

طراحی کامپایلرها

نیم سال اول ۰۳-۰۲



استاد: سمانه حسینمردی

دانشکده‌ی مهندسی کامپیوتر

مهلت ارسال: ۱۱ خرداد

تمرین سوم

مسئله‌ی ۱.

با توجه به قطعه کد زیر وضعیت Symbol Table و Scope Stack هنگام اجرای خطوط ۷ و ۱۱ را مشخص کنید:

```

1. program A()
2.   var i,j : integer
3.   procedure B(a : real)
4.     var b[1..10] : real
5.     procedure C(k : integer)
6.       var c : real
7.       b(4) := a + c
8.     end C
9.   end B
10.  procedure D(l : integer)
11.    l := i + j
12.    procedure E()
13.      var d : real
14.    end E
15.  end D
16. end A
    
```

| | Lex | Type | Datatype | Scope | | | Scope Stack |
|---|-----|-------|----------|-------|--|--|-------------|
| 0 | A | proc | | 1 | | | 0 |
| 1 | i | var | int | 1 | | | 4 |
| 2 | j | var | int | 1 | | | 7 |
| 3 | B | proc | | 1 | | | |
| 4 | a | param | real | 2 | | | |
| 5 | b | array | real | 2 | | | |
| 6 | C | proc | | 2 | | | |
| 7 | k | param | int | 3 | | | |
| 8 | c | var | real | 3 | | | |
| | | | | | | | |
| | | | | | | | |
| | Lex | Type | Datatype | Scope | | | Scope Stack |
| 0 | A | proc | | 1 | | | 0 |
| 1 | i | var | int | 1 | | | 5 |
| 2 | j | var | int | 1 | | | |
| 3 | B | proc | | 1 | | | |
| 4 | D | proc | | 1 | | | |
| 5 | l | param | int | 2 | | | |

مسئله‌ی ۲.

خطاهای static و dynamic قطعه کد زیر را بیابید و نوع آن‌ها را مشخص کنید. (راهنمایی: این کد در کل شامل ۴ خطا می‌باشد)

```
1. program A()
2.   var i : real
3.   procedure B(j : integer)
4.     var k : integer
5.     i := 0
6.     j := 6
7.     k := i + j
8.   end B
9.   procedure C(l : integer)
10.    var l : integer
11.    procedure D(k : real)
12.      var a[1..5] : real
13.      a[j] := 5
14.      k := 3/i
15.    end D
16.  end C
17. end A
```

- خط ۷: خطای type، یک int و یک real جمع شده و در یک int ریخته شده. (static)
- خط ۱۰: خطای unique بودن، متغیر با نام l در این scope تعریف شده. (static)
- خط ۱۳: خطای scope، متغیر j در این scope تعریف شده نیست. (static)
- خط ۱۴: خطای division by zero، مقدار i در رانتایم صفر خواهد بود. (dynamic)

مسئله‌ی ۳.

برای گرامر زیر:

۱. جدول (۱) LR را رسم کنید.

۲. جدول (۱) LALR را رسم کنید.

۳. آیا این گرامر از نوع (۱) LR و (۱) LALR است؟ برای هر یک دلیل بیاورید و یا یک رشته مثال بزنید که نتوان آن را پارس کرد.

۱. $S \rightarrow G x$

۲. $S \rightarrow y G z$

۳. $S \rightarrow H z$

۴. $S \rightarrow y H x$

۵. $G \rightarrow w$

۶. $H \rightarrow w$

مسئله ۳.

برای گرامر زیر:

۱. جدول LR (۱) را رسم کنید.

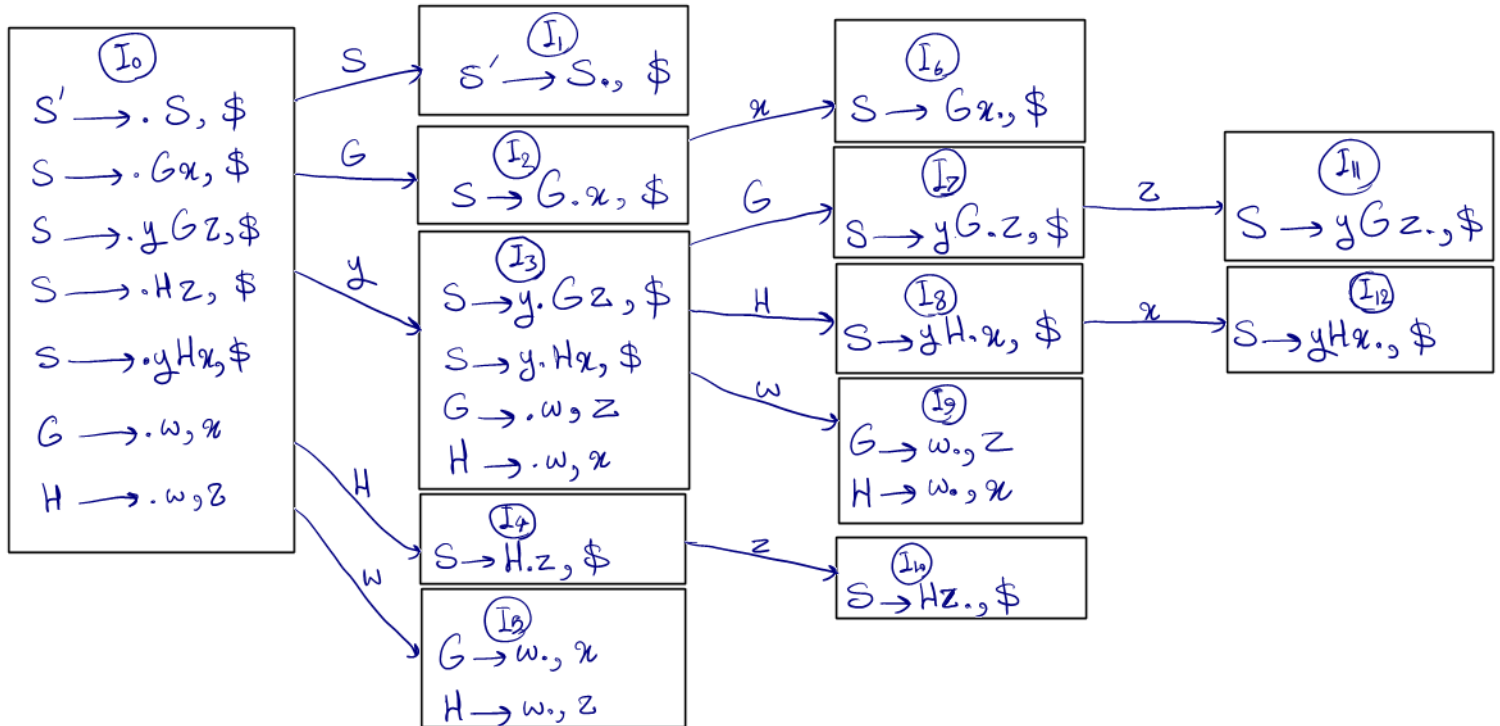
۲. جدول LALR (۱) را رسم کنید.

۳. آیا این گرامر از نوع LR (۱) و LALR (۱) است؟ برای هر یک دلیل بیاورید و یا یک رشته مثال بزنید که نتوان آن را پارس کرد.

۱. $S \rightarrow G x$
۲. $S \rightarrow y G z$
۳. $S \rightarrow H z$
۴. $S \rightarrow y H x$
۵. $G \rightarrow w$
۶. $H \rightarrow w$

$\alpha \cdot X \beta$

۱. با استفاده از قانون $S' \rightarrow S$ و دستگونیته lookahead states و جدول LR (۱) را رسم کنید:



بنابر این جدول LR (۱) به صورت زیر خواهد بود: صفحه بعد

جدول یاری LRU)

action

goto

| state | w | x | y | z | \$ | | S | G | H |
|-------|----|-----|----|-----|-----|--|---|---|---|
| 0 | S5 | | S3 | | | | 1 | 2 | 4 |
| 1 | | | | | acc | | | | |
| 2 | | S6 | | | | | | | |
| 3 | S9 | | | | | | | 7 | 8 |
| 4 | | | | S10 | | | | | |
| 5 | | R5 | | R6 | | | | | |
| 6 | | | | | R1 | | | | |
| 7 | | | | S11 | | | | | |
| 8 | | S12 | | | | | | | |
| 9 | | R6 | | R5 | | | | | |
| 10 | | | | | R3 | | | | |
| 11 | | | | | R2 | | | | |
| 12 | | | | | R4 | | | | |

2- برای ساخت جدول LALR(1) باید state هایی که فقط در lookahead با هم تفاوت دارند را ادغام در جای آن‌ها یک state با اجتماع lookahead ها قرار دهیم. در مثال بالا فقط دو state 9 و 5 این حالت را دارند که و را یک‌تک آن در طبق زیری سازیم:

$$G \rightarrow w \cdot \{x, z\}$$

$$H \rightarrow w \cdot \{x, z\}$$

بنابراین جدول LALR به صورت زیر خواهد بود: صفحه بعد

جدول یاری LALR(1)

action

goto

| state | w | x | y | z | \$ | | S | G | H |
|-------|-----|--------|----|--------|-----|--|---|---|---|
| 0 | S59 | | S3 | | | | 1 | 2 | 4 |
| 1 | | | | | acc | | | | |
| 2 | | S6 | | | | | | | |
| 3 | S9 | | | | | | | 7 | 8 |
| 4 | | | | S10 | | | | | |
| 59 | | R5, R6 | | R5, R6 | | | | | |
| 6 | | | | | R1 | | | | |
| 7 | | | | S11 | | | | | |
| 8 | | S12 | | | | | | | |
| | | | | | | | | | |
| 10 | | | | | R3 | | | | |
| 11 | | | | | R2 | | | | |
| 12 | | | | | R4 | | | | |

3- می دانیم که گرامر مد نظر از نوع LR(1) است. زیرا هیچ تداخل چه از نوع shift-reduce و چه از نوع reduce-reduce در جدول یاری LR(1) دیده نمی شود و می توان هر رشته ای را که عضو زبان گرامر باشد به کمک آن جدول یاری کرد.

از طرفی می دانیم که این گرامر LR(1) نیست، زیرا در state 59 تداخل reduce-reduce داریم و یاری نمی تواند تصمیم بگیرد که با تاقول پنجم یا ششم دردی را reduce کند. برای مثال رشته ورودی $wz\$$ یا $wx\$$ را در نظر بگیرید در ابتدای shift داده و به state 59 می ریم و در آن state با توجه به تداخل reduce-reduce نمی توانیم یاری کردن رشته را ادامه می دهیم.

اثبات وجود wx و wz در گرامر داده شده:

$$S \xrightarrow{S \rightarrow Gx} Gx \xrightarrow{G \rightarrow w} wx \quad S \xrightarrow{S \rightarrow Hz} Hz \xrightarrow{H \rightarrow w} wz$$

مسئله ۴.

گرامر زیر را در نظر بگیرید.

$$0. S \rightarrow A$$

$$1. A \rightarrow By$$

$$2. A \rightarrow \epsilon$$

$$3. B \rightarrow \epsilon$$

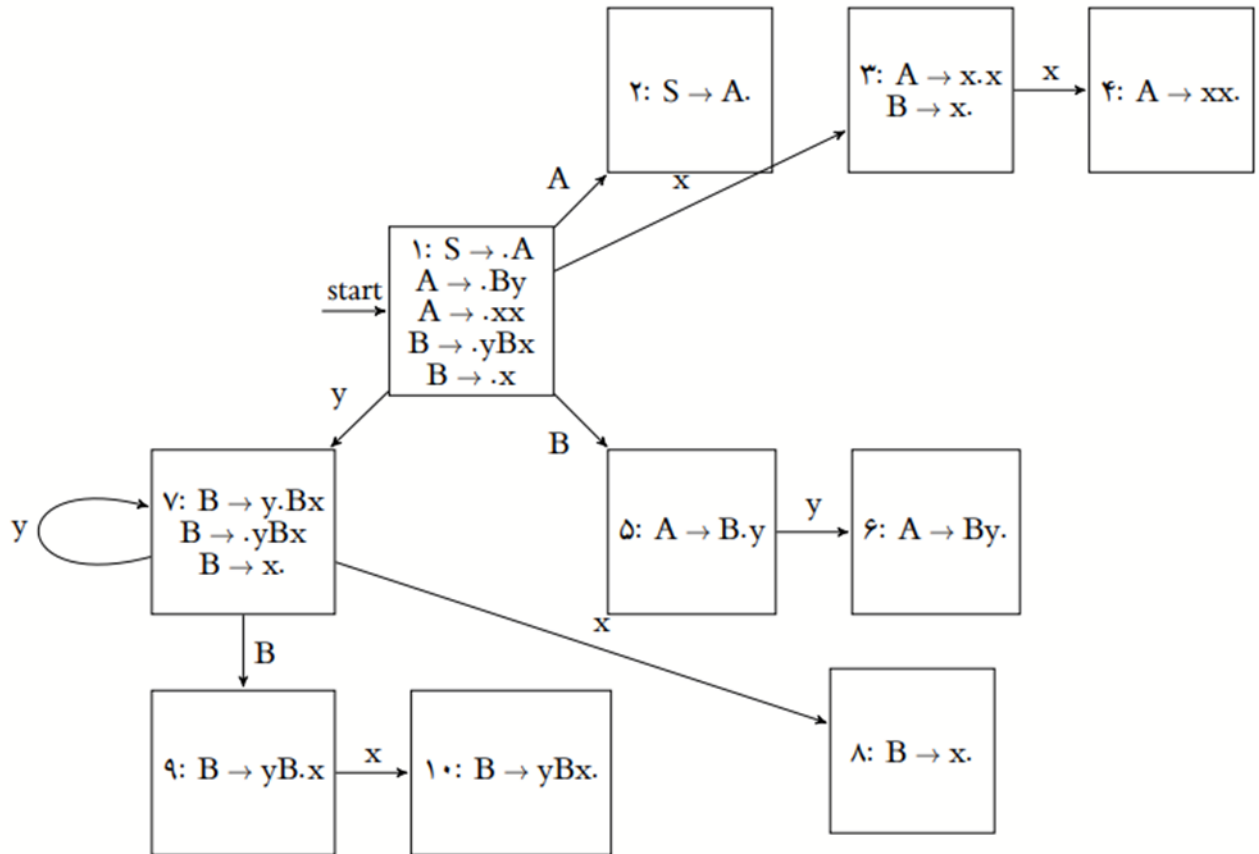
$$4. B \rightarrow yBx$$

$$S \rightarrow A$$

$$A \rightarrow By \mid \epsilon$$

$$B \rightarrow x \mid yBx$$

شکل زیر ترنزیشن دیاگرام LR (۰) مربوط به این گرامر است.



۱. توضیح دهید چرا این گرامر (۱) SLR نیست.

برای ساخت جدول یاسی (SLR(۱)، در حالت ۳، با ردی ϵ ، shift-reduce conflict پیش آمده و S4 و R4 در یک خانه قرار می‌گیرند.

۲. مجموعه FOLLOW غیرپایانه‌ها را مشخص کنید.

$$\text{First}(S) = \{\epsilon, y\}$$

$$\text{Follow}(S) = \{\$ \}$$

$$\text{First}(A) = \{\epsilon, y\}$$

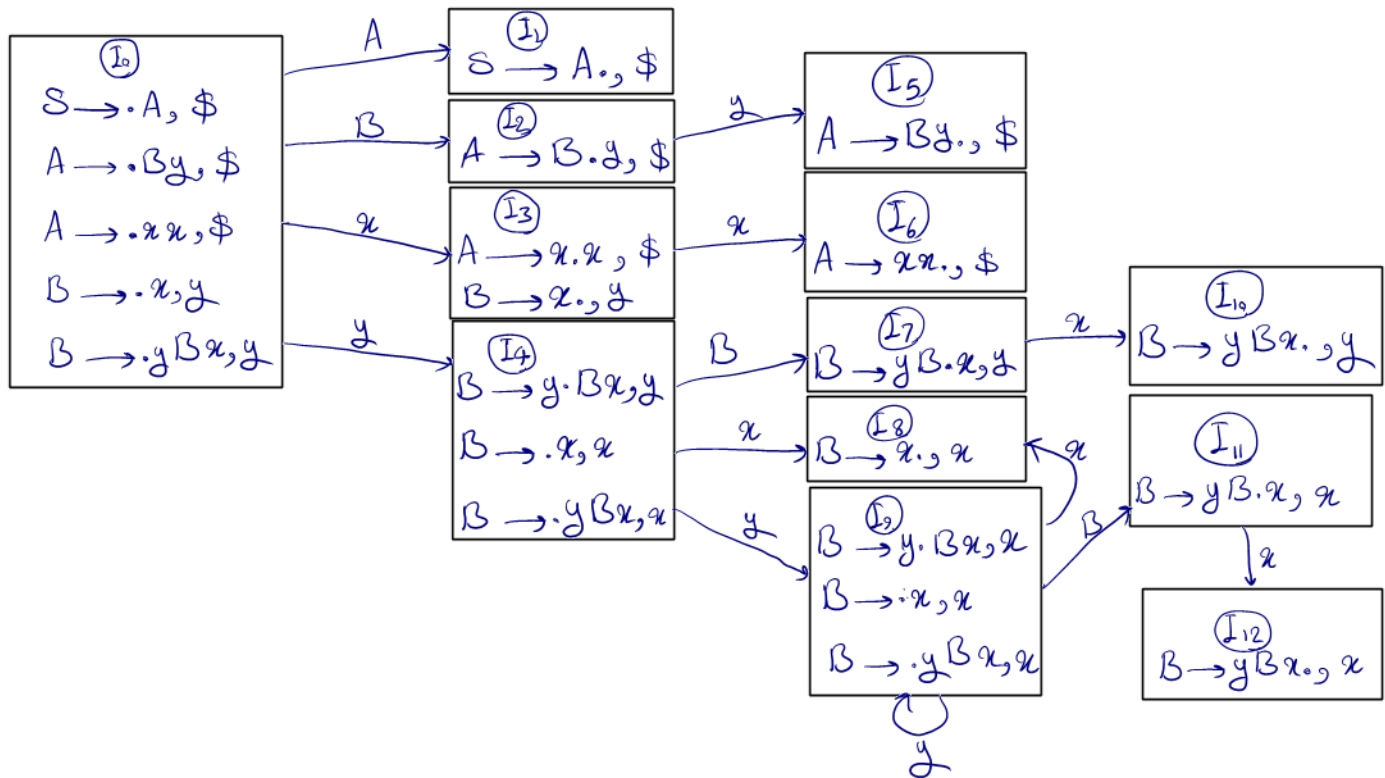
$$\text{Follow}(A) = \{\$ \}$$

$$\text{First}(B) = \{\epsilon, y\}$$

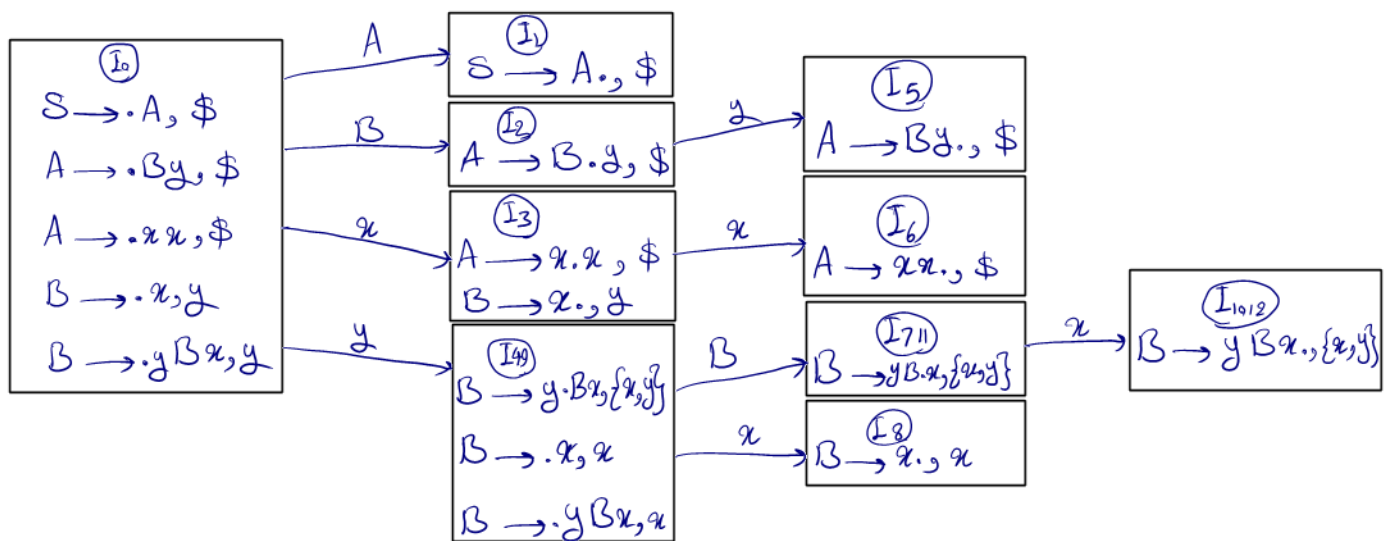
$$\text{Follow}(B) = \{\epsilon, y\}$$

۳. حال با توجه به مجموعه FOLLOW ها و ترنزشن دیاگرام، مجموعه LALR(1) lookahead ها را برای آیت‌های reduce مشخص کنید.

آیت‌های LR(1)، را رسم کنید و با ترکیب آن‌ها همچون سوال 3، آیت‌های LALR(1) را می‌سازیم:



در ترنزشن دیاگرام LALR(1)، آیت‌های 1, 4, 7, 11 - 12، ترکیبی می‌شوند و آیت‌ها به صورت زیر خواهند بود:



آیت‌های reduce در ترنزشن دیاگرام LALR(1)، I₁، I₃، I₅، I₆، I₈ و I_{10/12} هستند.

۴. آیا این گرامر (۱) LALR است؟ چرا؟

جدول پارس (۱) LALR را رسم کنید:

| State | action | | \$ | | goto | |
|-------|--------|-----|-----|--|------|-----|
| | x | y | | | A | B |
| 0 | S3 | S49 | | | 1 | 2 |
| 1 | | | acc | | | |
| 2 | | S5 | | | | |
| 3 | S6 | R3 | | | | |
| 49 | S8 | | | | | 711 |
| 5 | | | R1 | | | |
| 6 | | | R2 | | | |
| 711 | S1012 | | | | | |
| 8 | R3 | | | | | |
| 1012 | R4 | R4 | | | | |

همانطور که مشخص است در جدول پارس هیچ گونه تعارض (یعنی از نوع shift-reduce و یا از نوع reduce-reduce) وجود ندارد. بنابراین هر گرامر LR ساده شده (۱) LALR است.

۵. آیا این گرامر (۱) LR است؟ چرا؟

باقی به این که تصمیم گیرنده (۱) LALR است و از طریق می دانیم $LALR(k) \subseteq LR(k)$. بنابراین هر گرامر LALR یک گرامر LR (۱) نیز می باشد.

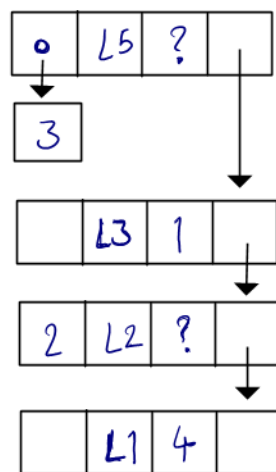
مسئله ۵.

جدول داده شده را مطابق لکچر نوت ۸ برای برنامه داده شده تکمیل نمایید. دقت بفرمایید پاسخ و جدول نهایی کافی نیست و برای هر خط توضیح لازم و همچنین همراه با linked list های لازم ارائه کنید. آیا برنامه به پاسخ درست و نهایی می‌رسد؟

```

0. goto L5
1. L3: Stmt1
2.      goto L2
3.      goto L5
4. L1: Stmt2
5. L5: Stmt3
6.      goto L3
7.      goto L5
8.      goto L2
9. L4: Stmt4
10.     goto L1
11.     goto L4
12. L2: Stmt5
    
```

| i | PB[i] |
|---|-------------|
| 0 | (د، د؟، د؟) |
| 1 | stmt1 |
| 2 | (د، د؟، د؟) |
| 3 | (د، د؟، د؟) |
| 4 | stmt2 |



جدول PB و linked-list برای خطوط ۴ مطابق ردیفی باشد.
در ادامه هر خط توضیح داده می‌شود:

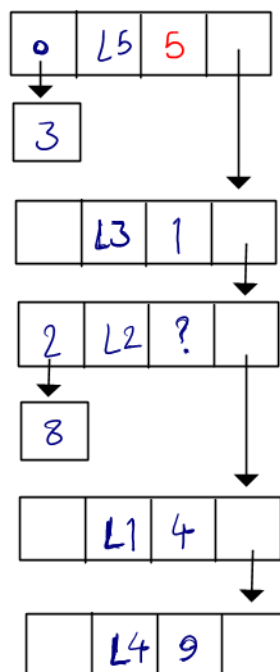
۲: در جدول من به مکان نامشخص انجام شده در linked-list نیز پیدا
label مربوط را قرار می‌دهیم (L5 یا پیدا)

۱: stmt1 تبدیل شده در linked-list مقصد L3 مشخص می‌شود.

۲: در جدول من به مکان نامشخص انجام شده در linked-list نیز پیدا
label مربوط را قرار می‌دهیم (L2 یا پیدا)

۳: در جدول من به مکان نامشخص انجام شده در linked-list مربوط به L5 که از پیش موجود است، ۳ به مبداها اضافه می‌شود.

| i | PB[i] |
|----|--------------|
| 0 | (د، د 5، د؟) |
| 1 | stmt1 |
| 2 | (د، د؟، د؟) |
| 3 | (د، د 5، د؟) |
| 4 | stmt2 |
| 5 | stmt3 |
| 6 | (و، واد، د؟) |
| 7 | (و، و 5، د؟) |
| 8 | (د، د؟، د؟) |
| 9 | stmt4 |
| 10 | (د، د 4، د؟) |
| 11 | (د، د 9، د؟) |



۴: stmt2 تبدیل شده در linked-list مقصد L1 مشخص می‌شود.

جدول PB و linked-list برای خطوط ۴ تا ۱۱ مطابق ردیفی است:

۵: stmt3 تبدیل می‌شود. مقصد L5 مشخص شده و در جدول د

update، linked-list می‌شود.

۶: من به L3 یا مقصد مشخص (۱) طبق linked-list انجام شده و به جدول
اضافه می‌شود.

۷: من به L5 یا مقصد مشخص (۵) طبق linked-list انجام شده و به جدول
اضافه می‌شود.

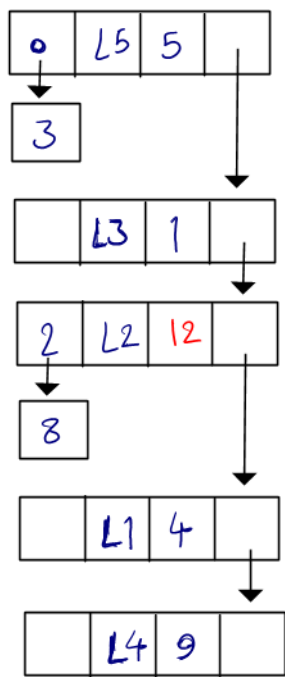
۸: در جدول من به مکان نامشخص انجام شده در linked-list مربوط به L2 که
از پیش موجود است، ۸ به مبداها اضافه می‌شود.

۹: stmt4 تبدیل شده و در linked-list مقصد L4 را مشخص می‌کنیم. (۹)

۱۰: من به L1 یا مقصد مشخص (۴) طبق linked-list انجام شده و به جدول اضافه می‌شود.

۱۱: من به L4 یا مقصد مشخص (۹) طبق linked-list انجام شده و به جدول اضافه می‌شود.

| i | PB[i] |
|----|------------------------|
| 0 | (د، 5، 5) (jp, 5, 5) |
| 1 | stmt1 |
| 2 | (د، 12، 5) (jp, 12, 5) |
| 3 | (د، 5، 5) (jp, 5, 5) |
| 4 | stmt2 |
| 5 | stmt3 |
| 6 | (و، 1، 4) (w, 1, 4) |
| 7 | (د، 5، 5) (jp, 5, 5) |
| 8 | (د، 12، 5) (jp, 12, 5) |
| 9 | stmt4 |
| 10 | (د، 4، 5) (jp, 4, 5) |
| 11 | (د، 9، 5) (jp, 9, 5) |
| 12 | stmt5 |



در نهایت افزاین خط ۱۲ جدول PB و linked-list به شکل گاهی زیر درمی آید:

۱۲. stmt5 تبدیل می شود. مقصد L2 مشخص شده در linked-list و جدول، update می شود.

مشخص است که در هنگام اجرای برنامه (نه در حین تولید کدیانی)، با شروع اجرا از خط ۵، به خط ۵، می کنیز. stmt3 اجرای شود. با خواندن خط ۶ به خط ۱ می کنیز. stmt1 اجرای شود. به خط ۲ می ریز و طبق آن به خط ۱۲ می کنیز.

stmt5 اجرا شده و در نهایت برنامه پایان می یابد. در نتیجه در صورت عدم وجود error در stmt های ۵، ۳ و ۱، برنامه به درستی اجرا شده و به پاسخ درست و گاهی می رسد.