
| 1 | | | | $r0$ | | | |
| 2 | | $s6$ | | | | | 3 |
| 3 | $s4$ | | | $r1$ | | | |
| 4 | | | $s5$ | | | | |
| 5 | $r4$ | | | $r4$ | | | |
| 6 | $r5$ | | | $r5$ | | | |
| 7 | | $r3$ | $r3$ | | | | |
| 8 | $s7$ | | $s8$ | | | 9 | |
| 9 | | | $s10$ | | | | |
| 10 | | $r2$ | $r2$ | | | | |

| prod no | LHS | RHS length |
|---|---|---|
| 0 | S' | 1 |
| 1 | S | 2 |
| 2 | A | 3 |
| 3 | A | 1 |
| 4 | B | 3 |
| 5 | B | 1 |

(a) Show the sequence of stacks and inputs when parsing "abad". Stacks should just contain state numbers (and \$). (The lectures showed stacks with states and symbols underneath them. Don't list the symbols.)

| Stack | Input |
|---|---|
| \$0 | abad\$ |
| \$07 | bad\$ |
| \$02 | bad\$ |
| \$026 | ad\$ |
| \$023 | ad\$ |
| \$0234 | d\$ |
| \$02345 | \$ |

| Stack | Input |
|---|---|
| \$02345 | \$ |
| \$023 | \$ |
| \$01 | \$ |
| accept | |
| | |
| | |
| | |

*Note:* Some people wrote an error at the end of the parse because after reducing using production 0, there's no GOTO entry for $S'$. Techincally this is correct, but the $r0$ action should just be accept, so that's what is written here.

(b) What is the context-free grammar from which the table was generated?

$$
\begin{array}{lcl}
S & \rightarrow & AB \\
A & \rightarrow & dAd \\
A & \rightarrow & a \\
B & \rightarrow & Bad \\
B & \rightarrow & b
\end{array}
$$