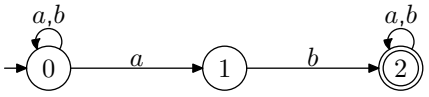
Midterm Exam

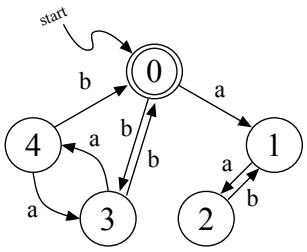
(Total points: 100 points)

1. Write a regular expression that denotes the following languages:
   (1) [**4**] All strings of `a`'s and `b`'s with the subsequence `abb`.
   (2) [**6**] All strings of `0`'s and `1`'s with an odd number of `0`'s and odd number of `1`'s.

2. [**6**] Convert the following NFA to a DFA using the subset construction.



3. [**8**] Minimize the following DFA by applying the DFA minization algorithm discussed in class. You need to write the derivation process and draw the resulting diagram.



4. Consider the following grammar with terminals a, b, c, and $+$.

$$S \to S + a \mid S + b \mid c$$

   (1) [**6**] Rewrite the grammar so that it can be parsed by a predicative top-down parser.
   (2) [**8**] Construct the LL(1) parsing table of the rewritten grammar.

5. Given the following ambiguous grammar G, which generates strings representing arithmetic expressions with two operators + and *, and numbers as operands,

$$E \to E + E \mid E * E \mid (E) \mid \textbf{num}$$
$$\quad\ \ (1) \qquad (2) \qquad (3) \qquad (4)$$

   (1) [**10**] Construct the SLR parsing table of grammar G. Use the number below a production body to indicate which production rule to reduce.
   (2) [**4**] Resolve the conflict(s), if any, in the table created in the first part of this question. Justify your answer.
   (3) [**8**] Rewrite grammar G to eliminate the ambiguity.
   (4) [**4**] Why would one write an ambiguous grammar, rather than an unambiguous grammar, for constructing a parser? Give at least one reason.

6. [**10**] Given the following augmented grammar with terminals =, *, and **id**, derive the following two states in its canonical sets of LR(1) items: the starting state and the state reached on a transition for shifting * from the starting state.

$S' \rightarrow S$
$S \rightarrow L{=}R \mid R$
$L \rightarrow {*}R \mid \mathbf{id}$
$R \rightarrow L$

7. Consider the following grammar with terminals y and $+$.

$S \rightarrow A$
$A \rightarrow A + A \mid B + +$
$B \rightarrow y$

(1) [**6**] Draw the parse tree for the input "$y + + + y + +$".

(2) [**14**] Complete the table below tracing an LR(1) parse of the input above. The *Stack* column shows the stack (with the top at right), the *Input* column shows the not-yet-processed input, and the *Action* column shows whether the parser performs a shift action, a reduce action, or accepts the input. In the case of a reduce action, indicate which production is used. (NOTE: you do not need to construct the parsing table; just use your knowledge of how the parser works and the parse tree given in the previous part.)

Stack (with top at right)	Input	Action

8. [**6**] Which of LL(1), SLR(1), and LR(1) can parse strings in the following grammar (with terminals a, b, and c), and why?

$E \rightarrow A \mid B$
$A \rightarrow a \mid c$
$B \rightarrow b \mid c$