1. (1) ab-c+x

(2) A not CD not or not or

(3) abcde/+*+

(4) AB and C not D or or

(5) a-bc-d+*+

(6) ABor CDnotE and or and

(7) if xy+z*0= then ab+c↑ else abc↑↑

3. (1) 三元式:

	op	arg1	arg2
(0)	+	a	b
(1)	uminus	(0)	
(2)	+	c	d
(3)	*	(1)	(2)
(4)	+	a	b
(5)	+	(4)	c
(6)	-	(3)	(5)

(2) 间接三元式

	op	arg1	arg2
(0)	+	a	b
(1)	uminus	(0)	
(2)	+	c	d
(3)	*	(1)	(2)
(4)	+	(0)	c
(5)	-	(3)	(4)

(3) 四元式序列

	op	arg1	arg2	result
(0)	+	a	b	T_1
(1)	uminus	T_1		T_2
(2)	+	c	d	T_3
(3)	*	T_2	T_3	T_4
(4)	+	a	b	T_5
(5)	+	T_5	c	T_6
(6)	-	T_4	T_6	T_7

4

步骤	输入串	state	val	三地址代码
(1)	A:=B*(-C+D)	-	-	
(2)	:=B*(-C+D)	A	A	
(3)	:=B*(-C+D)	id	A	
(4)	B*(-C+D)	id:=	A-	
(5)	*(-C+D)	id:=id	A-B	
(6)	*(-C+D)	id:=E	A-B	
(7)	(-C+D)	id:=E*	A-B-	
(8)	-C+D)	id:=E*(	A-B--	
(9)	C+D)	id:=E*(-	A-B---	
(10)	+D)	id:=E*(-C	A-B---C	
(11)	+D)	id:=E*(-id	A-B---C	
(12)	+D)	id:=E*(-E	A-B---C	T_1 := -C
(13)	+D)	id:=E*(E	A-B--T_1	
(14)	D)	id:=E*(E+	A-B--T_1-	
(15)	)	id:=E*(E+D	A-B--T_1-D	
(16)	)	id:=E*(E+id	A-B--T_1-D	
(17)	)	id:=E*(E+E	A-B--T_1-D	T_2 := T_1+D
(18)	)	id:=E*(E	A-B--T_2	
(19)		id:=E*(E)	A-B--T_2-	
(20)		id:=E*E	A-B-T_2	T_3 := B*T_2
(21)		id:=E	A-T_3	T_4 := A:=T_3
(22)				
(23)				
(24)				

6

1 (jnz, A, -, 0)
2 (j, -, -, 3)
3 (jnz, B, -, 5)
4 (j, -, -, 0)
5 (jnz, C, -, 0)
6 (j, -, -, 7)
7 (jnz, D, -, 0)
8 (j, -, -, 0)

7.

1 (j<, A, C, 3)
2 (j, -, -, 0)
3 (j<, B, D, 5)
4 (j, -, -, 0)
5 (j=, A, '1', 7)
6 (j, -, -, 10)
7 (+, C, '1', T_1)
8 (:=, T_1, -, C)
9 (j, -, -, 1)
10 (j≤, A, D, 12)
11 (j, -, -, 1)
12 (+, A, '2', T_2)
13 (:=, T_2, -, A)
14 (j, -, -, 10)
15 (j, -, -, 1)

11.

(1) 属性文法

S → for(E_1; E_2; E_3)S_1
E_2.lable = newlable (∵ E_2也可以执行赋值等语句)
E_3.lable = newlable
E_2.true = newlable
E_2.false = S.next
S_1.next = E_3.lable
S.code = E_1.code || gen(E_2.lable ':') ||
E_2.code || gen(E_2.true ':') ||
S_1.code || gen(E_3.lable) ||
E_3.code || gen('goto' E_2.lable)

(2) 翻译模式

S → for(E_1; M_1 E_2; M_2 E_3) M_3 S_1 { backpatch(S_1.nextlist, M_2.quad)
backpatch(E_2.truelist, M_3.quad)
backpatch(E_3.nextlist, M_1.quad)
S.nextlist := E_2.falselist
emit('j, -, -,' M_1.quad }

M → ε { M.quad := nextquad }