

COP4020 Spring 2010 – Midterm Exam

Name: _____ (Please print)

Put the answers on these sheets. Use additional sheets when necessary. Show how you derived your answer (this is required for full credit and helpful for partial credit). You can collect 100 points in total for this exam.

1. Which of the following characteristics are considered essential (by the textbook) to make a programming language successful? (mark **none, one, or more**) (4 points)
 - (a) It is easy to use by a novice
 - (b) It is object oriented
 - (c) It is interpreted
 - (d) It has excellent compilers
2. To which programming language classification category belongs C? (mark **one**) (4 points)
 - (a) Logic or constraint-based
 - (b) Functional
 - (c) Dataflow
 - (d) Imperative, von Neumann
3. Type checking in statically typed programming languages is performed in. . . (mark **one**) (4 points)
 - (a) the scanning phase
 - (b) the parsing phase
 - (c) the semantic analysis phase
 - (d) the code generation phase
4. Which attributes in the following grammar are *synthesized*? (mark **none, one, or more**) (4 points)

$$\begin{array}{l} A \rightarrow \mathbf{a} B \quad A.a := B.x; \quad B.b := 'a' \\ B \rightarrow \mathbf{b} C \quad C.c := B.b + 'b'; \quad B.x := C.x \\ C \rightarrow \mathbf{c} \quad C.x := C.c \end{array}$$

- (a) $A.a$
 - (b) $B.b$
 - (c) $B.x$
 - (d) $C.c$
 - (e) $C.x$
5. Given the regular expression below, which strings are valid? (mark **none, one, or more**) (4 points)

$$R \rightarrow a^*(b \mid \varepsilon)b$$

- (a) a
- (b) b
- (c) ab
- (d) aabb

6. Which of the following is true? (mark **one**) (4 points)
- (a) An LL(1) parser's sentential derivation is always rightmost.
 - (b) An LL(1) parser cannot implement ambiguous grammars while an LR(1) parser always can (without disambiguating rules).
 - (c) An LL(1) parser can be constructed for any context-free grammar without modifications.
 - (d) An LL(1) parser decides the next parsing actions based on FIRST and FOLLOW sets (as part of the PREDICT set).
7. Andy and Bob are shopping for groceries. Andy knows how to program Prolog and Bob agrees to Andy's suggestion to use the following program to purchase items:

```
likes(andy, twinkies).
likes(andy, cake).
likes(andy, pie).
likes(andy, juice).
likes(bob, pie).
likes(bob, twinkies).
likes(bob, apples).
price(twinkies, 1).
price(cake, 5).
price(pie, 7).
price(juice, 2).
buy(X) :- likes(andy, X), \+ likes(bob, X), price(X, P), P < 3.
```

When searching **for all solutions** to the query

```
?- buy(X).
```

which items X are found on backtracking? (mark **one**) (4 points)

- (a) X = twinkies, X = cake, X = pie, X = juice.
- (b) X = cake, X = juice.
- (c) X = juice.
- (d) None

8. Consider the Scheme function:

```
(define dogwalk
  (lambda (arg)
    (cond
      ((null? arg) '())
      ((eq? (car arg) 'cat) '(chase-forever))
      ((eq? (car arg) 'pole) (cons 'leg-up (dogwalk (cdr arg))))
      (else (cons 'jump (dogwalk (cdr arg))))
    )
  )
)
```

Which one of the following function evaluations is correct? (mark **one**) (4 points)

- (a) (dogwalk '()) evaluates to (jump)
- (b) (dogwalk '(cat pole)) evaluates to (jump leg-up)
- (c) (dogwalk '(pole cat)) evaluates to (leg-up chase-forever)
- (d) (dogwalk '(cat puddle)) evaluates to (jump jump)

9. Construct a *recursive-descent* parser for the following grammar (10 points):

$$\begin{array}{l} S \rightarrow \mathbf{a} \\ \quad | X \mathbf{b} \\ \quad | \varepsilon \\ X \rightarrow \mathbf{c} X \\ \quad | \mathbf{d} \end{array}$$

Use C and the match(token) and lookahead() functions.

10. What is an *integrated development environment*? (7 points)

11. What is an AST? (7 points)

12. Given the following Prolog unification goal, what variables are bound to which values? (7 points)

```
person(bob, male, X) = person(Y, male, married_to(sue, female, Z))
```

13. In Project 1 we need to use the `let*` special form. Why is this? (8 points)

14. Consider the regular expression of question 5

$$R \rightarrow \mathbf{a^*(b \mid \varepsilon)b}$$

Construct an NFA or simplified DFA that accepts strings defined by R . (10 points)

15. Consider the grammar for the language that describes the moves and actions of a robot:

$$\begin{aligned}
 A_1 &\rightarrow M A_2 \\
 A &\rightarrow \varepsilon \\
 M &\rightarrow \mathbf{N} \\
 M &\rightarrow \mathbf{E} \\
 M &\rightarrow \mathbf{S} \\
 M &\rightarrow \mathbf{W} \\
 M &\rightarrow \mathbf{L}
 \end{aligned}$$

Where A represents a string of \mathbf{N} , \mathbf{E} , \mathbf{S} , \mathbf{W} (the robot's moves North, East, South, or West), and \mathbf{L} to let the robot linger at the current spot (no movement).

- Is this grammar LL(1)? Why or why not? (4 points)
- Draw the parse tree of " $\mathbf{L E N S}$ ". (5 points)
- Add semantic rules to keep track of the robot's position. To do so you need the following attributes:

$A.x$	the East-West coordinate (inherited)
$A.y$	the North-South coordinate (inherited)
$A.newx$	the East-West coordinate moved to (synthesized)
$A.newy$	the North-South coordinate moved to (synthesized)
$M.dx$	1 or -1 when moving East or West, respectively (synthesized)
$M.dy$	1 or -1 when moving North or South, respectively (synthesized)

For example, suppose we start with $A.x=A.y=0$ then after parsing the string $\mathbf{L E N S}$ we obtain $A.newx = 1$ and $A.newy = 0$ for the root nonterminal A (\mathbf{L} doesn't change x and y). (10 points)