page 1 page 2 page 3 Total / 32

Please print clearly :

Name : Solution

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. Write a grammar in `bison` which will recognize an input stream of balanced parentheses. The only input tokens are the left parenthesis and the right parenthesis. Each left must match a corresponding right somewhere to the right of it. Explained another way, if you count parentheses from left to right where a left counts +1 and a right counts –1, the count may never go negative, and must be 0 in the end. The example at the left shows examples of balanced parentheses. **[1✔]**

```
(((((()))))
()(())()()
((((()()()))))
((())(())())
```

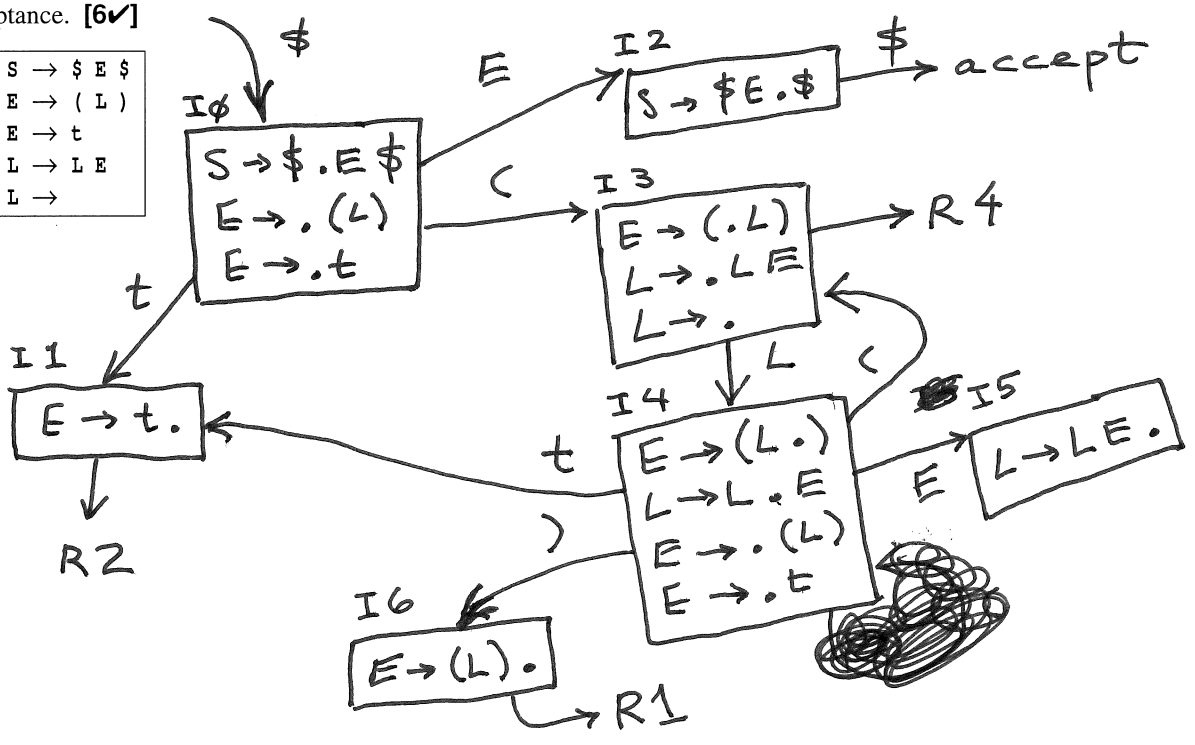
```
p: '(' p ')' p
 |
 ;
```

2. Write an unambiguous grammar using `bison` where an expression is a sequence of terms connected by at sign (`@`) operators. A term is a sequence of tokens connected by percent (`%`) operators. A token is either an identifier or a number. Both operators are right associative. Do not show semantic actions. **[3✔]**

```
expr: term '@' expr
    | term
    ;
term: token '%' term
    | token
    ;
token: IDENT
     | NUMBER
     ;
```

3. Given the grammar presented here, and using the style of the handout, use LR(0) analysis to construct the characteristic finite state machine (CSFM), sets of items and transition diagram, showing shifts, reductions, and acceptance. **[6✔]**

```
0.  S → $ E $
1.  E → ( L )
2.  E → t
3.  L → L E
4.  L →
```



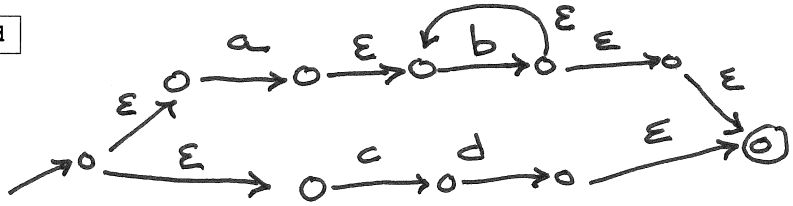
4. Define a grammar using `bison` for a language described as follows. **[4✓]**
   (a) A program consists of zero or more statements, each terminated with a semi-colon.
   (b) A statement is a function call.
   (c) A function call is an identifier, followed by a left parenthesis, followed by zero or more arguments separated by commas, followed by a right parenthesis.
   (d) An argument is a function call, an identifier, or a number.

```
%start prog
%token ID NUM
%%
prog : prog stmt
     |
     ;
stmt : call ';'
     ;
call : ID '(' args ')'
     | ID '(' ')'
     ;
```

```
args: args ',' arg
    | arg
    ;
arg : call
    | ID
    | NUM
    ;
```

5. Using Thompson's construction exactly, without optimization or minimization, draw the non-deterministic finite state machine given by the following `flex` regular expression. **[2✓]**

`ab*|cd`



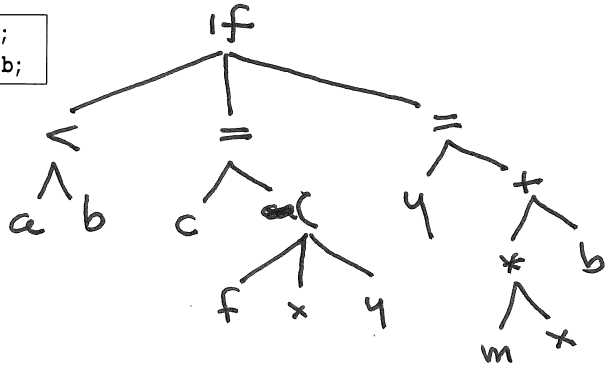
6. There are two kinds of things permitted in an NFA that are prohibited in a DFA. What are they? For each, draw a very small NFA that illustrates it. **[2✓]**

ε - transitions

same symbol leaving a state on more than one outgoing transition

7. Draw the abstract syntax tree as per the project 3 specifications for the following code. **[2✓]**

```
if (a < b) c = f (x, y);
      else y = m * x + b;
```



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. **[12✓]**

number of correct answers		× 1 =	$= a$
number of wrong answers		× ½ =	$= b$
number of missing answers		× 0 =	0
column total $c = \max(a-b, 0)$	12		$= c$

1. The variable that communicates semantic information from `yylex()` to `yyparse()` is :
   C
   (A) `yydebug`
   (B) `yyleng`
   (C) `yylval`
   (D) `yytext`

2. Which is an example of an imput recognized by the following grammar ?
   $A \rightarrow xAy$
   $A \rightarrow$
   B
   (A) *xxxxxxxxxy*
   (B) *xxxxxyyyyy*
   (C) *xyxyxyxyxy*
   (D) *xyyyyyyyyy*

3. If an NFA is constructed from a regex r and is used to scan a string s, then its memory requirement M and running time T are :
   B
   (A) $M = O(2^{|r|})$; $T = O(|s|)$
   (B) $M = O(|r|)$; $T = O(|r| \times |s|)$
   (C) $M = O(|r| \times |s|)$; $T = O(|r|)$
   (D) $M = O(|s|)$; $T = O(2^{|r|})$

4. For unambiguous grammars, what kind of analysis is performed by `bison` ?
   A
   (A) LALR(1)
   (B) LL(k)
   (C) LR(0)
   (D) recursive descent

5. Which rule shows that `=` is a right-associative operator ?
   C
   (A) $E \rightarrow E = E$
   (B) $E \rightarrow E = T$
   (C) $E \rightarrow T = E$
   (D) $E = E \rightarrow E$

6. What parsing action pops some number of elements (equal to the length of the right-hand side of a rule) off the parsing stack and then pushes a non-terminal onto the stack ?
   B
   (A) accept
   (B) reduce
   (C) scan
   (D) shift

7. If a regular expression r of length $|r|$ is converted into a deterministic automaton and then used to scan a string s of length $|s|$, how fast will it run ?
   C
   (A) $O(|r|)$
   (B) $O(|r| \times |s|)$
   (C) $O(|s|)$
   (D) $O(|s| + |r|)$

8. To solve the problem of the dangling else with an ambiguous grammar, `else` should be declared as ____-associative and given a precedence ____er than the arithmetic operators.
   D
   (A) left-associative, higher than
   (B) left-associative, lower than
   (C) right-associative, higher than
   (D) right-associative, lower than

9. Given the semantics of `adopt1` from the provided code, what is the appropriate action for the rule
   `expr : expr '+' expr`
   C
   (A) `{ $$ = adopt1 ($1, $2, $3); }`
   (B) `{ $$ = adopt1 ($1, $3, $2); }`
   (C) `{ $$ = adopt1 ($2, $1, $3); }`
   (D) `{ $$ = adopt1 ($3, $2, $1); }`

10. If `command` contains the Unix command to start the preprocessor, what will be used to allow reading its output ?
    C
    (A) `yyin = fopen (command, "r");`
    (B) `yyin = fopen (command, "w");`
    (C) `yyin = popen (command, "r");`
    (D) `yyin = popen (command, "w");`

11. As a semantic action embedded in `yylex()`, which is obviously wrong ?
    D
    (A) `{ return '+'; }`
    (B) `{ return *yytext; }`
    (C) `{ return PLUS; }`
    (D) `{ return "+"; }`

12. If a DFA is constructed from each of the following regular expressions, then minimized, which will have the fewest states ?
    B
    (A) `x`
    (B) `x*`
    (C) `x+`
    (D) `x?`