6. 按 7.4.2 节的办法，写出布尔式 A or (B and not (C or D))的四元式序列。

归约步骤：1. $E_1 \to A$
2. $M_1 \to \varepsilon$
3. $E_2 \to B$
4. $M_2 \to \varepsilon$
5. $E_3 \to C$
6. $M_3 \to \varepsilon$
7. $E_4 \to D$
8. $E_5 \to E_3$ or $M_3 E_4$
9. $E_6 \to (E_5)$
10. $E_7 \to$ not E_6
11. $E_8 \to E_2$ and $M_2 E_7$
12. $E_9 \to (E_8)$
13. $E_{10} \to E_1$ or $M_1 E_9$

1. E_1.truelist := makelist(100)
   E_1.falselist := makelist(101)
   emit('jnz' ',' A ',' '_' ',' '0')
   emit('j,-,-,0')
   此时 E_1.truelist 为 (100)
   E_1.falselist 为 (101)
   nextquad = 102
   四元式序列为 100: jnz, A, -, 0
   101: j, -, -, 0

2. M1.quad := 102

3. E_2.truelist := makelist(102)
   E_2.falselist := makelist(103)
   emit('jnz' ',' B ',' '_' ',' '0')
   emit('j,-,-,0')
   此时 E_2.truelist 为 (102)
   E_2.falselist 为 (103)
   nextquad = 104
   四元式序列为 100: jnz, A, -, 0
   101: j, -, -, 0
   102: jnz, B, -, 0
   103: j, -, -, 0

4. M_2.quad := 104

5. E_3.truelist := makelist(104)
   E_3.falselist := makelist(105)
   emit('jnz' ',' C ',' '_' ',' '0')
   emit('j,-,-,0')
   此时 E_3.truelist 为 (104)
   E_3.falselist 为 (105)
   nextquad = 106
   四元式序列为 100: jnz, A, -, 0
   101: j, -, -, 0
   102: jnz, B, -, 0
   103: j, -, -, 0

续: 104: jnz , C , - , 0
105: j , - , - , 0

6. M_3.quad := 106

7. E_4.truelist := makelist(106)
E_4.falselist := makelist(107)
emit('jnz', 'D', '-', '0')
emit('j, -, -, 0')
此时 E_4.truelist 为 106
E_4.falselist 为 107
nextquad = 108
回填式序列为
100: jnz , A , - , 0
101: j , - , - , 0
102: jnz , B , - , 0
103: j , - , - , 0
104: jnz , C , - , 0
105: j , - , - , 0
106: jnz , D , - , 0
107: j , - , - , 0

8. backpatch(105, 106)
E_5.truelist := merge(104, 106)
E_5.falselist := 107
此时 E_5.truelist 为 104—106
E_5.falselist 为 107
nextquad = 108
回填式序列为
100: jnz , A , - , 0
101: j , - , - , 0
102: jnz , B , - , 0
103: j , - , - , 0
104: jnz , C , - , 0
105: j , - , - , 106
106: jnz , D , - , 0
107: j , - , - , 0

9. E_6.truelist := 104—106
E_6.falselist := 107

10. E_7.truelist := 107
E_7.falselist := 104—106

11. backpatch(102, 104)
E_8.truelist := 107
E_8.falselist := merge(103, 104—106)
此时 E_8.truelist 为 107
E_8.falselist 为 103—104—106
nextquad = 108
回填式序列为
100: jnz , A , - , 0
101: j , - , - , 0
102: jnz , B , - , 104
103: j , - , - , 0
104: jnz , C , - , 0
105: j , - , - , 106
106: jnz , D , - , 0
107: j , - , - , 0

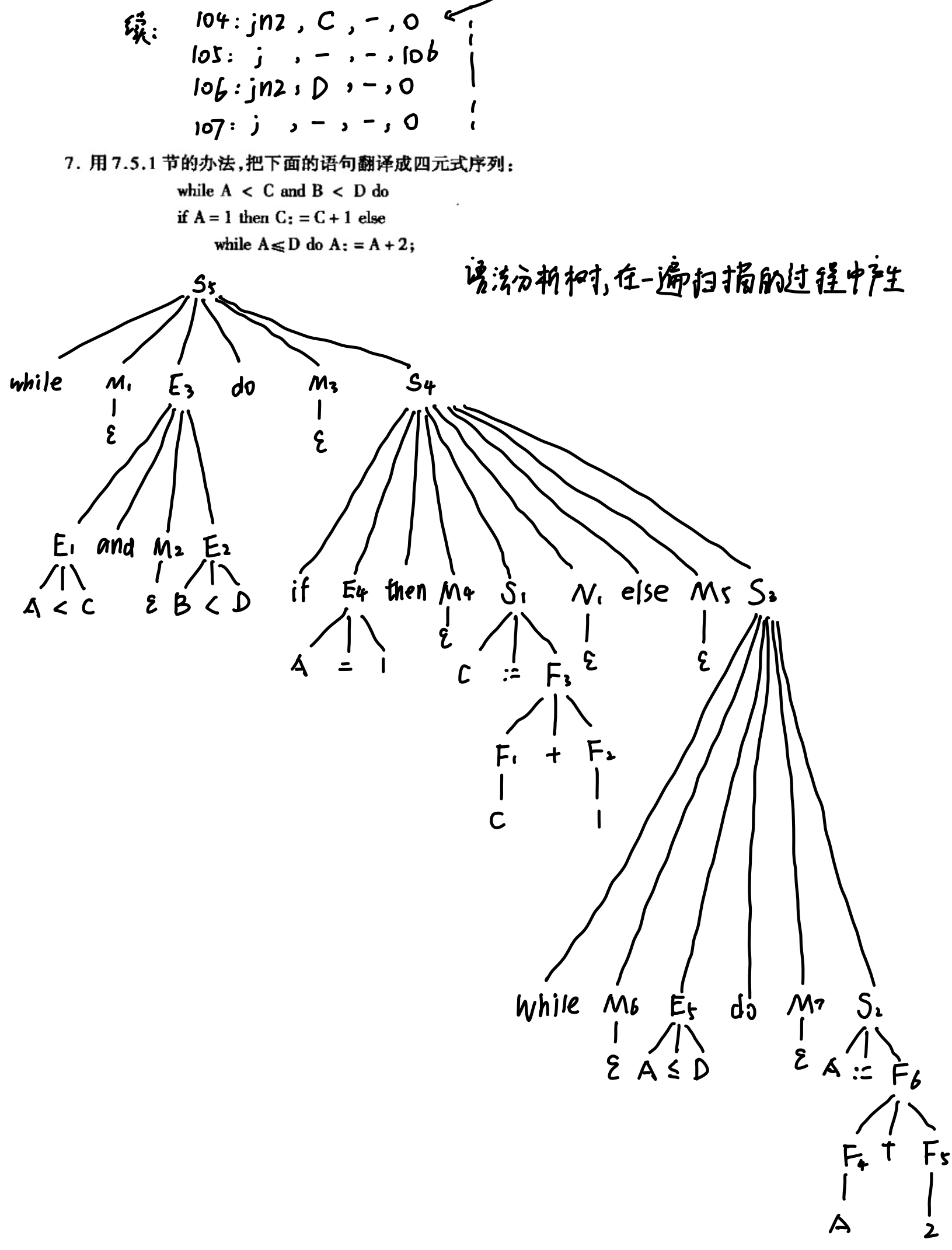
12. E_9.truelist := 107
E_9.falselist := 103—104—106

13. backpatch(101, 102)
E_{10}.truelist := merge(103, 107)
E_{11}.falselist := 103—104—106
此时 E_{10}.truelist 为 103—107
E_{10}.falselist 为 103—104—106
nextquad = 108
回填式序列为(最终)
100: jnz , A , - , 0
101: j , - , - , 102
102: jnz , B , - , 104
103: j , - , - , 0

续：
104：jn2，C，-，0
105：j，-，-，106
106：jn2，D，-，0
107：j，-，-，0

7. 用 7.5.1 节的办法，把下面的语句翻译成四元式序列：

```
while A < C and B < D do
if A = 1 then C: = C + 1 else
    while A≤D do A: = A + 2;
```



归约过程：1. $M_1 \to \varepsilon$

2. $E_1 \to A < C$

3. $M_2 \to \varepsilon$

4. $E_2 \to B < D$

5. $E_3 \to E_1$ and M_2 E_2

6. $M_3 \to \varepsilon$

7. $E_4 \to A = 1$

8. $M_4 \to \varepsilon$

9. $F_1 \to C$

10. $F_2 \to 1$

11. $F_3 \to F_1 + F_2$

12. $S_1 \to C := F_3$

13. $N_1 \to \varepsilon$

14. $M_5 \to \varepsilon$

15. $M_6 \to \varepsilon$

16. $E_5 \to A \leq D$

17. $M_7 \to \varepsilon$

18. $F_4 \to A$

19. $F_5 \to 2$

20. $F_6 \to F_4 + F_5$

21. $S_2 \to A := F_6$

22. $S_3 \to$ while M_8 E_5 do M_7 S_2

23. $S_4 \to$ if E_4 then M_4 S_1 N_1 else M_5 S_2

24. $S_5 \to$ while M_1 E_3 do M_3 S_4

1. M_1.quad := 100

2. E_1.truelist := makelist(100)

   E_1.falselist := makelist(101)

   emit后：100: j<, A, C, 0

   101: j, -, -, 0

   此时nextquad = 102

3. M_2.quad := 102

4. E_2.truelist := makelist(102)

   E_2.falselist := makelist(103)

   emit后：100: j<, A, C, 0

   101: j, -, -, 0

   102: j<, B, D, 0

   103: j, -, -, 0

   此时nextquad = 104

5. backpatch(100, 102)

   E_3.truelist := 102

   E_3.falselist := merge(101, 103)

   backpatch后：100: j<, A, C, 102

   101: j, -, -, 0

   102: j<, B, D, 0

   103: j, -, -, 0

6. M_3.quad := 104

7. E_4.truelist := makelist(104)

   E_4.falselist := makelist(105)

   emit后：100: j<, A, C, 102

   101: j, -, -, 0

   102: j<, B, D, 0

   103: j, -, -, 0

   104: j=, A, 1, 0

   105: j, -, -, 0

   此时nextquad = 106

8. M_4.quad := 106

9. F_1.place := C

10. F_2.place := 1

11. F_3.place := T_1

    emit后：100: j<, A, C, 102

    101: j, -, -, 0

    102: j<, B, D, 0

    103: j, -, -, 0

    104: j=, A, 1, 0

    105: j, -, -, 0

    106: +, C, 1, T_1

    此时nextquad = 107

12. emit后：100: j<, A, C, 102

    101: j, -, -, 0

    102: j<, B, D, 0

    103: j, -, -, 0

    104: j=, A, 1, 0

    105: j, -, -, 0

    106: +, C, 1, T_1

    107: :=, T_1, -, C

    此时nextquad = 108

13. N_1.nextlist := makelist(108)

    emit后：100: j<, A, C, 102

    101: j, -, -, 0

    102: j<, B, D, 0

    103: j, -, -, 0

    104: j=, A, 1, 0

    105: j, -, -, 0

    106: +, C, 1, T_1

    107: :=, T_1, -, C

    108: j, -, -, -

    此时nextquad = 109

14. M_5.quad := 109

15. M_6.quad := 109

16. E_5.truelist := makelist(109)

E_5.falselist := makelist(110)

emit后：
100: j< , A , C , 102
101: j , - , - , 0
102: j< , B , D , 0
103: j , - , - , 0
104: j= , A , 1 , 0
105: j , - , - , 0
106: + , C , 1 , T_1
107: := , T_1 , - , C
108: j , - , - , -
109: j<= , A , D , 0
110: j , - , - , 0

此时nextquad = 111

17. M_7.quad := 111

18. F_4.place := A

19. F_5.place := 2

20. F_6.place := T_2

emit后：
100: j< , A , C , 102
101: j , - , - , 0
102: j< , B , D , 0
103: j , - , - , 0
104: j= , A , 1 , 0
105: j , - , - , 0
106: + , C , 1 , T_1
107: := , T_1 , - , C
108: j , - , - , -
109: j<= , A , D , 0
110: j , - , - , 0
111: + , A , 2 , T_2

此时nextquad = 112

21. emit后：
100: j< , A , C , 102
101: j , - , - , 0
102: j< , B , D , 0
103: j , - , - , 0
104: j= , A , 1 , 0
105: j , - , - , 0
106: + , C , 1 , T_1
107: := , T_1 , - , C
108: j , - , - , -
109: j<= , A , D , 0
110: j , - , - , 0
111: + , A , 2 , T_2
112: := , T_2 , - , A

此时nextquad = 113

22. backpatch(空, 109)

backpatch(109, 111)

S_3.nextlist := 110

backpatch 和emit后：
100: j< , A , C , 102
101: j , - , - , 0
102: j< , B , D , 0
103: j , - , - , 0
104: j= , A , 1 , 0
105: j , - , - , 0
106: + , C , 1 , T_1
107: := , T_1 , - , C
108: j , - , - , -
109: j<= , A , D , 111
110: j , - , - , 0
111: + , A , 2 , T_2
112: := , T_2 , - , A
113: j , - , - , 109

此时nextquad = 114

23. backpatch(104, 106)
backpatch(105, 109)
S_4.nextlist := merge(空, 108, 空)
backpatch后:

100: j< , A , C , 102
101: j , - , - , 0
102: j< , B , D , 0
103: j , - , - , 0
104: j= , A , 1 , 106
105: j , - , - , 109
106: + , C , 1 , T_1
107: := , T_1 , - , C
108: j , - , - , -
109: jK= , A , D , 111
110: j , - , - , 0
111: + , A , 2 , T_2
112: := , T_2 , - , A
113: j , - , - , 109

24. backpatch(108, 100)
backpatch(102, 104)
S_5.nextlist := 101 - 103
backpatch 和emit后: (最终结果)

100: j< , A , C , 102
101: j , - , - , 0
102: j< , B , D , 104
103: j , - , - , 0
104: j= , A , 1 , 106
105: j , - , - , 109
106: + , C , 1 , T_1
107: := , T_1 , - , C
108: j , - , - , 100
109: jK= , A , D , 111
110: j , - , - , 0
111: + , A , 2 , T_2
112: := , T_2 , - , A
113: j , - , - , 109
114: j , - , - , 100

此时 nextquad = 115