

سوال ۱)

(آ)

در حالت نوعدهی ایستا با خطای کامپایل مواجه خواهیم شد زیرا متغیر v دو تایپ مختلف گرفته است. در نوعدهی پویا برحسب مقدار n در زمان اجرا v نوع `string` یا `int` می گیرد و خطای اجرا رخ نمی دهد.

(ب)

در حالت نوعدهی ایستا دوباره با خطای کامپایل مواجه خواهیم شد زیرا در یک `if` متغیر v که از نوع `int` می باشد با یک استرینگ جمع شده است و خطای نوع رخ می دهد. در نوعدهی پویا اگر `n == "type"` برقرار باشد برنامه بدون خطا اجرا می شود اما در صورتی که در زمان اجرا برنامه وارد `else` شود خطای اجرا صورت می گیرد.

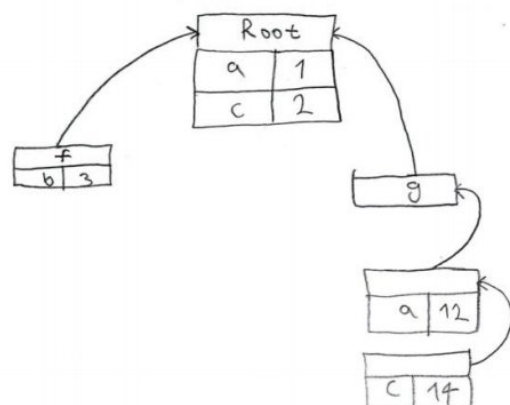
(ج)

به صورت کلی نوعدهی ایستا موجب می شود که خطاها در زمان کامپایل شناسایی شوند تا در زمان اجرا با خطا مواجه نشویم. اما به دلیل محافظه کار بودن آنها، نوعدهی ایستا از کامپایل شدن برخی برنامه ها که در واقع درست هستند و مشکلی ایجاد نمی کند جلوگیری می کند. در سوی مقابل نوعدهی پویا انعطاف پذیری بیشتری برای برنامه نویس فراهم می کند اما شناسایی خطاها را به زمان اجرا موکول می کند.

در بخش آ نوعدهی پویا عملکرد بهتری دارد زیرا نوعدهی ایستا در حالی از کامپایل شدن کد جلوگیری می کند که کد بدون اشکال است. در طرف مقابل در بخش ب نوعدهی ایستا مناسبتر است زیرا برنامه در نوعدهی پویا به ازای اکثر ورودی ها با خطای اجرا مواجه خواهد شد.

سوال ۲)

(آ)



(ب)

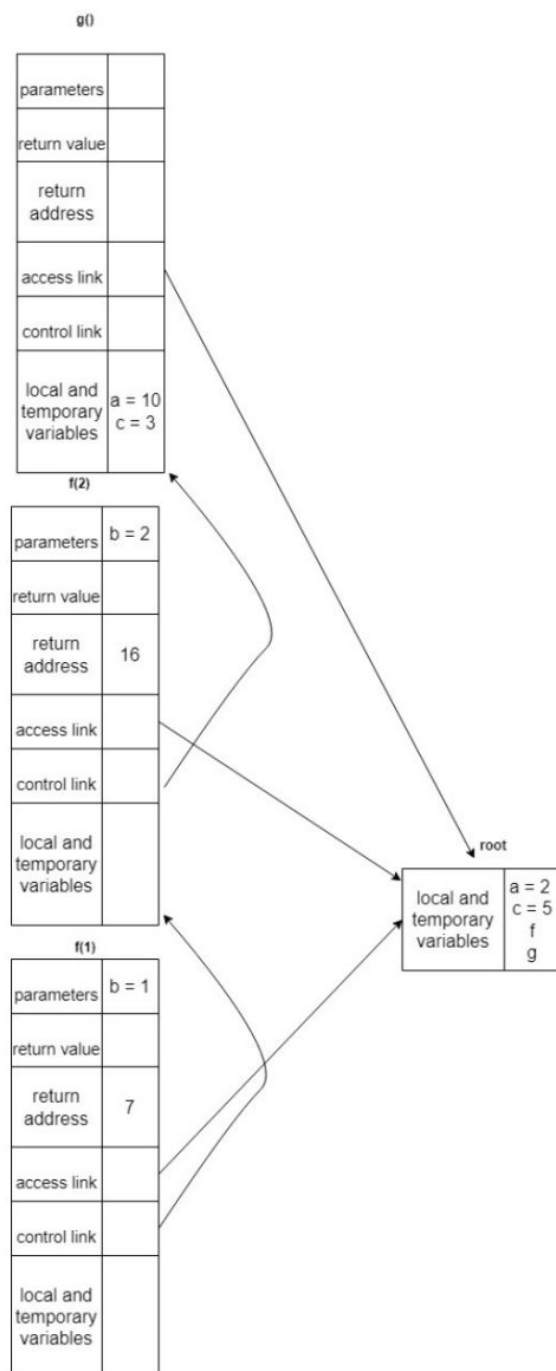
۲۶۰

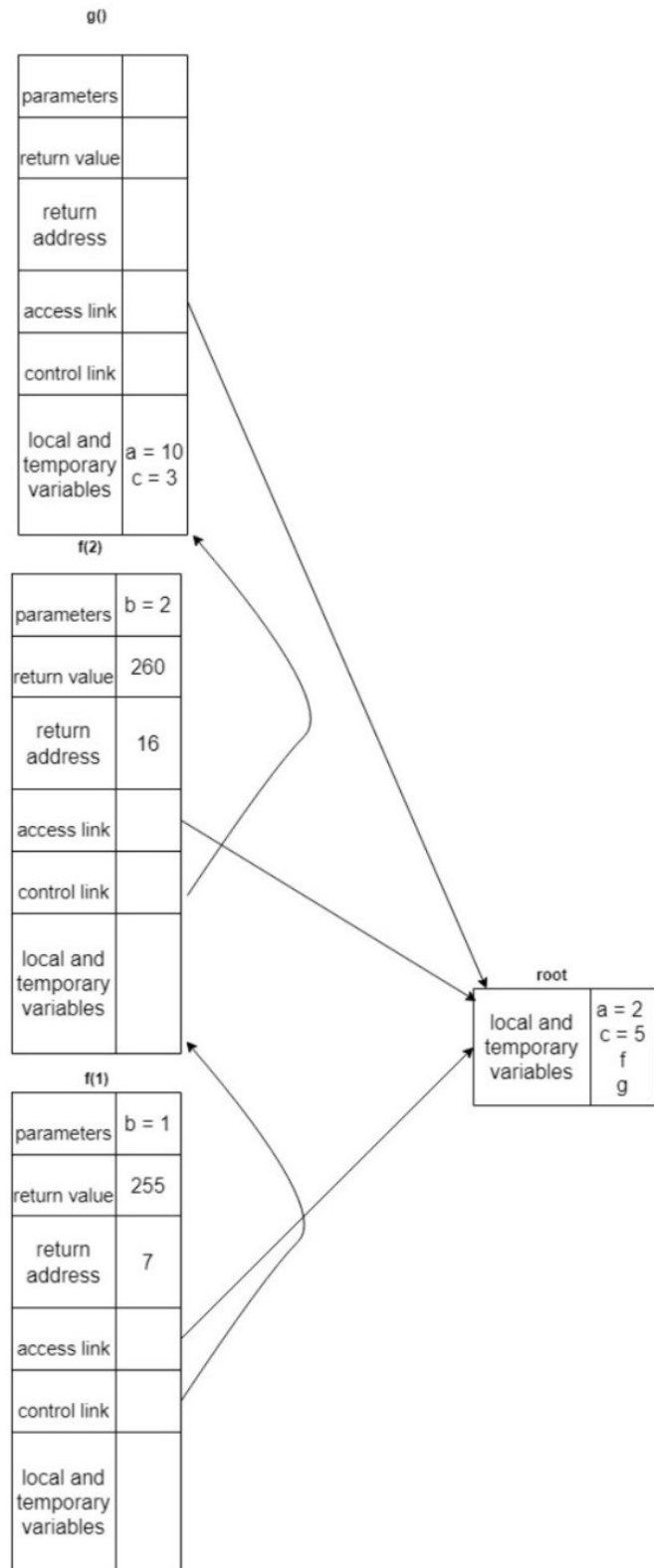
(ج)

۱۰۳۶

(د)

در محدوده ایستا خطایابی و آنالیز برنامه بسیار راحت تر از محدوده پویا است.





سوال ۳
(۱)

$t_1 = a + b$
 $t_2 = -t_1$
 $t_3 = c \times d$
 $t_4 = t_2 + t_3$
 $t_5 = a + b$
 $t_6 = t_4 + t_5$

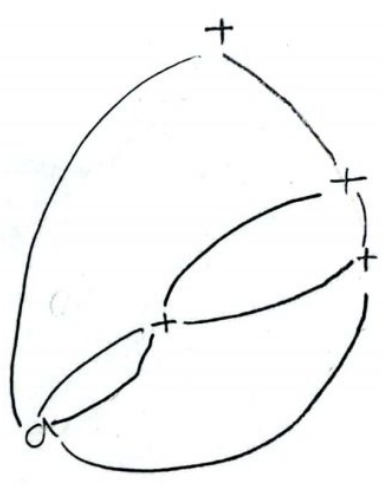
op	arg1	arg2	result
+	a	b	t1
minus	t1		t2
x	c	d	t3
+	t2	t3	t4
+	a	b	t5
+	t4	t5	t6

n	(0)
n+1	(1)
n+2	(2)
n+3	(3)
n+4	(4)
n+5	(5)

	op	arg1	arg2
0	+	a	b
1	minus	(0)	
2	x	c	d
3	+	(1)	(2)
4	+	a	b
5	+	(3)	(4)

a b + minus c d x + a b + +

(۱)



سوال ۴

(الف)

S -> repeat S1 until B

```
{  
    check = newlabel()  
    B.true = newlabel()  
    B.false = S.next  
    S1.next = check  
    S.code = label(B.true) || S1.code || check || B.code  
}
```

(ب)

S -> repeat M1 S1 until M2 B

```
{  
    backpatch(B.truelist, M1.instr);  
    backpatch(S1.nextlist, M2.instr);  
    S.nextlist = B.falselist  
}
```

(ج)

$M1.instr = \{100\}$

$A \rightarrow K: K+1$

$S \rightarrow A$ $S.nextlist = null$

$M2.instr = \{101\}$

100: $K := K+1$

101: if $K < 100$ goto —

102: goto —

103: if $y > 300$ goto —

104: goto —

back patch ($B1.truelist, M3.instr$)

101: if $K < 100$ goto 103

105: if $a == 20$ goto —

106: goto —

back patch ($B3.falselist, M4.instr$)

102: goto 105

104: goto 105

$B1 \rightarrow E1 < E2$ $E1 \rightarrow K$
 $E2 \rightarrow 100$

$B1.truelist = \{101\}$

$B1.falselist = \{102\}$

$M5 \rightarrow \{ M3.nextinstr = \{103\} \}$

$E3 \rightarrow y$

$B2 \rightarrow E3 > E4$

$E4 \rightarrow 300$

$B2.truelist = \{103\}$

$B3.falselist = \{104\}$

$B3 \rightarrow B1$ and $B2$

$\hookrightarrow B3.truelist = B2.truelist = \{103\}$

$B3.falselist = \{102, 104\}$

$M4 \rightarrow \{ M4.nextinstr = \{105\} \}$

$E5 \rightarrow a$

$E6 \rightarrow 20$

$B4 \rightarrow E5 == E6$

$B4.truelist = \{105\}$ $B4.falselist = \{106\}$

$B5 \rightarrow ! B4$

$B5.truelist = \{106\}$ $B5.falselist = \{105\}$

$B6 \rightarrow B3$ or $B5$

$B6.truelist = \{103, 106\}$

$B6.falselist = \{105\}$

100: $x := x + 1$

101: if $x < 100$ goto 103

102: goto 105

103: if $y > 300$ goto —

104: goto 105

105: if $a \leq 20$ goto —

106: goto —

backPatch($B_6.truelist, M_1.instr$)

103: goto 100

106: goto 100

backPatch($S.nextlist, M_2.instr$)

$S.nextlist = null$

$S_2 \rightarrow$

$S_2 \rightarrow$ repeat M_1 S until M_2 B_6

$S_2.nextlist = \{105\}$

$$S \rightarrow TS \mid \epsilon$$

$$T \rightarrow aTb \mid bTc \mid cTa \mid \epsilon$$

Note: Left & Right attributes indicate whether the string derived from the corresponding variable starts or ends with 'a' respectively.

$$S \rightarrow \epsilon \left\{ S.val = 0; S.right = S.left = false \right\}$$

$$S \rightarrow TS_1 \left\{ \begin{array}{l} S.val = T.val + S_1.val; \\ \text{if } (T.right \ \& \ S_1.left) \\ \quad S.val = S.val - 1; \\ S.left = T.left; \\ S.right = S_1.right; \end{array} \right\}$$

$$T \rightarrow aT_1b \left\{ \begin{array}{l} T.left = true; \\ T.right = false; \\ \text{if } (T_1.left) \ T.val = T_1.val; \\ \text{else } T.val = T_1.val + 1; \end{array} \right\}$$

$$T \rightarrow cT_1a \left\{ \begin{array}{l} T_1.right = true; \\ T_1.left = false; \\ \text{if } (T_1.right) \ T.val = T_1.val; \\ \text{else } T.val = T_1.val + 1; \end{array} \right\}$$

$$T \rightarrow bT_1c \left\{ T.val = T_1.val; \right\}$$

$T.\text{left} = T.\text{right} = \text{false}; \}$

$T \rightarrow \{ T.\text{left} = T.\text{right} = \text{false};$
 $T.\text{val} = 0; \}$