CS131 Compilers: Midterm Exams

Chinese Name:____

Pinyin Name:____

Student ID:____

E-Mail ... @shanghaitech.edu.cn::____________________________

Question	Points	Score
1	10	
2	30	
3	40	
4	90	
5	10	
Total:	180	

- This test contains 8 numbered pages, including the cover page, printed on both sides of the sheet.
- We will use gradescope for grading, so only answers filled in at the obvious places will be used.
- Use the provided blank paper for calculations and then copy your answer here.
- Please turn **off** all cell phones, smartwatches, and other mobile devices. Remove all hats and headphones. Put everything in your backpack. Place your backpacks, laptops and jackets out of reach.
- You have 120 minutes to complete this exam. The exam is open book, but no computers, phones, or calculators are allowed. You may use any number of A4 pages (front and back) of handwritten or printed notes in addition to the Dragon Book.
- There may be partial credit for incomplete answers; write as much of the solution as you can. We will deduct points if your solution is far more complicated than necessary. When we provide a blank, please fit your answer within the space provided.
- Do **NOT** start reading the questions/ open the exam until we tell you so!

1. (10 points) Fill in the blanks in the first three parts using **Lexical Analysis**, **Syntax Analysis** or **Semantic Analysis** (for the COOL compiler).

   (a) (2 points) Which phase of a compiler may generate a syntax error?

   (a) **Syntax Analysis**

   (b) (2 points) Which phase of a compiler may generate an error of undefined variables.

   (b) **Semantic Analysis**

   (c) (2 points) If you try to add a real-valued number to an integer variable, which compiler phase would generate an error?

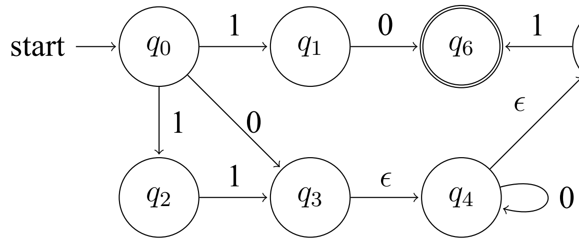
   (c) **Semantic Analysis**

   (d) (4 points) Please write a regular definition for the unsigned numbers in C.

   Solution: $digits \rightarrow digit^{+}$
   $opt\_fraction \rightarrow (.digit)?$
   $opt\_exponent \rightarrow (E(+|-)?digits)?$
   $unsigned\_num \rightarrow digits\ opt\_fraction\ opt\_exponent]$

2. (15 points) (a) (5 points) Construct an NFA of the Regular Expression $10|((0|11)0^*1)$.

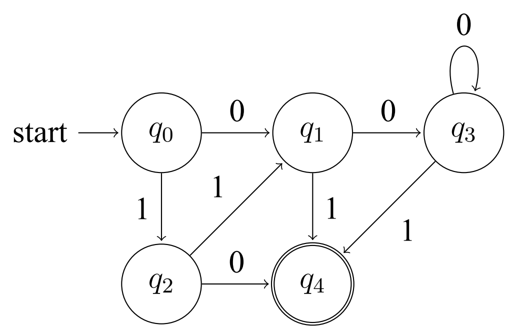
Solution:



(b) (5 points) Construct the subset states and corresponding DFA of the above NFA

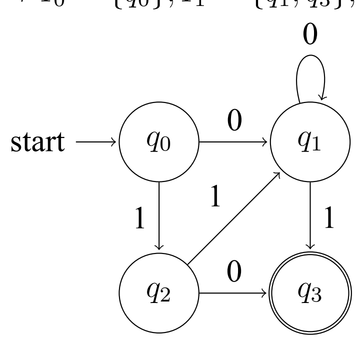
Solution:

Subset state	DFA state
$\{q_0\}$	q_0
$\{q_3, q_4, q_5\}$	q_1
$\{q_1, q_2\}$	q_2
$\{q_4, q_5\}$	q_3
$\{q_6\}$	q_4



(c) (5 points) Minimize the above DFA

Solution: I_0=$\{q_0, q_1, q_2, q_3\}$, $I_4 = \{q_4\}$
$\rightarrow I_0 = \{q_0, q_1, q_3\}, I_2 = \{q_2\}, I_4 = \{q_4\}$
$\rightarrow I_0 = \{q_0\}, I_1 = \{q_1, q_3\}, I_2 = \{q_1, q_3\}, I_4 = \{q_4\}$



3. (20 points) Consider the language $G = \{$binary sequences that can be divided by 5$\}$. For example, 00/5=0, 1010/5=2 and 00001010/5=2, thus $00, 1010, 00001010 \in G$.

(a) (6 points) Write down the context-free grammar for G

Solution: $S \to 0S|1A|\varepsilon$
$A \to 0B \mid 1C$
$B \to 1S \mid 0D$
$C \to 0A \mid 1B$
$D \to 0C \mid 1D$

(b) (7 points) Write down the FIRST sets and FOLLOW sets for the above grammar, and give the LL(1) parsing table

Solution: $First(S) = \{0, 1\epsilon\}$
$First(A) = \{0, 1\}$
$Follow(S) = Follow(A) = Follow(B) = Follow(C) = Follow(D) = \{\$\}$

	0	1	\$
S	$S \to 0S$	$S \to 1A$	$S \to \epsilon$
A	$A \to 0B$	$A \to 1C$	
B	$B \to 0D$	$B \to 1S$	
C	$C \to 0A$	$C \to 1B$	
D	$D \to 0C$	$D \to 1D$	

(c) (7 points) Write down the recursive predictive parsing program for the above grammar (with one look ahead)

Solution:

```
void match(terminal t){
    if (lookahead==t) lookahead = nextToken();
    else error();
}
void S(){
    if (lookahead=='0'){match("0");S();}
    else if(lookahead=='1'){match("1");A();}
    else if(lookahead=='$'){succeed();}
    else error();
}
void A(){
    if (lookahead=='0'){match("0");B();}
    else if(lookahead=='1'){match("1");C();}
```

```
    else error();
}
void B(){
    if (lookahead=='0'){match("0");D();}
    else if(lookahead=='1'){match("1");S();}
    else error();
}
void C(){
    if (lookahead=='0'){match("0");A();}
    else if(lookahead=='1'){match("1");B();}
    else error();
}
void D(){
    if (lookahead=='0'){match("0");C();}
    else if(lookahead=='1'){match("1");D();}
    else error();
}
```

4. (45 points) Consider the following two grammars G_1 and G_2, where one of them is an ambiguous grammar and the another one is a non-ambiguous grammar.

G_1, S is the start symbol
S $\to aBS \mid bAS$
$S \to \varepsilon$
$A \to a \mid bAA$
$B \to b \mid aBB$

G$_2$, S is the start symbol
S $\to aB \mid bA$
$S \to \varepsilon$
$A \to aS \mid bAA$
$B \to bS \mid aBB$

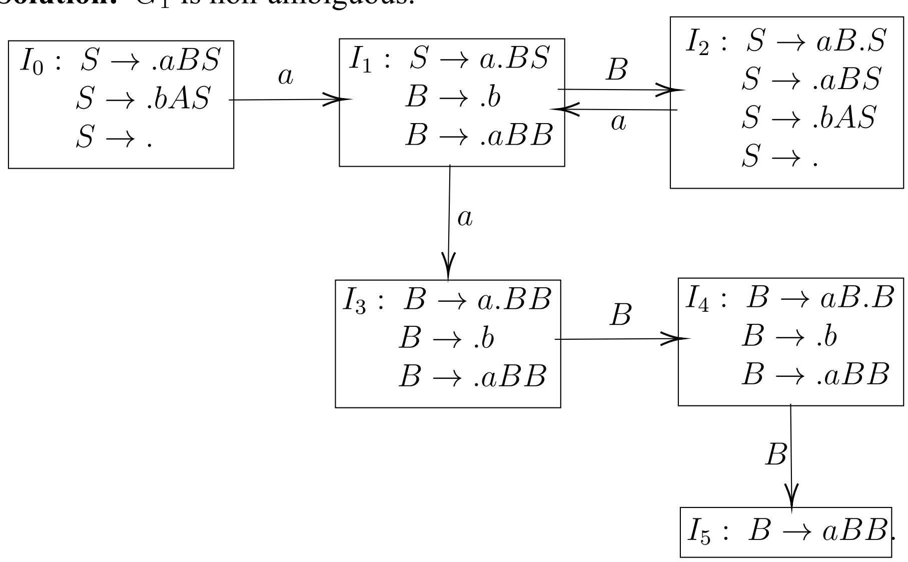
(a) (5 points) Which grammar is ambiguous? Use $aababb$ to prove its ambiguity.

Solution: G_2 is ambiguous, because it has 2 left most deriviation.
$S =>_{lm} aB =>_{lm} aaBB =>_{lm} aabSB =>_{lm} aabB =>_{lm} aabaBB =>_{lm} aababSB =>_{lm} aababB =>_{lm} aababbS =>_{lm} aababb$

$S =>_{lm} aB =>_{lm} aaBB =>_{lm} aabSB =>_{lm} aabaBB =>_{lm} aababSB =>_{lm} aababB =>_{lm} aababbS =>_{lm} aababb$

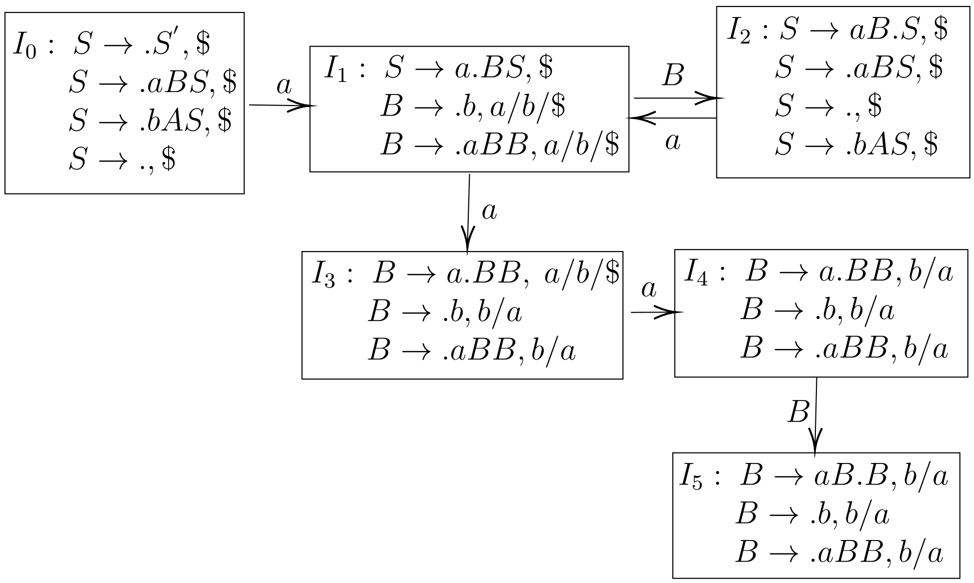
(b) (10 points) For the non-ambiguous grammar, write down the fragment of the LR(0) automaton such that this fragment accepts the input word $aBaaBB$. Here fragment means some states and some transitions between states of the LR(0) automaton.

Solution: G_1 is non-ambiguous.



(c) (15 points) For the non-ambiguous grammar, write down the fragment of the LR(1) automaton such that this fragment accepts the input word $aBaaaB$. Here fragment means some states and some transitions between states of the LR(1) automaton.

Solution:

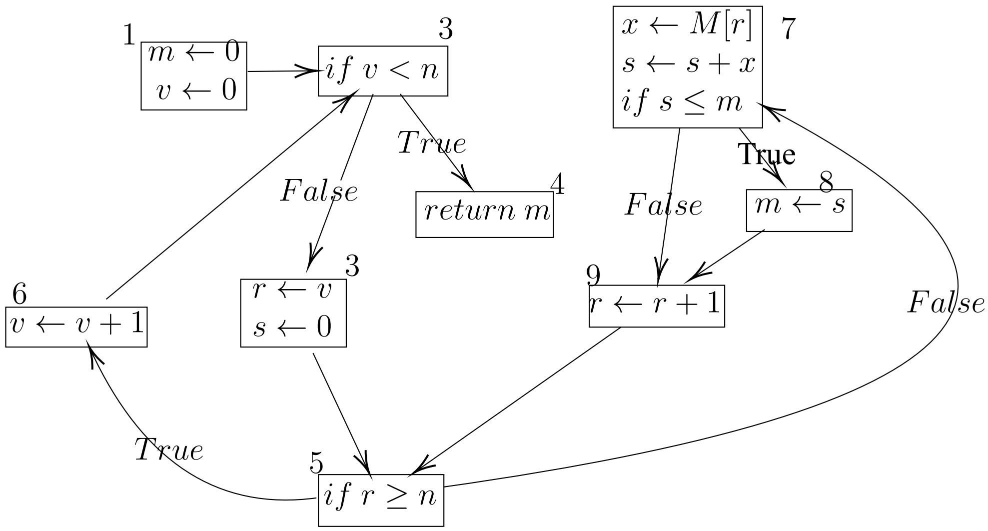


(d) (15 points) Is the non-ambiguous grammar in LALR(1)? If there exist any conflicts, write down the states that have at least one conflict, else write down the LALR Action Table and Goto Table.

Solution:

state	Action Table			Goto Table			
	a	b	$\$$	S'	S	A	B
0	s_2	s_3	r_2		1		
1			acc				
2	s_6	s_5					4
3	s_8	s_9				7	
4	s_2	s_3	r_2		10		
5	r_6	r_6	r_6				
6	s_6	s_5					11
7	s_2	s_3	r_2		12		
8	r_4	r_4	r_4				
9	s_8	s_9				13	
10			r_1				
11	s_6	s_5					14
12			r_3				
13	s_8	s_9				15	
14	r_7	r_7	r_7				
15	r_5	r_5	r_5				

5. (10 points) Transform the program in the following control flow graph into SSA form.



Solution:

m_1=0
v_1=0
3: v_3=$\phi(v_1, v_2)$
if v_3<n
  return m_1
r_1=v_3
s_1=0
5: r_3=$\phi(r_1, r_2)$
if $r_3 \geq n$
  v_2=v_3+1
  goto 3
7: x_1=M[r_3]
s_2=s_1+x_1
if $s_2 \leq m_1$
  m_1=s_2
9: r_1=r_1+1
goto 5