

COP5621 Exam 1 - Spring 2005

Name: _____ (Please print)

Put the answers on these sheets. Use additional sheets when necessary. Show how you derived your answer when applicable (this is required for full credit and helpful for partial credit). You can collect 100 points in total for this exam. A bonus question for an additional 15 points is included. If the total number of points for this exam exceed 100, the excess points are carried over to the next exams.

1. Match the terms below with the given sentences so as to best complete each sentence. Use no term more than once. Some terms will go unused. (10 points)

- | | |
|-------------------------|------------------------|
| (1) preprocessor | (9) lexeme |
| (2) attribute | (10) lexical analysis |
| (3) back end | (11) syntax analysis |
| (4) front end | (12) token |
| (5) nonterminal | (13) terminal |
| (6) NFA | (14) DFA |
| (7) regular expressions | (15) grammar |
| (8) regular definitions | (16) translation rules |

- (a) The _____ of a compiler produces intermediate code.
- (b) Syntax errors are detected in the _____ phase of a compiler.
- (c) In a typical compiler, the _____ of a token of an identifier is a pointer to the identifier's symbol table entry containing a pointer to the _____ of the identifier.
- (d) The time complexity of lexical analysis using a _____ representation of lexical patterns is linear in the size of the input.
- (e) A lex/flex specification consists of three parts. The *second part* defines _____.
2. The textbook claims that "*The analysis part of a compiler breaks up the source program into pieces and creates an intermediate representation.*" What are the three different analysis phases performed by a compiler? (5 points)

3. What are the contents of a method *frame* in the JVM? (5 points)
4. Write a regular expression defining the representation of a signed decimal fraction consisting of an optional sign, a non-empty sequence of digits, followed by an optional decimal period and a non-empty sequence of digits. (5 points)
5. List the five *operations on languages* and give their set-based definitions. (10 points)

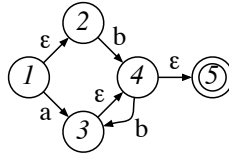
6. Consider the following Lex specification

```
digit      [0-9]
integer    {digit}({integer}|"")
%%
{integer}  { printf("integer: %s\n", yytext); }
%%
```

Show what is wrong with the regular definitions in this specification. Fix the specification so that it correctly scans integer literals. (10 points)

7. Use Thompson's algorithm to construct an NFA for the regular expression $(a|\varepsilon)b$ (10 points)

8. Given the NFA with $S = \{1, 2, 3, 4, 5\}$, $\Sigma = \{a, b\}$, $s_0 = 1$, $F = \{5\}$ and the transition graph portrayed below



convert the NFA to a DFA using the subset construction algorithm (do not attempt to minimize the DFA). Express your answer as a transition graph and identify the start and final states. (15 points)

9. Consider the following state transition table of a DFA with $S = \{0, 1, 2, 3, 4\}$, $\Sigma = \{\mathbf{a}, \mathbf{b}\}$, $s_0 = 0$, $F = \{3, 4\}$.

State	Input	
	a	b
0	1	2
1	3	4
2	1	2
3	4	-
4	3	-

- (a) Draw the transition graph. (5 points)
- (b) Minimize the DFA using the algorithm in the book and illustrated in class. Identify the start and final states of the minimized DFA. (15 points)
- (c) Write an equivalent regular expression that represents the same language as defined by the (minimized) DFA. (10 points)

10. **(Bonus question)** Consider the regular expression augmented with an endmarker #

$$(\mathbf{ab|c})\mathbf{a^*}\mathbf{\#}$$

- (a) Create the syntax tree of the regular expression and annotate the resulting tree with *nullable*, *firstpos*, and *lastpos* as defined in the book and illustrated in class. (5 points)
- (b) From the annotated syntax tree, create a table of *followpos* for each position. (5 points)
- (c) Construct the DFA for the regular expression. Identify the start and final states. (5 points)