

پایخ تکلیف شماره ۳ کاسلر

$$S \rightarrow SS \mid \epsilon \quad \text{سوال ۱}$$

الف) عبارت منظم معادل با زبان گرامر بالا ؟ ϵ^*

ب) چرا این گرامر $LL(1)$ نیست ؟ $S \rightarrow \epsilon, First(SS) \cap First(\epsilon) =$

$$\{\epsilon, \epsilon\} \cap \{\epsilon\} = \{\epsilon\} \neq \emptyset$$

در نتیجه $LL(1)$ نیست

ج) صر، فاعده $S \rightarrow SS$ همواره شکل سازاست و با اضافه کردن یک یا چند فاعده S

دیگر می توان این گرامر را به یک گرامر $LL(1)$ معادل تبدیل کرد بلکه باید بازگشت از چپ این فاعده را

برطرف کرد.

د) یک گرامر $LL(1)$ برای زبان تولید شده توسط این گرامر ؟ $S \rightarrow \epsilon \mid S \mid \epsilon$

$$S \rightarrow \epsilon, First(\epsilon S) \cap First(\epsilon) = \emptyset \quad \checkmark$$

$$, First(S) \cap Follow(S) = \emptyset \quad \checkmark$$

در نتیجه این گرامر $LL(1)$ است.

سؤال ۲) گرامر زیر را می توان به روش تردلی بازسی تجزیه کرد. با نوشتن الگوریتم تجزیه گر به روش تردلی بازسی مشکل این گرامر را پیدا کنید.

$$S \rightarrow 0S \mid 0 \mid 1$$

$$\text{First}(0S) = \{0\} \quad \text{First}(0) = \{0\} \quad \text{First}(1) = \{1\}$$

```
void S () {
```

```
    if (lookahead == '0') { match ('0') ; S (); }
```

```
    else if (lookahead == '0') { match ('0') ; }
```

```
    else if (lookahead == '1') { match ('1') ; }
```

```
    else error ();
```

```
}
```

اگر lookahead برابر صفر باشد وارد if اول شده و else if اول هیچ گاه اجرا نمی شود.

recursive descent parser

- ابتدا باید گرامر LL(1) معادل با گرامر داده شده را بیابیم. اگر این کار ممکن نباشد، برای آن زبان پارسر RD نمی توان نوشت.

$G = (N, T, S, P)$

LL(1) Grammar : if $A \rightarrow \alpha | \beta \in P$
 then $First(\alpha) \cap First(\beta) = \emptyset$
 and $(\alpha \Rightarrow^* \epsilon) \Rightarrow First(\beta) \cap Follow(A) = \emptyset$

الف) $S \rightarrow OS1 | O1$ LL(1) نیست
 $S \rightarrow OA \quad A \rightarrow S1 | 1$ LL(1) است $\xrightarrow{\text{زیرا}}$ $\begin{cases} First(S1) = \{0\} \cap First(1) = \emptyset \\ S1 \not\Rightarrow^* \epsilon, 1 \not\Rightarrow^* \epsilon \end{cases}$ ناکتورگیری از چپ

```
void match (token t)
{
    if (lookAhead == t)
        lookAhead = nextToken();
    else error();
}
```

```
void S()
{
    if (lookahead == '0')
    {
        match('0'); A();
    }
    else error();
}

void A()
{
    if (lookahead == '0')
    {
        S(); match('1');
    }
    else if (lookahead == '1') match('1');
    else error();
}
```

ب) $S \rightarrow +SS | -SS | a$ LL(1) است $\xrightarrow{\text{زیرا}}$ $\begin{cases} First(+SS) \cap First(-SS) \cap First(a) = \emptyset \\ +SS, -SS, a \not\Rightarrow^* \epsilon \end{cases}$

```
void S()
{
    if (lookahead == '+') { match('+'); S(); S(); }
    if (lookahead == '-') { match('-'); S(); S(); }
    if (lookahead == 'a') match('a');
    else error();
}
```

پ) $S \rightarrow \overbrace{S(S)S}^{\alpha} | \epsilon$ LL(1) نیست
 $S \rightarrow \epsilon A, A \rightarrow (S)SA | \epsilon \equiv S \rightarrow A, A \rightarrow (S)SA | \epsilon$
 $\equiv S \rightarrow \epsilon, S \rightarrow A, A \rightarrow (S)S, A \rightarrow (S)SA \equiv S \rightarrow (S)S | \epsilon$
 $\begin{cases} First((S)S) \cap First(\epsilon) = \emptyset \\ \epsilon \Rightarrow^* \epsilon \Rightarrow (Follow(S) = \{ \$, \}) \cap First((S)S) = \emptyset \end{cases}$ زیرا:

حذف چپگردی ؛
حذف قاعده ϵ در یک

ناکتورگیری از چپ و حذف چپگردی با هم انجام را از بین می برد.

```
void S()
{
    if (lookahead == '(') { match('('); S(); match(')'); S(); }
    else if (lookahead == '$' || lookahead == ')') return;
    else error();
}
```

$S \rightarrow (L) | a$
 $L \rightarrow L, S | S$

$S \rightarrow (L) | a$
 $L \rightarrow SA$
 $A \rightarrow , SA | \epsilon$

$First((L)) \cap First(a) = \{ (\} \cap \{ a \} = \emptyset$
 $First(, SA) \cap First(\epsilon) = \{ , \} \cap \{ \epsilon \} = \emptyset$
 $\epsilon \Rightarrow^* \epsilon, First(, SA) \cap Follow(A) = \{ , \} \cap \{ \} = \emptyset$

(۴)

```

void match(token t)
{
  if (lookahead == t)
    lookahead = nextToken();
  else error();
}

```

```

void S()
{
  if (lookahead == '(') { match('('); L(); match(')'); }
  else if (lookahead == 'a') match('a');
  else error();
}

```

```

void A()
{
  if (lookahead == ',') { match(','); S(); A(); }
  else if (lookahead == ')') return;
  else error();
}

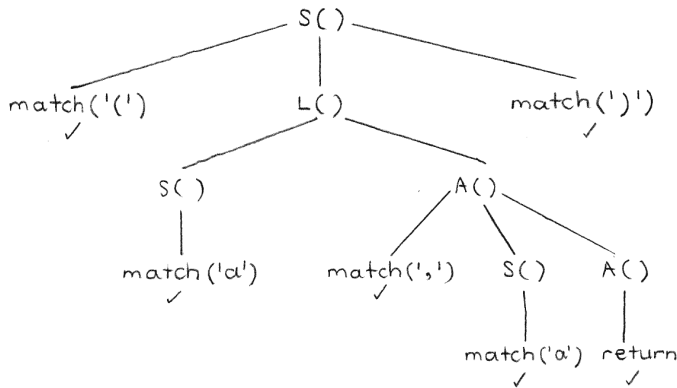
```

```

void L()
{
  if (lookahead == '(' || lookahead == 'a')
    { S(); A(); }
  else error();
}

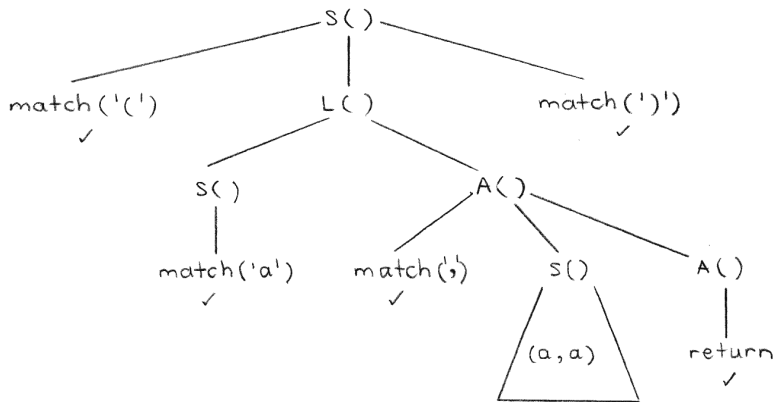
```

(a, a)



درخت خوانی

(a, (a, a))



(۵) برای هر گرامر زیر مجموعه‌های First و Follow را برای ناپایانها محاسبه کنید :

$$\begin{cases} S' \rightarrow S\$ \\ S \rightarrow AdB \\ A \rightarrow aA | \epsilon \\ B \rightarrow bBS | \epsilon \end{cases}$$

$$\text{First}(S) = \text{First}(AdB) = (\text{First}(A) - \{\epsilon\}) \cup \text{First}(dB) = \{a\} \cup \{d\} = \{a, d\}$$

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

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

$$\text{Follow}(S) = \{\$\} \cup \text{Follow}(B) = \{\$, a, d\}$$

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

$$\text{Follow}(B) = \text{Follow}(S) \cup \text{First}(S) = \{\$\} \cup \underbrace{\text{Follow}(B)}_{\text{تاییدی ندارد!}} \cup \text{First}(S) = \{\$, a, d\}$$

پایخ تکلیف شماره ۳ کاسابلر

سؤال ۶) مشخص کنید که کدام یک از گرامرهای زیر LL(1) هستند.

الف) $S \rightarrow aAb, A \rightarrow dA | \epsilon$

$A \Rightarrow^* \epsilon$

$\left. \begin{aligned} \text{First}(dA) \cap \text{First}(\epsilon) &= \emptyset \\ \text{First}(A) \cap \text{Follow}(A) &= \emptyset \end{aligned} \right\} \Rightarrow$ این گرامر LL(1) است.

ب) $S \rightarrow aAb, A \rightarrow bA | \epsilon$

$A \Rightarrow^* \epsilon$

$\text{First}(A) \cap \text{Follow}(A) = \{b\} \neq \emptyset \Rightarrow$ این گرامر LL(1) نیست.

ج) $S \rightarrow ABCDE | f$

$A \rightarrow aA | B$

$A \Rightarrow^* \epsilon$

$B \rightarrow bB | \epsilon$

$\text{First}(A) \cap \text{Follow}(A) = \{a, b\} \cap \{b, c\} = \{b\} \neq \emptyset$

$C \rightarrow cC | d$

$D \rightarrow dD | Be$

شرط لازم برای LL(1) را ندارد پس این گرامر LL(1) نیست.

$E \rightarrow gE | \epsilon$

$S \rightarrow aSbS | bSaS | \epsilon$

خیزد: $S \rightarrow A | \epsilon$

$\equiv A \rightarrow aAbA | bAaA |$

$aAb | abA | ab |$

$bAa | baA | ba$

bool match(Token t)

{ if (*lookahead == t)

{ lookahead ++;

return true;

else { lookahead ++; return false; }

}

// return *lookahead ++ = t;

نکته: lookahead آرایه توکنهای ورودی است که در ابتدا توسط scanner آماده می‌شود.

bool S()

{ if (*lookahead == '\$') return true;

else return A(); }

bool A()

{ Token * save = lookahead;

return (lookahead = save, match('a') && A() && match('b') && A()) ||

(lookahead = save, match('b') && A() && match('a') && A()) ||

(lookahead = save, match('a') && A() && match('b')) ||

(lookahead = save, match('a') && match('b') && A()) ||

(lookahead = save, match('a') && match('b')) ||

(lookahead = save, match('b') && A() && match('a')) ||

(lookahead = save, match('b') && match('a') && A()) ||

(lookahead = save, match('b') && match('a')) ;

}

برای این گرامر، گرامر معادل زیر وجود دارد که LL(1) است:

$S \rightarrow aAbs | bBaS | \epsilon$

$First(aAbs) = \{a\}$, $First(bBaS) = \{b\}$, $First(\epsilon) = \{\epsilon\}$

$A \rightarrow aAbA | \epsilon$

$First(aAbA) \cap First(\epsilon) = \{a\} \cap \{\epsilon\} = \emptyset$

$B \rightarrow bBaB | \epsilon$

$First(bBaB) \cap First(\epsilon) = \{b\} \cap \{\epsilon\} = \emptyset$

$(First(aAbs) \cup First(bBaS)) \cap Follow(S) = \{a, b\} \cap \{\$ \} = \emptyset$

$First(aAbA) \cap Follow(A) = \{a\} \cap \{b\} = \emptyset$

$First(bBaB) \cap Follow(B) = \{b\} \cap \{a\} = \emptyset$

پس می‌توان پارسی‌ترین گرامر برای آن ساخت:

void S()

{ if (lookahead == 'a') { match('a'); A(); match('b'); S(); }

else if (lookahead == 'b') { match('b'); B(); match('a'); S(); }

else if (lookahead == '\$') return;

else error(); }

void A()

{ if (lookahead == 'a') { match('a'); A(); match('b'); A(); }

else if (lookahead == 'b') return; else error(); }

void B()

{ if (lookahead == 'b') { match('b'); B(); match('a'); B(); }

else if (lookahead == 'a') return; else error(); }