第一章

手工整理，有误私聊

一、单项选择题

1、C,B,C,D
2、A,B,C,D
3、B
4、B,E,E,D,A
5、C,D,F
6、D,D
7、B
8、A
9、A
10、A,B

二、填空题

1、是否生成目标代码
2、执行性语句、说明性语句
3、源程序、单词符号

三、名词解释

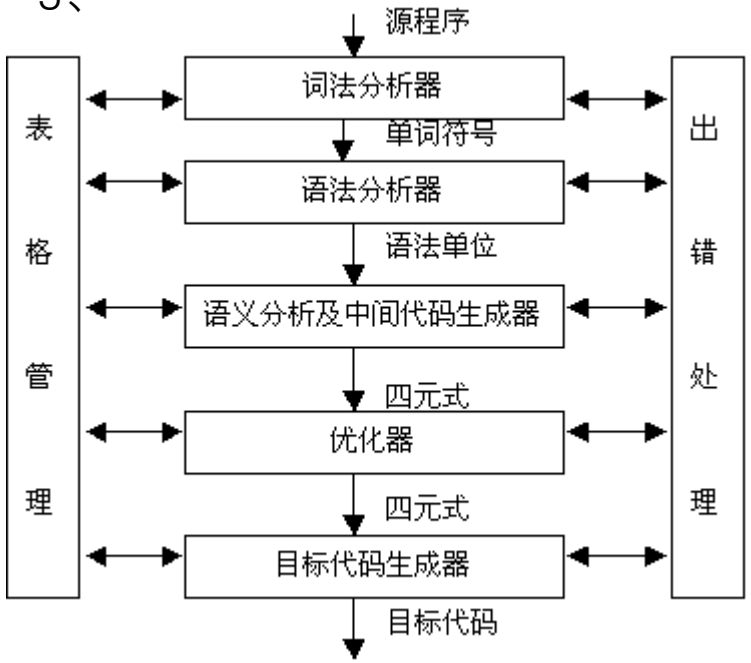
编译程序：也称翻译程序，将源程序译成逻辑上等价的目标程序的程序。有两种工作方式：编译和解释。
语义：指这样的一组规则，使用它可以规定一个程序的意义。
语法：指这样的一组规则，使用它可以形成和产生一个合式的程序。
遍：对源程序或源程序的中间结果从头到尾扫描一次，并做有关的加工处理，生成新的中间结果或目标程序。

四、简答题

1、词法分析、语法分析、语义分析、中间代码生成、中间代码优化、目标代码生成、错误处理、表格管理。

2、前端：语言结构和意义的分析（中间代码之前）；后端：语言意义处理（中间代码之后）。

3、



P6

词法分析器，又称扫描器，它接受输入的源程序，对源程序进行词法分析，识别出一个个的单词符号。

语法分析器对单词符号串进行语法分析（根据语法规则进行推导或归纳），识别出程序中的各类语法单位，最终判断输入串是否构成语法上正确的“程序语句”。

语义分析及中间代码产生器，按照语义规则对语法分析器归纳出（或推导出）的语法单位进行语义分析并把它们翻译成一定形式的中间代码。

优化器对中间代码进行优化处理。其过程实际上是对中间代码进行等价替换，使程序在执行时能更快，并占用更小的空间。

目标代码生成器把中间代码翻译成目标程序。中间代码一般是一种与机器无关的表示形式，只有把它再翻译成与机器硬件相关的机器能识别的语言，即目标程序，才能在机器上运行。

表格管理模块保持一系列的表格，登记源程序的各类信息和编译各阶段的进展状况。编译程序各个阶段所产生的中间结果都记录在表格中，所需要的信息也大多从表格中，所需要的信息也大多从表格中获取，整个编译过程都在不断地和表格打交道。

出错处理程序对出现在源程序中的错误进行处理。编译程序的各个阶段都有可能发现错误，出错处理程序要对发现的错误进行处理、记录，并反映给用户。

4、考虑因素：语言的大小与结构、机器规模、设计目的。

优点：结构清晰、构造时间短、运行时需内存少、产生的目标代码质量高。

缺点：时间效率低。

第二章

一、填空题

1、单词符号
2、记号
3、语言
4、非确定型有限自动机（NFA）
5、定义、词法规则、用户子程序

二、选择题

1、C
2、含奇数个0 且 每个以1结尾的前缀均含奇数个0　　0010✖　　0100✔
3、B

三、判断题

1、✖
2、✔
3、✔

四、名词解释

正规式：一种表示正规集的工具，是描述程序语言单词的表达式。
非确定型有限自动机（NFA）：在一个状态下，对于同一字符，可能有若干个下一个状态的转移的有限自动机。

五、问答题

1、① 滤掉源程序中的无用成分，如注释、空格、回车等。
② 处理与平台有关的输入，如文件结束符的不同表示等。
③ 根据模式识别记号，并交给语法分析器。
④ 调用符号表管理器或出错处理器，进行相关处理。

2、① 直接编码的词法分析器，将DFA和DFA识别输入序列的过程合并在一起，直接用程序代码模拟DFA识别输入序列。

② 先构造DFA，再由DFA构造表驱动型的词法分析器。

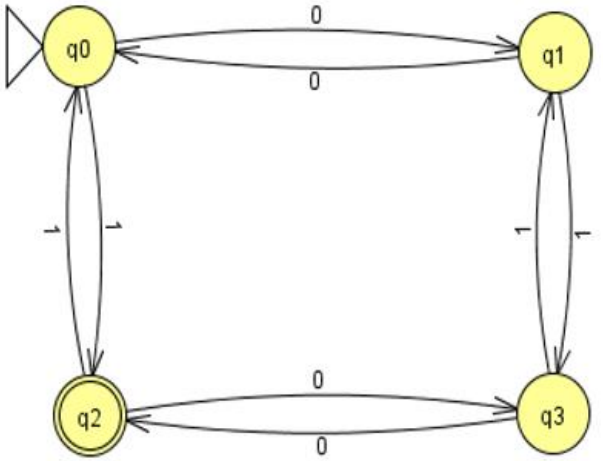
③ 使用自动生成工具Lex等。

3、 DFA是NFA的一个特例，其中：

① 没有 ε 状态转移。

②同一状态下没有重复字符的状态转移。

4、所有由偶数个0和偶数个1构成的串。



第4题

5、

(1) 设正规式 A 表示的正规集为 L(A)。

因为 $L(A|A) = L(A) \cup L(A) = L(A)$

所以 $A|A = A$

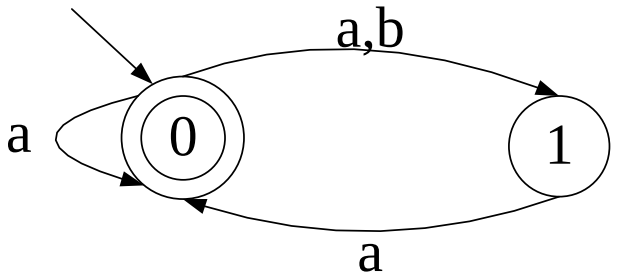
(2) 因为 $(A^*)^* = A^{**}$ $A^{**} = A^*$

所以 $(A^*)^* = A^*$

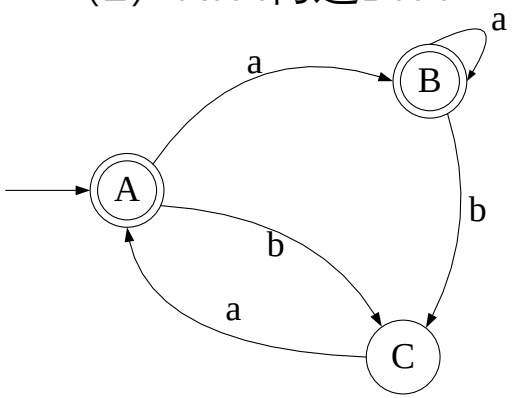
(3) 因为 $AA^* = A^+$ $A^* = A^+ | \varepsilon = \varepsilon | A^+$

所以 $A^* = \varepsilon | AA^*$

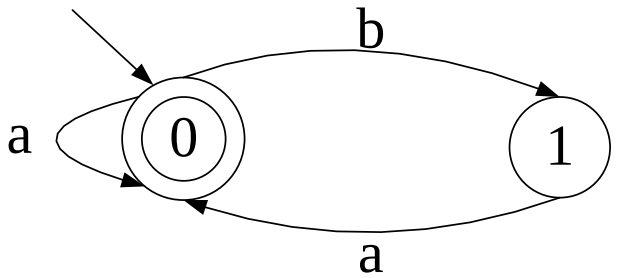
6、 （1）NFA



（2）NFA构造DFA



最小化DFA



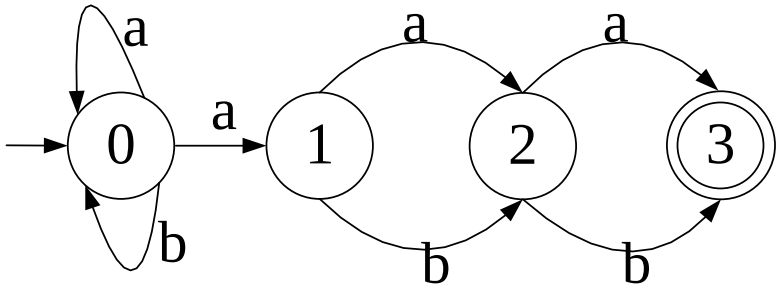
（3）(a|ba)*

（4）a：0a0

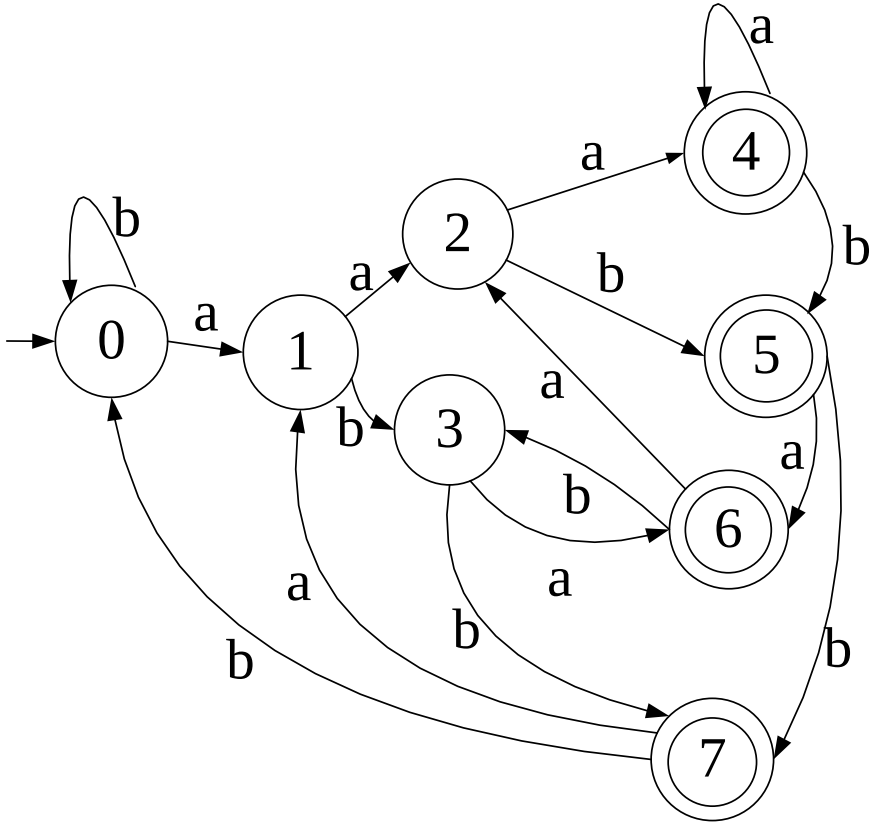
ba：0b1a0

baa：0b1a0a0

7、（1）NFA



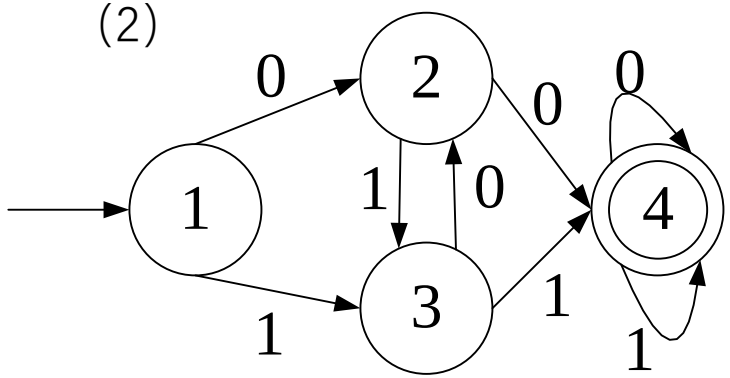
（2）NFA构造DFA



（3）最小化DFA，同（2）

8、（1）(0| 1)*(00|11)(0|1)*

（2）



（3）令 A、B、C、D分别对应状态1、2、3、4，则有正规文法如下：

A → 0B | 1C

B → 1C | 0D

C → 0B | 1D

D → 0D | 1D | ε

第三章

一、填空题

1、终结符集、非终结符集、产生式集合、开始符号
2、句型
3、句子
4、自上而下、自下而上
5、多步、直接
6、可归约串、左部
7、直接归约、归约、开始符号
8、自左向右扫描输入串、最左规约、向前看0个终结符

二、选择题

1、B
2、C
3、C
4、B
5、B
6、D

三、判断题

1、✖
2、✖
3、✖
4、✖
5、✔

四、名词解释

上下文无关文法：0型语法 + G的任一产生式 $A \to \beta$，均有$A \in N, \beta \in (N \cup T)^*$

句柄：一个句型的最左直接短语 。

活前缀：规范句型的一个前缀，不含句柄之后的任何符号。

五、问答题

1、最右推导：在推导过程中，若每次直接推导均替换句型中最右边的非终结符。

右句型：由最右推导产生的句型。

2、对输入序列ω，从S开始进行最左推导，直到得到一个合法句子或非法结构；从左到右扫描输入序列，试图用一切可能的方法，自上而下建立它的分析树。

3、直接以程序的方式模拟产生式产生语言的过程，每个产生式对应一个子程序，产生式右边的非终结符对应子程序调用，终结符则与输入序列匹配。

4、语法错误和语义错误。

5、从句子ω开始，从左到右扫描ω，反复用产生式的左部替换产生式的右部、谋求对ω的匹配，最终得到文法的开始符号，或者发现一个错误。

6、① 预测分析器：推导、匹配、回溯、接受、报错。

②移进-归约分析器：移进、归约、接受、报错。

7、①若项目 A→α.aβ 属于Ik且GO(Ik, a)=Ij，a为终结符，则置ACTION[k, a]为“将(j,a)移进栈”，简记为“sj”;

② 若项目 A→α. 属于Ik，则对任何终结符a(或结束符#)，置ACTION[k, a]为“用产生式A→α进行归约”，简记为“rj”（A→α为文法G'的第j个产生式）；

③若项目 S'→S. 属于Ik，则置ACTION[k, #]为“接受”，简记为“acc”;

④ 若GO(Ik, A)=Ij，A为非终结符，则置GOTO[k,A]=j;

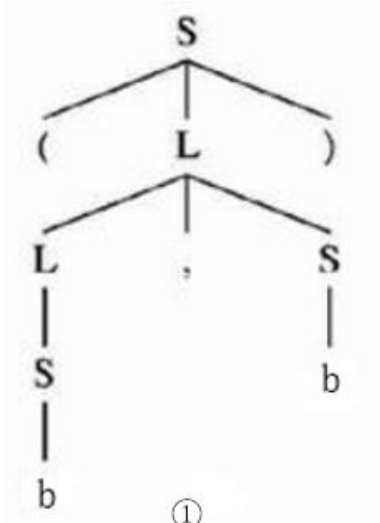
⑤ 分析表中凡不能用规则①~④填入信息的空白格均置为“报错标志”。

8、（1）开始符号：S
终结符：（ ），b
非终结符：S L

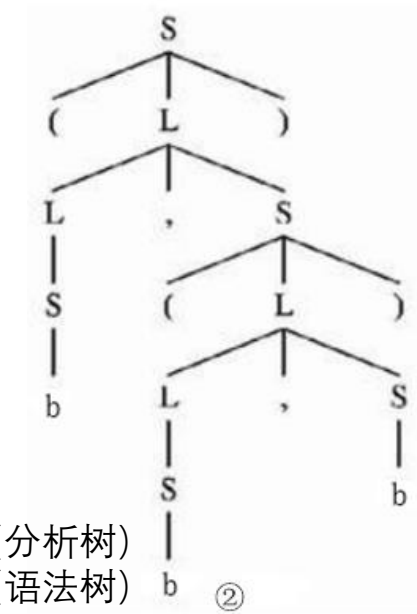
（2）(b,b)

最左推导：$S \Rightarrow (L) \Rightarrow (L,S) \Rightarrow (S,S) \Rightarrow (b,S) \Rightarrow (b,b)$

最右推导：$S \Rightarrow (L) \Rightarrow (L,S) \Rightarrow (L,b) \Rightarrow (S,b) \Rightarrow (b,b)$



①



②

这是具体语法树（分析树）
不是抽象语法树（语法树）

$(b,(b,b))$

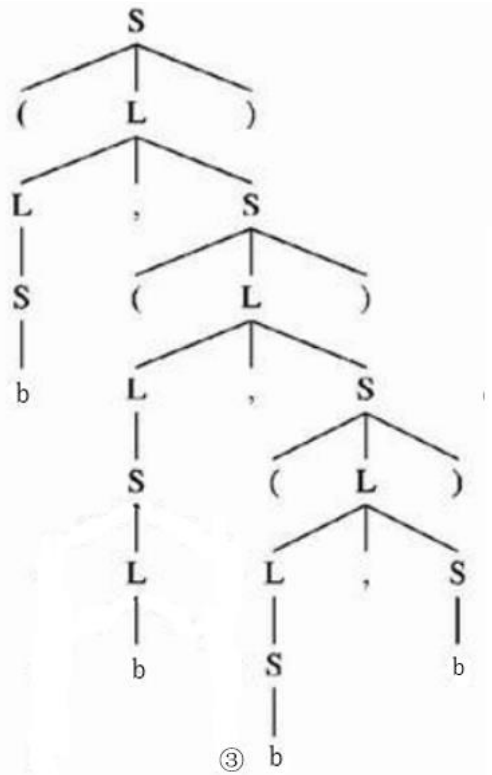
最左推导：$S \Rightarrow (L) \Rightarrow (L,S) \Rightarrow (S,S) \Rightarrow (b,S) \Rightarrow (b,(L)) \Rightarrow (b,(L,S))$
$\Rightarrow (b,(S,S)) \Rightarrow (b,(b,S)) \Rightarrow (b,(b,b))$

最右推导：$S \Rightarrow (L) \Rightarrow (L,S) \Rightarrow (L,(L)) \Rightarrow (S,(L,S)) \Rightarrow (L,(L,b))$
$\Rightarrow (L,(S,b)) \Rightarrow (L,(b,b)) \Rightarrow (S,(b,b)) \Rightarrow (b,(b,b))$

$(b,(b,(b,b)))$

最左推导：$S \Rightarrow (L) \Rightarrow (L,S) \Rightarrow (S,S) \Rightarrow (b,S) \Rightarrow (b,(L)) \Rightarrow (b,(L,S))$
$\Rightarrow (b,(S,S)) \Rightarrow (b,(b,S)) \Rightarrow (b,(b,(L))) \Rightarrow (b,(b,(L,S))) \Rightarrow (b,(b,(S,S)))$
$\Rightarrow (b,(b,(b,S))) \Rightarrow (b,(b,(b,b)))$

最右推导：$S \Rightarrow (L) \Rightarrow (L,S) \Rightarrow (L,(L)) \Rightarrow (S,(L,S)) \Rightarrow (L,(L,(L)))$
$\Rightarrow (L,(L,(L,S))) \Rightarrow (L,(L,(L,b))) \Rightarrow (L,(L,(S,b))) \Rightarrow (L,(L,(b,b)))$
$\Rightarrow (L,(S,(b,b))) \Rightarrow (L,(b,(b,b))) \Rightarrow (S,(b,(b,b))) \Rightarrow (b,(b,(b,b)))$



③

（3）n元组，其中任何一元可为b或n元组（n = 1,2,…）

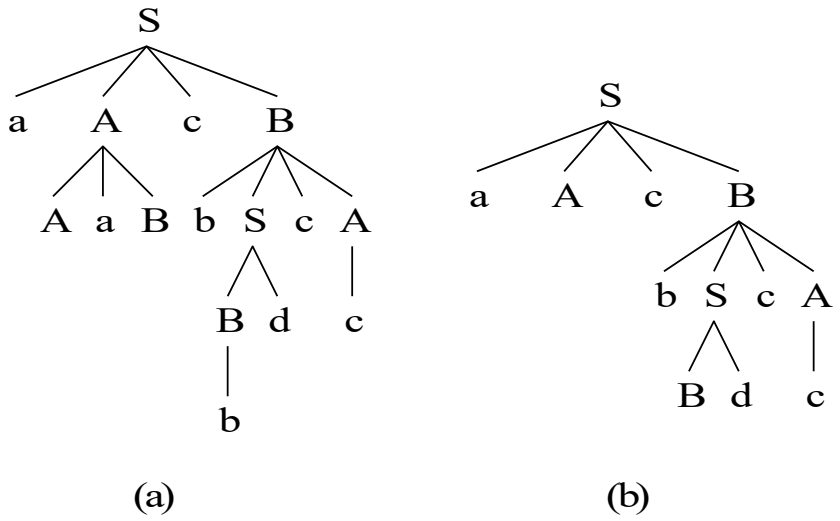
9、 (1) AaB、Bd

(2) $s \Rightarrow aAcB \Rightarrow aAaBcB \Rightarrow acaBcB \Rightarrow acabcB$
$\Rightarrow acabcbScA \Rightarrow acabcbBdcA \Rightarrow acabcbbdcA \Rightarrow acabcbbdcc$

10、 (1) E

(2) 终结符T: + - * / ( ) i
非终结符N: E T F

(3) 短语: T+T*F+i、T+T*F、T*F、T、i
直接短语: T*F、T、i
句柄: T



11、
$$E \to TE'$$
$$E' \to -TE' \mid \varepsilon$$
$$T \to FT'$$
$$T' \to /FT' \mid \varepsilon$$
$$F \to (E) \mid i$$

12、
$$V \to NV'$$
$$V' \to \varepsilon \mid [E]$$
$$E \to VE'$$
$$E' \to \varepsilon \mid +E$$
$$N \to i$$

13、 (1) 改写后的文法：S→aABe　A→bA'　A'→bcA'|ε　B→d

(2) FIRST(S)= {a}, FOLLOW(S)={#}

FIRST(A)={b}, FOLLOW(A)={d}

FIRST(A') = {b, ε }, FOLLOW(A')={d}

FIRST(B) = {d}, FOLLOW(B)={e}

(3) 预测分析表：

	a	b	c	d	e	#
S	aABe					
A		bA'				
A'		bcA'		ε		
B				d		

(4) 对 abcde 的分析：

步骤	栈内容	当前输入	动作	解释
(1)	#S	abcde#	pop(S)，push(aABe)	按 S→aABe 展开
(2)	#eBAa	abcde #	pop(a)	匹配 a
(3)	#eBA	bcde#	pop(A), push(bA')	按 A→bA'展开
(4)	#eBA'b	bcde#	pop(b)	匹配 b
(5)	# eBA'	cde#	error	c 不属于 FIRST(A')

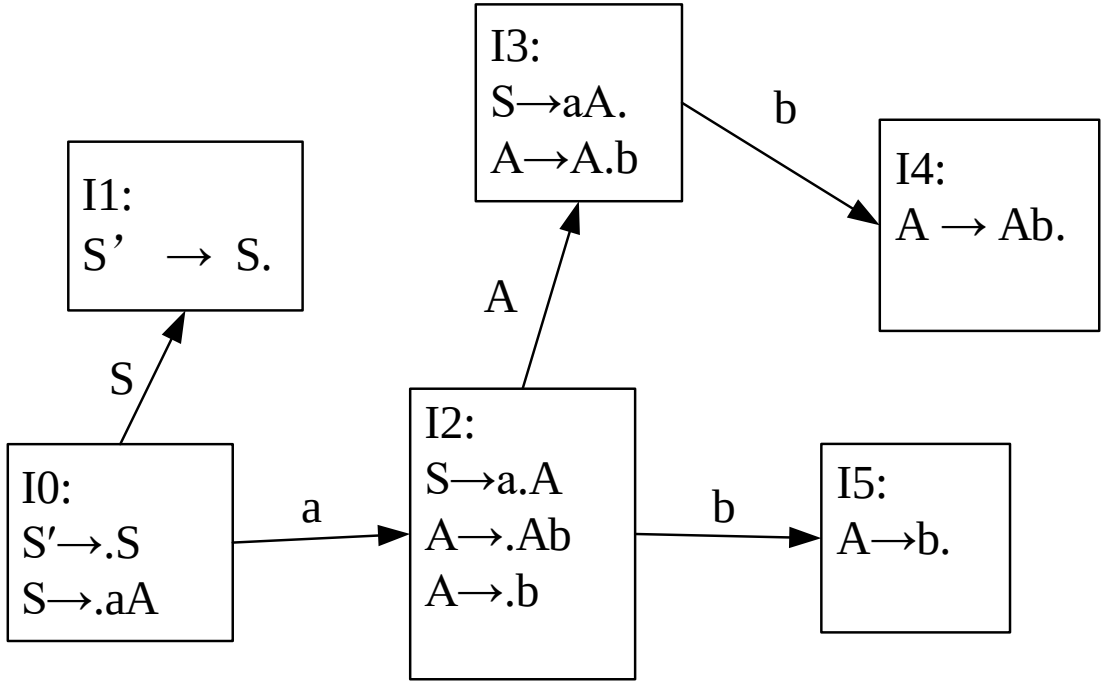
对 abcce 的分析：

步骤	栈内容	当前输入	动作	解释
(1)	#S	abcce#	pop(S)，push(aABe)	按 S→aABe 展开
(2)	#eBAa	abcce #	pop(a)	匹配 a
(3)	#eBA	bcce#	pop(A), push(bA')	按 A→bA'展开
(4)	#eBA'b	bcce#	pop(b)	匹配 b
(5)	# eBA'	cce#	error	出错，c 不属于 FIRST(A')

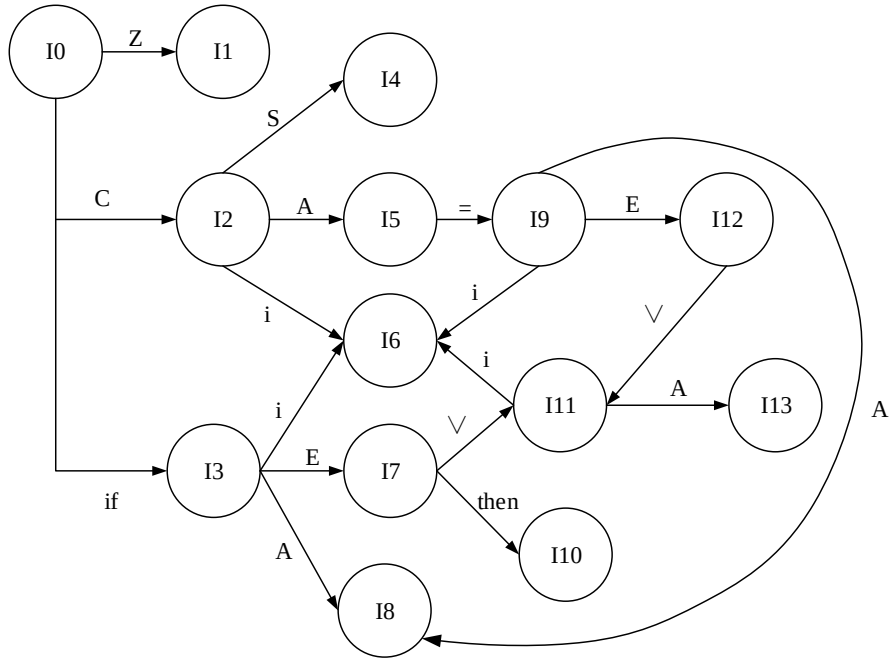
对 abbcde 的分析：

步骤	栈内容	当前输入	动作	解释
(1)	#S	abbcde#	pop(S)，push(aABe)	按 S→aABe 展开
(2)	#eBAa	abbcde #	pop(a)	匹配 a
(3)	#eBA	bbcde#	pop(A), push(bA')	按 A→bA'展开
(4)	#eBA'b	bbcde#	pop(b)	匹配 b
(5)	#eBA'	bcde#	pop(A'), push(bcA')	按 A'→bcA'展开
(6)	#eBA'cb	bcde#	pop(b)	匹配 b
(7)	#eBA'c	cde#	pop(c)	匹配 c
(8)	#eBA'	de#	pop(A')	按 A'→ ε 展开
(9)	#eB	de#	pop(B), push(d)	按 B→d 展开
(10)	#ed	de#	pop(d)	匹配 d
(11)	#e	e#	pop(e)	匹配 e
(12)	#	#	accept	正确结束

14、



15、



状态	ACTION						GOTO				
	if	then	=	V	i	#	Z	C	S	E	A
0	s3						1	2			
1						acc					
2					s6				4		5
3					s6					7	8
4						r1					
5			s9								
6		r6	r6	r6		r6					
7		s10		s11							
8		r5		r5		r5					
9					s6					12	8
10					r2						
11					s6						13
12				s11		r3					
13		r4		r4		r4					

第四章

一、选择题

1、C
2、A
3、B
4、D
5、C
6、B

二、填空题

1、翻译属性文法
2、语法制导定义、翻译方案
3、自下而上、自上而下
4、综合
5、语法、语义
6、中间、目标、执行解释
7、子节点、父节点和/或兄节点

三、名词解释

三元式：由三个域op、arg1和arg2记录的三地址码的表示方式。
语义规则：规定产生式所代表的语言结构之间关系的规则。
翻译方案：用具体属性和运算表示的语义规则。

四、简答题

1、将语言结构的语义以属性的形式赋予代表此结构的文法符号，而属性的计算以语义规则的形式赋予由文法符号组成的产生式，在语法分析推导或者规约的每一步骤中，通过语义规则实现对属性的计算。

2、后缀式、三地址码（三元式、四元式）、树形表示（树、DAG）

3、 （1）a – b ≤ a 0 > ^ b 0 < v

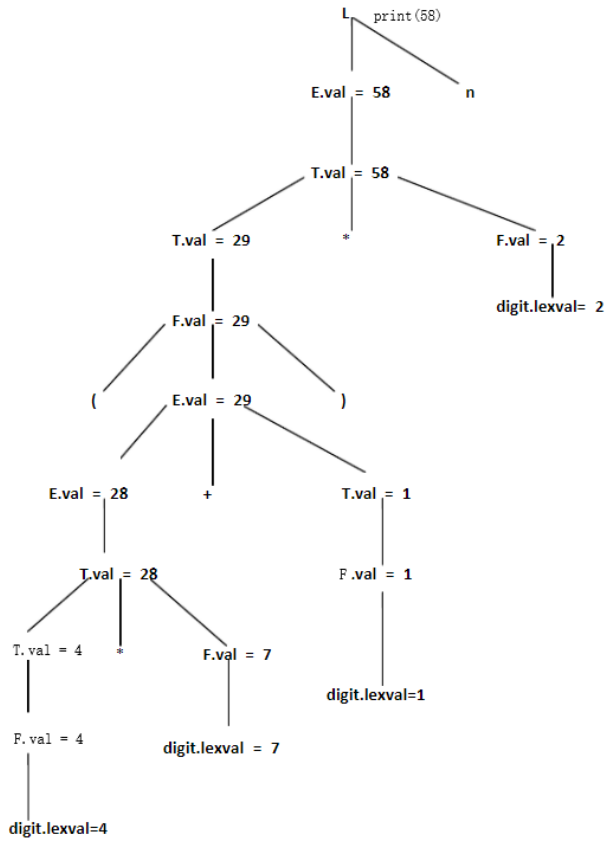
（2）a a b * d – a b d * - * d / -

（3）a – b + 0 ≤ a 0 < a b – 2 > ^ v

（4）a b c * a - * b c + ≤ d ^

五、解答题

1、



2、424513034125