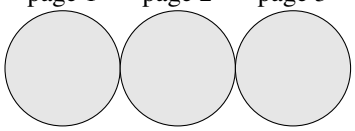
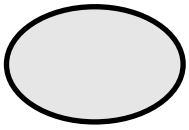
$Id: cmps104a-2010q4-exam3.mm,v 1.35 2010-12-02 14:34:49-08 - - $

page 1 page 2 page 3



Total / 52



Please print clearly :

Name :

Login : @ucsc.edu

No books ; No calculator ; No computer ; No email ; No internet ; No notes ; No phone. Neatness counts ! Do your scratch work elsewhere and enter only your final answer into the spaces provided.

1. Given the grammar presented here, and using the style from the LALR(1) handout :
   (a) Construct the characteristic finite state machine (CFSM), sets of items and transition diagram, showing shifts, reductions, and acceptance. **[6✔]**
   (b) Construct the FOLLOW sets. **[3✔]**
   (c) Answer ***yes*** or ***no*** to each of the following questions : **[1✔]**

   Is the grammar LR(0) ? _____ Is the grammar SLR(1) ? _____

0.	**S**	→	**$ M $**
1.	**M**	→	**{ X }**
2.	**X**	→	**X A**
3.	**X**	→	
4.	**A**	→	**i x M M**
5.	**A**	→	**x**

2. For the `bison` grammar on the left, fill in the table on the right. The contents of the box at the left end of each row indicates the rule (handle) at the top of the stack, and the symbol at the top of each column indicates the lookahead symbol. Write '**S**' if the appropriate action is a shift ; write '**R**' if the appropriate action is a reduce ; write '**X**' if something else should be done or if it is not possible to determine whether or not the action should be a shift or reduce. **[3✔]**

```
%left  '+' '-'
%right '*' '/'
%token NUM
%%
expr : expr '+' expr | expr '-' expr
     | expr '*' expr | expr '/' expr
     | '(' expr ')'  | NUM ;
%%
```

	`'*'`	`'/'`	`'+'`	`'-'`
`expr : expr '*' expr`				
`expr : expr '/' expr`				
`expr : expr '+' expr`				
`expr : expr '-' expr`				

3. Code (fast) `malloc`, assuming a copying collector with semispaces and one large chunk of free memory with `free` pointing at it start and `end` pointing at its end. Call the collector if needed. **[3✔]**

4. Write a `bison` grammar for a simple language, described here. **[4✔]**
   (a) A program is a sequence of zero or more elements.
   (b) An element is an `ATOM` or a list.
   (c) A list is a left parenthesis followed by zero or more elements, followed by a right parenthesis.
   (d) The scanner returns only one of three kinds of tokens : `ATOM`, `'('`, `')'`. Do not code the scanner.

   Use semantic actions to construct the entire program as a list, using the function `cons`. For example, `$$ = cons ($1, $2)` will take an already constructed list (`$2`) and a new node (`$1`) and return the list with the new node prepended to the list. Thus, your rules must be right associative, so that for any list, the tail is constructed first. Do not use `adopt` from the project.

5. Consider the following program.
   (a) To the right of the program, draw the abstract syntax tree as per the project specifications. **[3✔]**
   (b) Annotate the abstract syntax tree with attributes. Abbreviate each attribute using one letter beside each node in the AST: `C` (char), `I` (int), `V` (void), `*` (pointer), `**` (pointer to pointer), `L` (lval), `[]` (array), `T` (temporary). Ignore other attributes from the program, and only annotate those nodes that should have attributes. **[3✔]**
   (c) On the column of line numbers in the program, draw a circle around each set of number that represent a basic block. **[1✔]**
   (d) Translate the program into icode, putting icode into the particular box below that matches the line numbers in the program.
   (e) **[3✔]**

```
1.  int sum (int *a,
2.           int n) {
3.     int i;
4.     int s;
5.     i = 0;
6.     s = 0;
7.     while (i < n) {
8.        s = s + a[i];
9.        i = i + 1;
10.    }
11.    return s;
12. }
```

1.	7.
2.	8.
3.	9.
4.	10.
5.	11.
6.	12.

Multiple choice. To the ***left*** of each question, write the letter that indicates your answer. Write ***Z*** if you don't want to risk a wrong answer. Wrong answers are worth negative points. **[11✔]**

number of correct answers		× 1 =	$= a$
number of wrong answers		× ½ =	$= b$
number of missing answers		× 0 =	0
column total $c = \max(a - b, 0)$	11		$= c$

1. The symbol **`YYSTYPE`** denotes objects on the ____ stack.
   (A) lexical
   (B) semantic
   (C) state
   (D) symbol

2. The lookahead symbol in **`yyparse`** is declared as
   (A) `int yydebug;`
   (B) `int yylex;`
   (C) `YYSTYPE yylval;`
   (D) `char *yytext;`

3. Memory management most friendly to locality of reference is :
   (A) copying collector with semispaces
   (B) **`malloc`** and **`free`**
   (C) mark and sweep
   (D) reference counting

4. For a grammar $G = \langle V_N, V_T, P, S \rangle$, the grammar symbols returned by the scanner consist of elements of :
   (A) V_N
   (B) V_T
   (C) $V_N \cap V_T$
   (D) $V_N \cup V_T$

5. Which **`bison`** grammar will recognize any number of left parentheses followed by any number of right parentheses, provided that there are the same number of each, and will fail to recognize anything else ?
   (A) `b : '(' ')' b | ;`
   (B) `b : '(' b ')' b | ;`
   (C) `b : '(' b ')' | ;`
   (D) `b : '(' b | b ')' | ;`

6. If the following regular expressions are each converted into a DFA with the smallest number of possibl states for each, of the choices, which will necessarily result in the largest number of states ?
   (A) `a|b`
   (B) `a*`
   (C) `ab`
   (D) `(a)`

7. For a grammar $G = \langle V_N, V_T, P, S \rangle$:
   (A) $S \in V_N$
   (B) $S \in V_T$
   (C) $S \in P$
   (D) $P \in V_N \times V_T^*$

8. Which of the following context-free grammar rules shows that **`+`** is left associative ?
   (A) $E + T \rightarrow E$
   (B) $E \rightarrow E + E$
   (C) $E \rightarrow E + T$
   (D) $E \rightarrow T + E$

9. The following grammar :
   `A` → **`x A`**
   `A` →
   (A) is both LR(0) and LALR(1)
   (B) is neither LR(0) nor LALR(1)
   (C) is LR(0) but not LALR(1)
   (D) is LALR(1) but not LR(0)

10. Which of the following is permitted in an NFA but not a DFA ?
    (A) a set of input symbols
    (B) epsilon transitions
    (C) nonterminal symbols
    (D) undeclared identifiers

11. If function **`f`** contains function **`g`** nested inside of it, what will be used inside **`g`** to refer to the local variables of **`f`** ?
    (A) dynamic link
    (B) static link
    (C) frame pointer
    (D) stack pointer

Multiple choice. To the ***left*** of each question, write the letter that indicates your answer. Write ***Z*** if you don't want to risk a wrong answer. Wrong answers are worth negative points. **[11✔]**

number of correct answers		× 1 =	$= a$
number of wrong answers		× ½ =	$= b$
number of missing answers		× 0 =	0
column total $c = \max(a - b, 0)$	11		$= c$

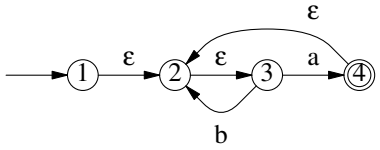
1. For a grammar $G = \langle V_N, V_T, P, S \rangle$, the grammar symbols on the parsing stack consist of elements of :
   (A) V_N
   (B) V_T
   (C) $V_N \cap V_T$
   (D) $V_N \cup V_T$

2. Which kind of memory management will not work if there are cycles in the data structure ?
   (A) copying collector with semispaces
   (B) `malloc` and `free`
   (C) mark and sweep
   (D) reference counting

3. The C declaration `int a[6];` will declare `a` as an lvalue in what context ?
   (A) a local variable
   (B) a global variable
   (C) a parameter
   (D) a field of a structure

4. The `flex` expression `a|bc*` means :
   (A) `(a|(bc))*`
   (B) `(a|b)(c*)`
   (C) `a|((bc)*)`
   (D) `a|(b(c*))`

5. Given the finite αυτοματον :



   (A) ε–closure(2) = {3}
   (B) ε–closure(2) = {1, 2}
   (C) ε–closure(2) = {1, 2, 3}
   (D) ε–closure(2) = {2, 3}

6. A sequence of instructions where if the first one is executed, then all of the rest of the instructions in that sequence will also be executed (except possibly if an exception occurs) is called :
   (A) basic block
   (B) dominator tree
   (C) function
   (D) natural loop

7. If an NFA is constructed from a regular expression whose length is $|r|$ and then used to scan a string whose length is $|s|$, the running time will be :
   (A) $O(|s|)$
   (B) $O(|s| + |r|)$
   (C) $O(|s|/|r|)$
   (D) $O(|s| \times |r|)$

8. According to the standard Unix memory map, the heap is an extension of what other segment ?
   (A) text
   (B) data
   (C) BSS
   (D) stack

9. For the frame pointer register `fp` and the stack pointer register `sp`, if `N` is the size of a local stack frame, then after saving `fp`, the following instructions will allocate the local stack frame :
   (A) `*sp++ -= N; fp = sp;`
   (B) `fp = *sp++; fp -= N;`
   (C) `fp = sp; sp -= N;`
   (D) `sp = fp; sp += N;`

10. The object-oriented function call in C++ coded as `a->f(x,y)` will be translated into machine code equivalent to which C expression :
    (A) `(a->f) (x, y)`
    (B) `(a->f) (a, x, y)`
    (C) `(a->classtable->f) (x, y)`
    (D) `(a->classtable->f) (a, x, y)`

11. In an expression tree :
    (A) operators are children of their operands
    (B) operators are leaf nodes
    (C) operators are parents of their operands
    (D) operators are root nodes