Midterm I
CS164, Spring 2004

February 26, 2003

- Please read all instructions (including these) carefully.
- **Write your name, login, SID, and circle the section time.**
- There are 7 pages in this exam and 4 questions, each with multiple parts. Some questions span multiple pages. All questions have some easy parts and some hard parts. If you get stuck on a question move on and come back to it later.
- You have 1 hour and 20 minutes to work on the exam.
- The exam is closed book, but you may refer to your two pages of handwritten notes.
- Please write your answers in the space provided on the exam, and clearly mark your solutions. You may use the backs of the exam pages as scratch paper. Please do not use any additional scratch paper.
- Solutions will be graded on correctness and clarity. Each problem has a relatively simple and straightforward solution. We might deduct points if your solution is far more complicated than necessary. Partial solutions will be graded for partial credit.

LOGIN: _____

NAME: _____

SID: _____

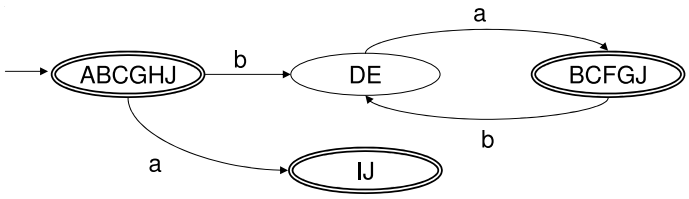
Circle the time of your section: Tue 3:00 Tue 4:00 Wed 10:00 Wed 11:00 Wed 1:00 Wed 2:00

Problem	Max points	Points
1	20	
2	15	
3	50	
4	15	
TOTAL	100	

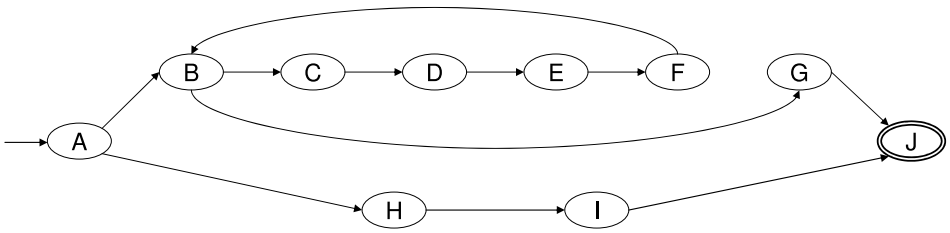
1 Finite Automata and Regular Expressions (20 points)

(a) Draw a **4-state** DFA over the alphabet $\Sigma = \{a, b\}$ that accepts the language consisting of strings with an even number of a (this includes zero a), or an odd number of b.

(b) Consider the following DFA over the alphabet $\Sigma = \{a, b\}$.



i. Label the transitions of the following NFA so that it accepts the same language as the DFA. Your NFA should transform to the given DFA by applying the NFA-to-DFA conversion algorithm given in lecture.



ii. Write the regular expression for this language. Among the several possible answers, write the one that would transform to this NFA by applying the regular expression-to-NFA conversion algorithm given in lecture.

2 LL Parsing (15 points)

Consider the following grammar:

$$
\begin{array}{lcl}
S & \rightarrow & [S\ X] \mid a \\
X & \rightarrow & \epsilon \mid +\ S\ Y \mid Y\ b \\
Y & \rightarrow & \epsilon \mid -\ S\ X\ c
\end{array}
$$

The nonterminals are S, X, and Y, and the terminals are a, b, c, $+$, $-$, [, and].

(a) Compute the First and Follow sets for this grammar. Enter your results in the table below. The first one has been done for you.

	First	Follow
S	$a, [$	
X		
Y		

(b) Fill in the LL(1) parsing table for this grammar. The first cell has been done for you.

	a	b	c	$+$	$-$	[	]	\$
S	a							
X								
Y								

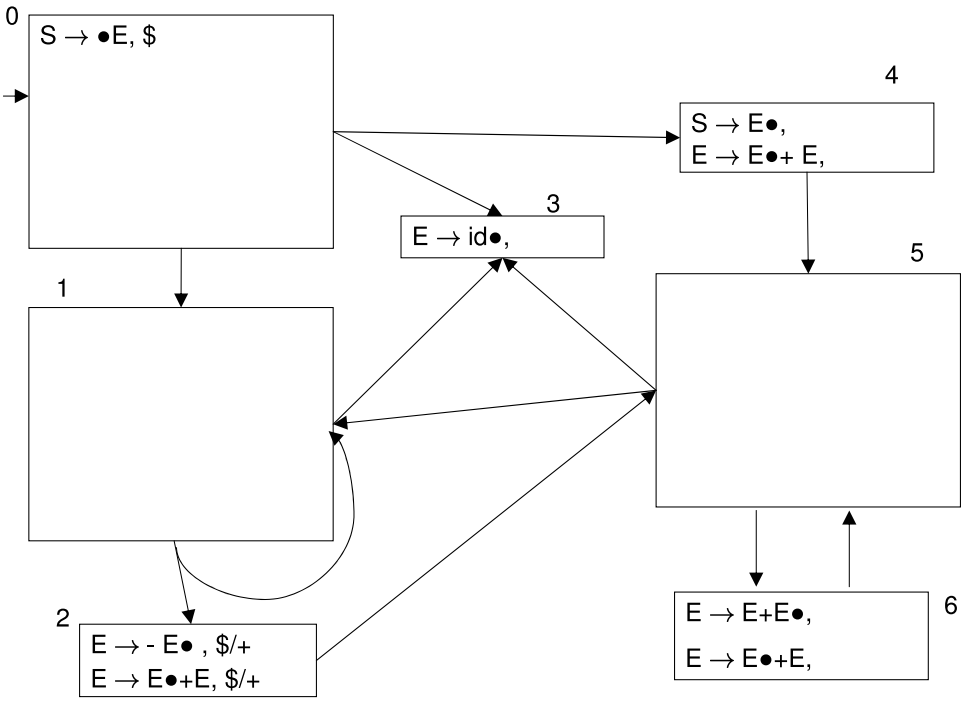
(c) Is this grammar LL(1)? Why or why not?

3 LR Parsing and Ambiguity (50 points)

Consider the following grammar over the terminals $+$, $-$ (the negation operator) and id.

$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow E + E \mid -E \mid id \end{aligned}$$

Below is a partial DFA for this grammar.



(a) Complete the above DFA. You need to do the following:

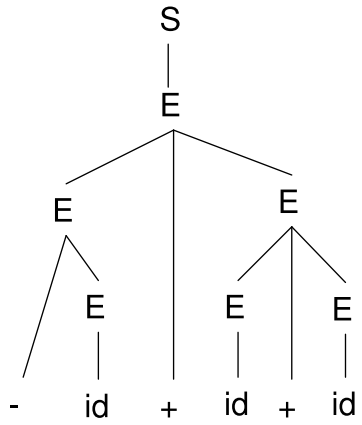
- Complete state 0 by performing closure on the item listed.
- Fill in all elements of states 1 and 5, and the lookahead items in states 3, 4 and 6.
- Fill in the missing transition labels on all edges.
- Write the necessary "reduce by ..." labels on states.

(b) For each state with a conflict, list the state, the lookahead token, and the type of conflict (i.e. shift-reduce conflict, or reduce-reduce conflict).

(c) Draw all the parse trees for the string $id + - id + id$.

(d) Is this grammar ambiguous? Why or why not?

Suppose we want the string $- \ id + id + id$ to have only the following parse tree (call this property P).



(e) Describe in English the precedence and associativity rules necessary to ensure property P.

(f) Explain, for each conflict in the LR(1) parsing DFA for this grammar, how it should be resolved to ensure property P (as a shift, or as a reduce).

(g) Rewrite the grammar to an **equivalent unambiguous** grammar to ensure property P. Two grammars are equivalent when they accept the same languages.

4 Grammars (15 points)

Consider two LL(1) grammars, each with a single non-terminal, and without ϵ productions. The non-terminal for the first grammar is A and for the second one is B.

Consider a third grammar obtained by merging the productions of the two grammars and adding the production $S \rightarrow A \mid B$. Answer the following questions, and explain your answers.

(a) Is it possible to derive the empty string from S in the third grammar?

(b) Write the necessary and sufficient condition(s) that the two original grammars must satisfy, in order to ensure that the third grammar is unambiguous. Do not give examples of grammars for A and B, but give instead a general condition that applies to all grammars of the form mentioned above.

(c) Write the necessary and sufficient condition(s) that the two original grammars must satisfy, in order to ensure that the third grammar is LL(1). Do not give examples of grammars for A and B, but give instead a general condition that applies to all grammars of the form mentioned above.