

page 1

page 2

page 3

page 4

page 5

Total / 54

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. Given the grammar presented here, and using the style from the LALR(1) handout:

- Construct the characteristic finite state machine (CFSM), sets of items and transition diagram, showing shifts, reductions, and acceptance. [6✓]
- Construct the FOLLOW sets. [3✓]
- Answer yes or no to each of the following questions: [1✓]

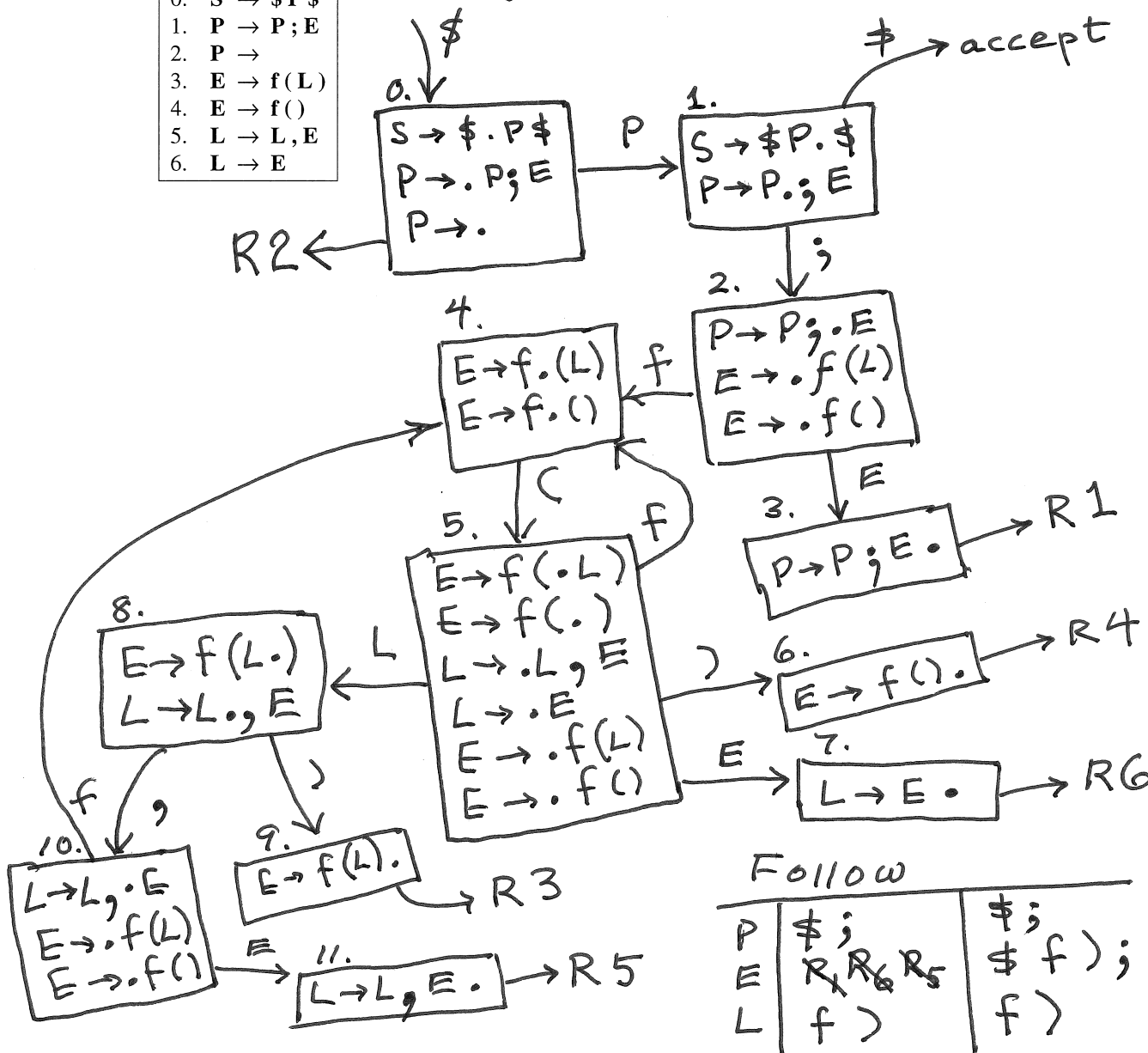
Is the grammar LR(0)?

yes

Is the grammar SLR(1)?

yes

0. $S \rightarrow \$ P \$$
1. $P \rightarrow P ; E$
2. $P \rightarrow$
3. $E \rightarrow f(L)$
4. $E \rightarrow f()$
5. $L \rightarrow L, E$
6. $L \rightarrow E$



2. Assume free space is one large chunk of memory delimited by the **free** pointer at the lower end and the **end** pointer at the upper end. Code a version of **malloc** which allocates the number of bytes given by its argument and updates **free** to point just past the area allocated. The amount of space allocated must be a multiple of 16. [3✓]

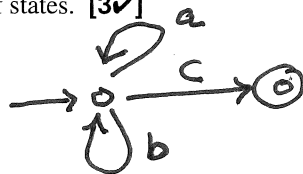
```
void* malloc (size_t n) {
    n = (n + 15) & ~15;
    if (free + n > end) gc();
    t = free;
    free += n;
    return t;
}
```

3. After a call instruction, on entry to a function, what machine instructions are executed to save the frame pointer and allocate enough space for N bytes on the stack? What instructions are executed just prior to a return instruction to remove the local stack frame from the function call stack? The frame pointer is **rbp** and the stack pointer is **rsp**. [2✓]

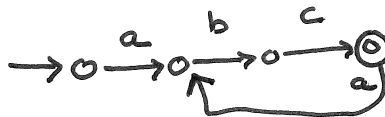
Code on entry to the function	Code just before the return instruction
<pre>push rbp movq rbp, rsp sub N, rsp</pre>	<pre>movq rsp, rbp pop rbp ret</pre>

4. Draw deterministic finite αὐτόματα for each of the following **flex** regular expressions. Use the minimum possible number of states. [3✓]

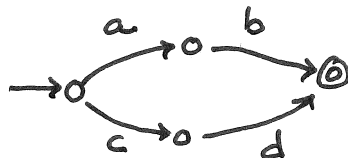
(i) $a|b)^*c$



(ii) $(abc)^+$



(iii) $ab|cd$



5. Using **bison** syntax, define a context free LALR(1) grammar for the following language: [2✓]

- (a) A program is a sequence of zero or more expressions, each expression separated from the next by a semi-colon (;). The last expression is **not** followed by a semi-colon.
 (b) An expression is an identifier followed by an argument list enclosed in parentheses.
 (c) An argument list consists of a sequence of zero or more expressions. If there are two or more expressions in the argument list, consecutive expressions are separated by commas (,).

```
prog : exprs
```

```
exprs : exprs ';' exprs
      | expr
```

```
expr : ID '(' args ')'
      | ID '(' ')'
      ;
```

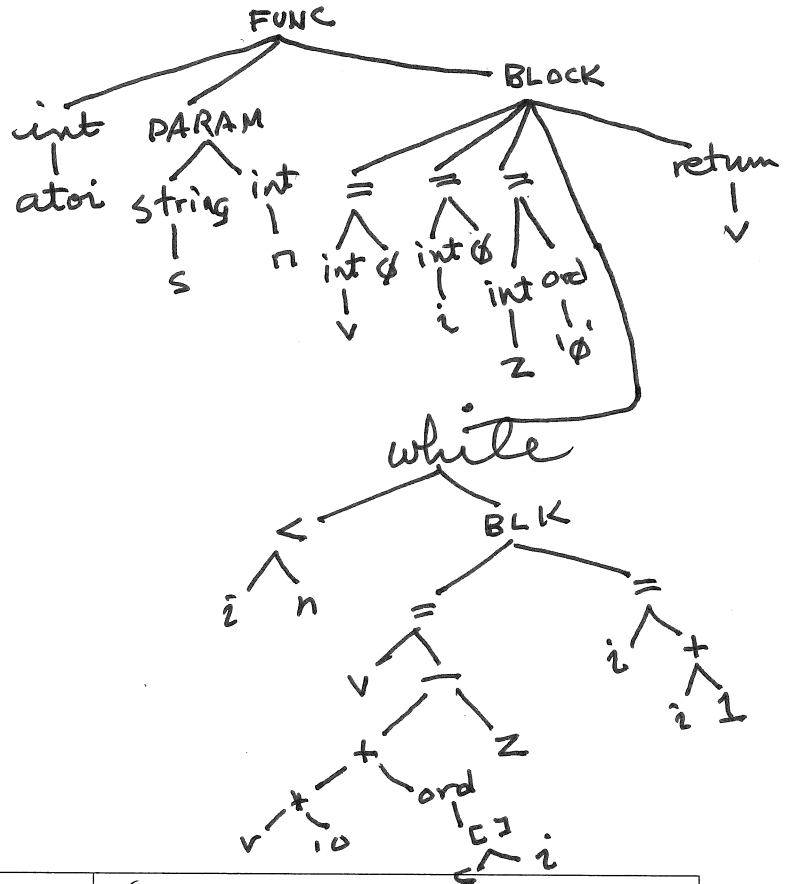
```
args : expr ',' expr
      |
```

6. Given the function presented here, and following the project specifications :

- Draw the abstract syntax tree as for project 3. Do not annotate with attributes. Draw the tree in the empty space to the right of the first two tables. [4✓]
- Fill in the the symbol table, with associated attributes. Just use plain names, without mangling. Attributes are: void, bool, char, int, null, string, struct, array, function, variable, field, typeid, param, lval, const, vreg, vaddr. [2✓]
- Show the translated oil code. Just use plain names, without mangling. The table at the bottom of the page shows numbered boxes. Write the oil code in each box at the bottom of the page whose number corresponds to the line numbers in the program. Name all temporaries (registers) as t1, t2, t3, etc., without declaring their specific types. [4✓]

1.	int atoi (string s, int n) {
2.	int v = 0;
3.	int i = 0;
4.	int z = ord '0';
5.	while (i < n) {
6.	v = v * 10 + ord s[i] - z;
7.	i = i + 1;
8.	}
9.	return v;
10.	}

symbol	attributes
atoi	int func.
s	string var param lval
n	int var param lval
v	int var lval
i	int var lval
z	int var lval



1.	int atoi (6.	t3 = v * 10;
	string s,		t4 = &s[i];
	int n)		t5 = (int)t4;
	{		t6 = t3 + t5;
			t7 = t6 - z;
2.	int v = 0;	7.	v = t7;
			t8 = i + 1;
3.	int i = 0;		i = t8;
		8.	goto while5;
4.	t1 = (int)'0';		
	int z = t1;	9.	return v;
5.	while 5:		
	t2 = i < n;	10.	}
	if (!t2) goto break5;		

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		$\times 1 =$	$= a$
number of wrong answers		$\times \frac{1}{2} =$	$= b$
number of missing answers		$\times 0 =$	0
column total	12		$= c$
$c = \max(a - b, 0)$			

1. Garbage collection might be expected to slow down a program by about how much?

(A) 10%
(B) 50%
(C) 100%
(D) 200%

A

2. For object oriented programming with single inheritance only, what is the expected space overhead per object allocated?

(A) one pointer per class table
(B) one pointer per instance field
(C) one pointer per object
(D) one pointer per virtual function

C

3. Given the declarations `int i` and `int* p`, which expression is in error?

(A) `i+p`
(B) `i-p`
(C) `p+i`
(D) `p-i`



B

4. In a classical Unix memory map of a process, which segment will be allocated to the highest addresses?

(A) data
(B) heap
(C) stack
(D) text

C

5. If N is the set of languages recognizable by a nondeterministic finite automaton and D is the set of languages recognizable by a deterministic finite automaton, then:

(A) $N \subset D$
(B) $N = D$
(C) $N \supset D$
(D) none of the above

B

6. On the x86-64, what is the alignment requirement (in bytes) for `rsp` and `rbp`?

(A) 0x10
(B) 0x1000
(C) 0x100000
(D) 0x10000000

A

7. On the x86-64, what is the page size (in bytes)?

(A) 16
(B) 4096
(C) 1048576
(D) 268435456

B

8. The virtual address of a static variable is determined at _____ time.

(A) compile
(B) link
(C) exec
(D) debugging

B

9. The largest class of grammars whose characteristic finite state machine (CFSM) is the same size as the LR(0) CSFM is:

(A) LALR(1)
(B) LR(0)
(C) LR(1)
(D) SLR(1)

A

10. Which grammar is unambiguous, allows for an arbitrary large number of `xs`, and makes the + operator left associative?

(A) $E \rightarrow E + x \mid x$
(B) $E \rightarrow E + E \mid x$
(C) $E \rightarrow x + E \mid x$
(D) $E \rightarrow x + x \mid x$

A

11. Which of the following items was added to a set of items during a closure operation?

(A) $E \rightarrow \bullet E + T$
(B) $E \rightarrow E \bullet + T$
(C) $E \rightarrow E + \bullet T$
(D) $E \rightarrow E + T \bullet$

A

12. Which of the following items will cause all rules of the form $T \rightarrow \beta$ to be added as items of the form $T \rightarrow \bullet \beta$, where $\beta \in V^*$?

(A) $E \rightarrow \bullet E + T$
(B) $E \rightarrow E \bullet + T$
(C) $E \rightarrow E + \bullet T$
(D) $E \rightarrow E + T \bullet$

C

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		$\times 1 =$	$= a$
number of wrong answers		$\times \frac{1}{2} =$	$= b$
number of missing answers		$\times 0 =$	0
column total $c = \max(a - b, 0)$	12		$= c$

1. The register that points at the local data area of the function in which the current function is nested is called :

(A) dynamic link
(B) frame pointer
(C) return address
(D) static link

2. In a typical implementation of `malloc(3)`, what is the usual boundary tag overhead per allocation ?

(A) 2 bytes
(B) 2 pages
(C) 2 pointers
(D) 2 strings

3. If parameters are passed on the stack, as in the x86 32-bit address space, what might be a reasonable address of one of the incoming parameters ?

(A) $+8(\text{ebp})$
(B) $+8(\text{esp})$
(C) $-8(\text{ebp})$
(D) $-8(\text{esp})$

ambiguous:
(B) on fn entry
(A) after alloc local frame

4. If `malloc(3)` runs out of heap space, what system call can be used to request more space from the kernel ?

(A) `brk(2)`
(B) `creat(2)`
(C) `execve(2)`
(D) `fork(2)`

5. In garbage collection terminology, reachable objects are :

(A) all variables local to any function on the function call stack.
(B) objects in a reference counting environment excluding cyclic references.
(C) objects on the heap accessible from the closure of the root set.
(D) static variables local to a given file, excluding external variables.

6. What will detect the error in the following Java statements ?

```
Object o = new Integer (3);
String s = (String) o;
```

(A) the parser
(B) the type checker
(C) the code generator
(D) the runtime library

7. The Java virtual machine (JVM) interpreter uses what kind of intermediate format ?

(A) abstract syntax tree
(B) stack machine code
(C) three-address code
(D) two-address code

8. Given that $V = V_N \cup V_T$, if a grammar has rules of the form $A \rightarrow \beta$, then

(A) $A \in V_N$ and $\beta \in V^*$
(B) $A \in V$ and $\beta \in V_N^+$
(C) $A \in V$ and $\beta \in V^+$
(D) $A \in V_T$ and $\beta \in V^*$

9. Putting reserved words into a **flex** scanner instead of recognizing them as identifiers and looking them up in a separate table will have what effect ?

(A) Makes the DFA run more quickly.
(B) Makes the DFA run more slowly.
(C) Makes the DFA take up less memory.
(D) Makes the DFA take up more memory.

10. What is prohibited in a deterministic finite $\alpha\upsilon\tau\acute{o}\mu\alpha\tau\omicron\nu$?

(A) ϵ -transitions
(B) cycles in the graph
(C) multiple final states
(D) all of the above

11. The following grammar is :

```
A → A + A
A → A * A
A → x
```

(A) LALR(1)
(B) LL(1)
(C) ambiguous
(D) regular

12. Which is an example of something recognizable by the following grammar ?

```
A → A x | y
```

(A) xxxxxxxxy
(B) xxxxxxxxy
(C) xxxxxxxxx
(D) xxxxxxxyx