

## 03-60-214 Computer Languages, grammars, and translators (2004)

## Final exam (3 hours)

Name	Student Number	Marks

There are 15 questions, 8 pages. You can use pen or pencil. Total score is 35.

1. (1 point) According to Chomsky hierarchy, which grammar is the most powerful in the following choices?
  - a. Regular grammar
  - b. Context sensitive grammar
  - c. Context free grammar
  - d. LR(1) grammar

Your answer: \_\_\_\_\_

2. (1 point) Which one of these is a topdown parsing method?
  - a. LR(1)
  - b. LL(1)
  - c. SLR
  - d. LALR

Your answer: \_\_\_\_\_

3. (1 point) LALR stands for: \_\_\_\_\_

4. (1 point) Given the following grammar, where **A** is a non-terminal, **a** and **b** are terminals:  
 $A \rightarrow aA|b$

Write the regular expression that can recognize the same language.

5. (1 point) Given the following DTD. Write one XML document that is valid with respect to this DTD.

```
<!ELEMENT stocks (stock*)>
<!ELEMENT stock (name, symbol?, price)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT symbol (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

6. (1 point) Which of the following methods can parse strings in the following grammar, where **E**, **A**, and **B** are non-terminals, **a**, **b**, and **c** are terminals. Explain your answer.

$$E \rightarrow A \mid B$$

$$A \rightarrow a \mid c$$

$$B \rightarrow b \mid c$$

- a. LL(1)
  - b. Recursive Descent
  - c. SLR
  - d. None of them
7. (2 points) Write a CFG or RE for the following languages over alphabet {a, b}:
- b. Strings that have twice number of *a*'s as *b*'s. such as "*aab*", "*aababa*", and "*aba*".
  - c. Strings with one *a* and even number of *b*'s, such as "*abb*", "*abbbb*", "*babbb*".
8. (2 points) Given the regular expression  $(ab)^*$ , write the corresponding regular grammar that can recognize the same language.
9. (2 points) Consider the grammar
- $$S \rightarrow aSbS \mid bSaS \mid \epsilon$$
- Show the grammar is ambiguous
10. (2 points) Write JavaCup specification for formal parameter declaration in Tiny language used in our assignment 3 and assignment 4. Your answer should be complete and can be run without conflicts. Write the production rule(s) as well as the declarations for terminals and non terminals. You don't need to write the JLex specification.

11. (3 points) Given the following grammar:

$$S \rightarrow AaAb \mid BbBa$$
$$A \rightarrow \varepsilon$$
$$B \rightarrow \varepsilon$$

- a) is it LL(1)? explain your answer.
- b) is it SLR(1)? explain your answer.

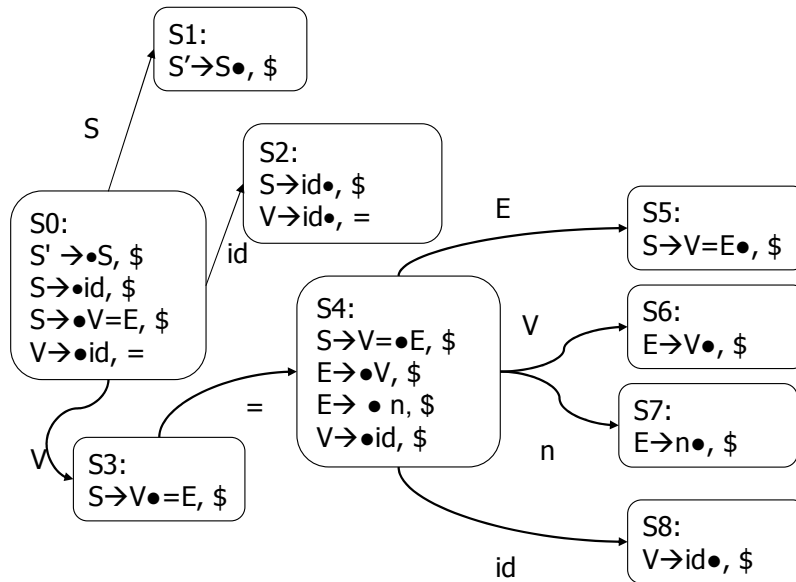
12. (5 points) For the regular expression  $(aa)^*|(aaa)^*$ ,
- a) (2 points) Draw its NFA. You can produce the NFA either by intuition or Thompson construction.
  - b) (2 points) Derive the DFA. You need to write down your derivation steps and the resulting diagram.
  - c) (1 points) Draw the minimized DFA. You need to write down your justifications.

13. (4 points) Consider the following grammar, where P, C, and D are non-terminals and other symbols are terminals.

$$P \rightarrow \{D ; C\}$$
$$D \rightarrow d; D \mid d$$
$$C \rightarrow c; C \mid c$$

- a) Is the grammar LL(1)? Explain your answer.
- b) Is the grammar SLR(1)? Explain your answer.
- c) Is the grammar LR(1)? Explain your answer.
- d) Is the grammar LALR? Explain your answer.

14. (5 points) Given the following transition diagram, where S, V, and E are non-terminals, and others are terminals.



- (1 point) Write the grammar that the diagram is parsing.
- (1 point) Is the grammar SLR or LR(1)? Explain your answer.
- (1.5 point) Construct the LR(1) parse table.



15. (4 points) Given the following grammar, where **E** is a non-terminal, +, \*, and **a** are terminals.

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow a$

- a) (1 point) Draw a partial transition diagram of exactly those states of an LR(1) parser that are pushed on the parse stack during a parse of the input **a+a\*a**. Do not add or change any productions in the grammar. If you accidentally draw other states, please cross out those states.
- b) (1 point) Is the grammar LR(1)? Explain your answer.
- c) (1 point) In JavaCup, can you directly use this grammar without transforming it? Explain why.
- d) (1 point) If you want to parse the string **a+a\*a** correctly, i.e., as **a+(a\*a)**, please write the necessary additional javaCup code to produce the correct parse tree.