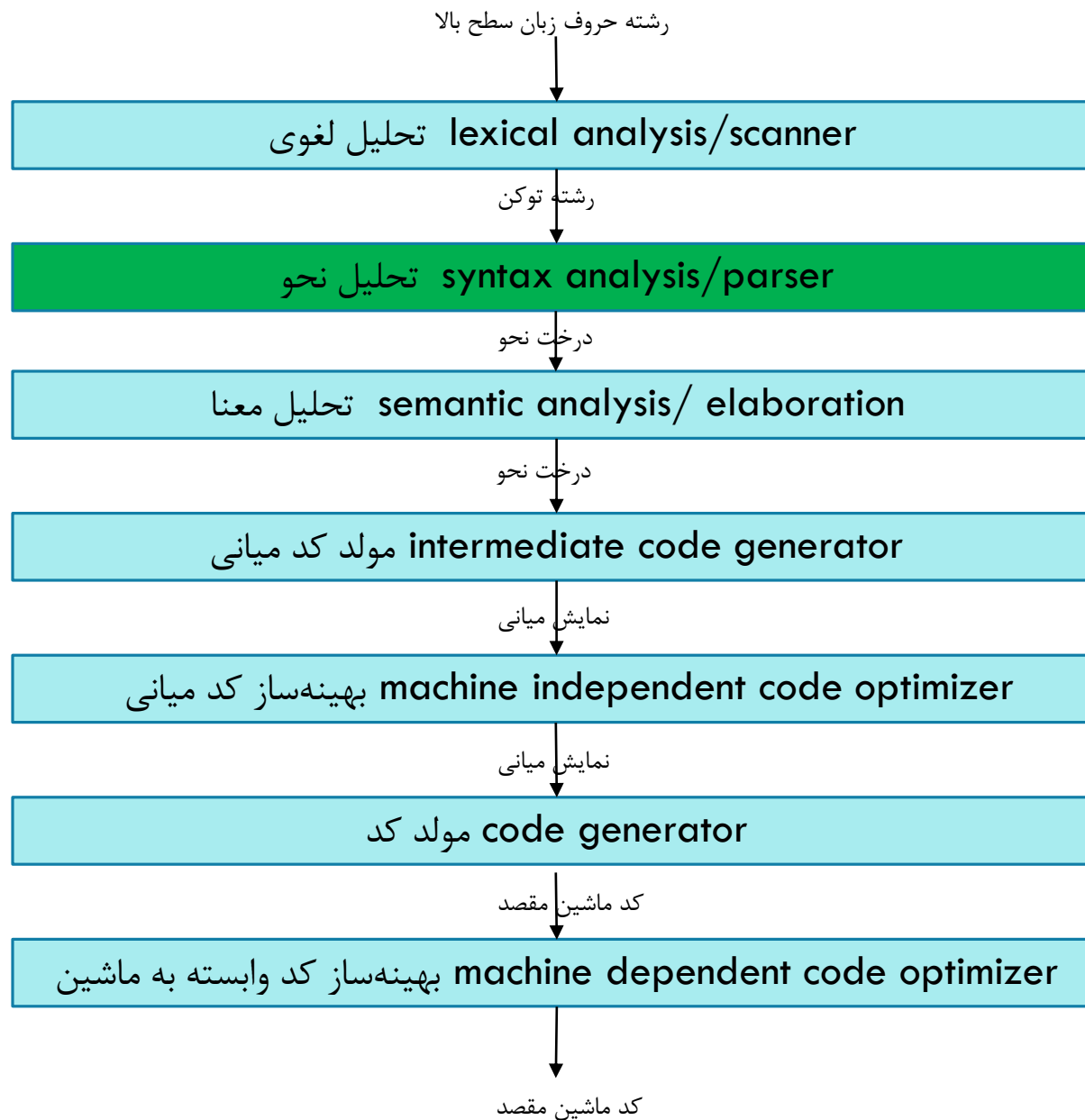


کامپیایدر
تجزیه گر
تجزیه بالابه پائین
فاکتور گیری چپ، چچ (۱)

محسن هوشمند
دانشکده تکنولوژی اطلاعات و علم رایانه
دانشگاه تحصیلات تکمیلی علوم پایه زنجان



مثال محاسبه کلی مجموعه آغاز و پیرو و متغیرهای پوچ پذیر

$$0 S \rightarrow E \$$$

$$1 E \rightarrow E + T$$

$$2 E \rightarrow E - T$$

$$3 E \rightarrow T$$

$$4 T \rightarrow T * F$$

$$5 T \rightarrow T / F$$

$$6 T \rightarrow F$$

$$7 F \rightarrow (E)$$

$$8 F \rightarrow num$$

$$9 F \rightarrow id$$

پیرو	آغاز	
\$	id num (S
\$+-)	id num (E
\$+-)* /	id num (T
\$+-)* /	id num (F

چگونگی محاسبه مجموعه آغاز

۱- اگر x حرف الفباء، آن گاه $\{x\} = First(x)$

۲- اگر $\epsilon \rightarrow x$ جزو قواعد تولید، آن گاه افزودن ϵ به مجموعه آغاز

۳- اگر $X \rightarrow Y_1 Y_2 Y_3 \dots Y_n$ جزو قواعد تولید،

الف- $First(X) = First(Y_1)$

ب- اگر $\epsilon \in First(Y_1)$ ، آن گاه $First(X) = \{First(Y_1) - \epsilon\} \cup \{First(Y_2)\}$

ج- اگر به ازای i تا n ، $\epsilon \in First(Y_i)$ ، آن گاه $First(Y_{i+1}) \in First(X)$

د- اگر به ازای تمامی Y_i ها و $i = 1:n$ داشته باشیم $\epsilon \in First(Y_i)$ ، آن گاه $\epsilon \in First(X)$

چگونگی محاسبه مجموعه پیرو

۱- پیرو متغیر شروع $\$, \$ = Follow(S)$

۲- اگر $A \rightarrow mBn$ ، آن گاه $\{ \epsilon \} = Follow(B) - First(n)$

۳- اگر $A \rightarrow mB$ ، آن گاه $Follow(B)$ دارای $Follow(A)$ است.

۴- اگر $A \rightarrow mBn$ و $\epsilon \in First(n)$ ، آن گاه

$Follow(B)$ شامل $\{First(n) - \epsilon\} \cup Follow(A)$

مثال - تولید جدول پیش بین

$Z \rightarrow d$
 $Z \rightarrow XYZ$
 $Y \rightarrow \epsilon$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

پيرو	آغاز	
acd	a,c ϵ	X
acd	c, ϵ	Y
	acd	Z

	a	c	d
X	$X \rightarrow a$ $X \rightarrow Y$	$X \rightarrow Y$	$X \rightarrow Y$
Y	$Y \rightarrow \epsilon$	$Y \rightarrow \epsilon$ $Y \rightarrow c$	$Y \rightarrow \epsilon$
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$ $Z \rightarrow XYZ$

- چگونگی ساخت:
- قرارگیری قانون تولید $X \rightarrow \gamma$ در ردیف X و ستون T که T عضوی از آغاز γ
- در صورتی که γ بوج پذیر، قرارگیری در ردیف X و ستون T که T عضوی از پیرو X

نزول-بازگشتی و جدول

دستور مبهم همیشه منجر به چند قانون در یک مدخل

نیاز به یافتن دستور غیر مبهم

جدول با مدخل‌های حداکثر یک عضوی

▪ چچ(۱)

▪ تجزیه از چپ به راست، اشتقاق چپ‌ترین، جلوبینی یک-علامت

امکان تعمیم مفهوم مجموعه‌های آغاز توضیح‌دهنده k تکه رشته

▪ چچ(k)

▪ استفاده کم - چرا؟

▪ نیاز به جدول بزرگ (افزایش پیچیدگی)

دوباره موضوع ابهام

$$0 S \rightarrow E \$$$

$$1 E \rightarrow E + T$$

$$2 E \rightarrow E - T$$

$$3 E \rightarrow T$$

$$4 T \rightarrow T * F$$

$$5 T \rightarrow T / F$$

$$6 T \rightarrow F$$

$$7 F \rightarrow (E)$$

$$8 F \rightarrow num$$

$$9 F \rightarrow id$$

به دنبال تولید تجزیه گر پیش بین جهت دستور روبرو

$E \rightarrow T$ و $E \rightarrow E + T$ مسئله ساز تجزیه چچ (۱)

▪ آغاز T و E یکسان

▪ دلیل؟

بازگشت چپ

▪ متغیر سمت چپ قاعده باز اولین متغیر در سمت راست قاعده باشد

دستور با بازگشت چپ چچ (۱) نیست

بازگشت چپ

بازنویسی دستورات تاثیر قوی بر تعداد پس روی

شماره قانون اجرائی	صورت جمله‌ای	ورودی
0	E	$\uparrow x - 2 \times y$
۱	$E + T$	$\uparrow x - 2 \times y$
۱	$E + T + T$	$\uparrow x - 2 \times y$
۱	$E + T + T + \dots$	$\uparrow x - 2 \times y$
۱	$E + T + T + \dots$	$\uparrow x - 2 \times y$
۱	...	$\uparrow x - 2 \times y$

مشکل: ترکیب تبپ و بازگشت چپ

قاعده تولید بازگشت چپ: قاعده تولیدی که اولین علامت سمت راست جاگذاری با علامت سمت چپ یکی باشد

$$A \rightarrow A\alpha$$

یا علامت سمت چپ در سمت راست تکرار شود و تمام علامت‌های قبل آن به تهی منتهی شوند

حذف بازگشت چپ

قاعده کلی تر

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \cdots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \cdots \mid \beta_n$$

متغیری با چند رابطه بازگشت چپ

تبدیل به

$$\begin{aligned} A &\rightarrow \beta_1 A' \mid \beta_2 A' \mid \cdots \mid \beta_n A' \\ A' &\rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \cdots \mid \alpha_m A' \mid \epsilon \end{aligned}$$

	+	-	*	/	id	num	()	\$
S					$S \rightarrow E \$$	$S \rightarrow E \$$	$S \rightarrow E \$$		
E					$E \rightarrow T E'$	$E \rightarrow T E'$	$E \rightarrow T E'$		
E'	$E' \rightarrow + T E'$	$E' \rightarrow - T E'$						$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T					$T \rightarrow F T'$	$T \rightarrow F T'$	$T \rightarrow F T'$		
T'	$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$	$T' \rightarrow * F T'$	$T' \rightarrow / F T'$				$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F					$F \rightarrow id$	$F \rightarrow num$	$F \rightarrow (E)$		

0 $S \rightarrow E \$$

1 $E \rightarrow T E'$

2 $E' \rightarrow + T E'$

3 $E' \rightarrow - T E'$

4 $E' \rightarrow \epsilon$

5 $T \rightarrow F T'$

6 $T' \rightarrow * F T'$

7 $T' \rightarrow / F T'$

8 $T' \rightarrow \epsilon$

9 $F \rightarrow (E)$

10 $F \rightarrow num$

11 $F \rightarrow id$

شماره قانون اجرائی	صورت جمله‌ای	ورودی
0	$E \$$	$\uparrow x - 2 \times y \$$
1	$T E' \$$	$\uparrow x - 2 \times y \$$
5	$F T' E' \$$	$\uparrow x - 2 \times y \$$
11	$id T' E' \$$	$\uparrow x - 2 \times y \$$
→	$id T' E' \$$	$x \uparrow - 2 \times y \$$
8	$id E' \$$	$x \uparrow - 2 \times y \$$
3	$id - T E' \$$	$x \uparrow - 2 \times y \$$
→	$id - T E' \$$	$x - \uparrow 2 \times y \$$
5	$id - F T' E' \$$	$x - \uparrow 2 \times y \$$
10	$id - num T' E' \$$	$x - \uparrow 2 \times y \$$
→	$id - num T' E' \$$	$x - 2 \uparrow \times y \$$
6	$id - num \times F T' E' \$$	$x - 2 \uparrow \times y \$$
→	$id - num \times F T' E' \$$	$x - 2 \times \uparrow y \$$
11	$id - num \times id T' E' \$$	$x - 2 \times \uparrow y \$$
→	$id - num \times id T' E' \$$	$x - 2 \times y \uparrow \$$
8	$id - num \times id E' \$$	$x - 2 \times y \uparrow \$$
4	$id - num \times id \$$	$x - 2 \times y \uparrow \$$

حذف
بازگشت چپ
- مثال ادامه

شبه کد مربوط به دستور-مثال

0 $S \rightarrow E \$$

1 $E \rightarrow T E'$

2 $E' \rightarrow + T E'$

3 $E' \rightarrow - T E'$

4 $E' \rightarrow \epsilon$

5 $T \rightarrow F T'$

6 $T' \rightarrow * F T'$

7 $T' \rightarrow / F T'$

8 $T' \rightarrow \epsilon$

9 $F \rightarrow (E)$

10 $F \rightarrow num$

11 $F \rightarrow id$

```
void T(void) {switch (tok) {
    case ID:
    case NUM:
    case LPAREN: F(); Tprime(); break;
    default: error;
}}
void Tprime(void) {switch (tok) {
    case PLUS: break;
    case MINUS: break;
    case TIMES: eat(TIMES); F(); Tprime(); break;
    case DIVIDES: eat(DIVIDES); F(); Tprime(); break;
    case EOF: break;
    case LPAREN: break;
    default: error;
}}
```

تجزیه پیش بین غیربازگشتی

استفاده از پشته به جای استفاده مستقیم از دستور بازگشتی

بازتولید اشتقاق چپ ترین

تجزیه پیش‌بین

$$0 S \rightarrow E \$$$

$$1 E \rightarrow T E'$$

$$2 E' \rightarrow + T E'$$

$$3 E' \rightarrow - T E'$$

$$4 E' \rightarrow \epsilon$$

$$5 T \rightarrow F T'$$

$$6 T' \rightarrow * F T'$$

$$7 T' \rightarrow / F T'$$

$$8 T' \rightarrow \epsilon$$

$$9 F \rightarrow (E)$$

$$10 F \rightarrow num$$

$$11 F \rightarrow id$$

در یکی از مثال‌های قبل نیاز به پس‌روی بود

امکان طراحی تجزیه‌گر بازگشت بدون نیاز به پس‌روی

مشکل کجاست؟

▪ زمانی که تجزیه‌گر قاعده تولید را انتخاب می‌کند.

▪ انتخاب دلخواهانه و بدون دقت نظر

▪ با انتخاب درست امکان جلوگیری از پس‌روی

مثال -

شماره قانون اجرائی	صورت جمله‌ای	ورودی
۸	$id E'$	$x \uparrow - 2 \times y \$$

ابهام دوباره! ابهام ابهام ابهام

11 $F \rightarrow id$
12 $| id [Elist]$
13 $| id (Elist)$
20 $Elist \rightarrow E, Elist$
21 $| E$

11 $F \rightarrow id \ arg$
12 $arg \rightarrow [Elist]$
13 $| (Elist)$
14 $| \epsilon$

مثال - فاکتور

▪ متغیر، آرایه، تابع

▪ پیش‌بین؟

▪ خیر

▪ یکسانی الفباء نخست (پیش‌وند یکسان) قوانین ۱۱ و ۱۲ و ۱۳

▪ امکان پیش‌بینی بسط درست، نگاه پیش‌رو به دو تکه اجازه را ممکن می‌کند

▪ پیش‌بین‌سازی دستور با یک **نگاه پیش‌رو**

▪ تعریف متغیری جدید جهت نمایش موارد اختلاف

▪ معروف به فاکتورگیری چپ جهت حذف پیشوند یکسان

ابهام دوباره! - ادامه

نیاز به یافتن خاصیت پیش‌بین‌کننده تجزیه‌گر
؟

$$A \rightarrow \alpha | \beta$$

- نیاز به تمایز الفباء حاصل از α از الفباء حاصل از β
- $First(\alpha)$ مجموعه تکه‌هایی که به عنوان نخستین علامت رشته‌های حاصل از α ظاهر می‌شوند،

نیاز به آغازگیری

$$First(\alpha) \cap First(\beta) = \emptyset$$

- نیاز به خاصیت مطلوب

$$\forall A \rightarrow \alpha_1 | \alpha_2 | \alpha_3 | \dots | \alpha_n$$

$$First(\alpha_1) \cap First(\alpha_2) \cap First(\alpha_3) \cap \dots \cap First(\alpha_n) = \emptyset$$

- حصول این خاصیت، عامل وقوع تجزیه پیش‌بین‌دستور
- معروف به خاصیت چچ (1) $LL(1)$
- همه دستورات پیش‌بین‌نیستند.

دستور نامعین یا مسئله پیش‌وند یکسان

Nondeterministic grammar or common prefix problem

فاکتورگیری چپ جهت حذف پس رو

الگوریتم - فاکتورگیری چپ

برای هر متغیر A عضو دستور

یافتن طولانی‌ترین پیشوند α مشترک بین دو یا چند طرف راست قانون

اگر $\alpha \neq \epsilon$ آنگاه

جاگذاری قانون بسط A

$$A \rightarrow \alpha\beta_1 | \alpha\beta_2 | \dots | \alpha\beta_n | \gamma$$

با

$$A \rightarrow \alpha \zeta | \gamma$$

$$\zeta \rightarrow \beta_1 | \beta_2 | \dots | \beta_n$$

افزودن ζ به متغیرها

تکرار تا زمان باقی‌نماندن پیشوند مشترک

فاکتورگیری چپ - مثال

$S \rightarrow iEtS|iEtSeS|a|b$



$S \rightarrow iEtSS'|a|b$
 $S' \rightarrow eS|\epsilon$

$S \rightarrow a|ab|abc|abcd|e|f$



$S \rightarrow aS_1|e|f$
 $S_1 \rightarrow bS_2|\epsilon$
 $S_2 \rightarrow cS_3|\epsilon$
 $S_3 \rightarrow d|\epsilon$

تجزیه چچ (۱) - ایجاد جدول تجزیه چچ (۱)

یافتن مجموعه‌های آغاز تمامی متغیرها

یافتن مجموعه‌های پیرو تمامی متغیرها

ایجاد جدول تجزیه

- ردیف‌ها نمایش‌گر متغیرها
- ستون‌ها نمایش‌گر حروف الفبا به علاوه علامت پایان
- تمامی قوانین تولید تهی: در ستون عضو پیرو قرار می‌گیرند
- بقیه قوانین تولید در ستون مجموعه آغاز قرار می‌گیرند.

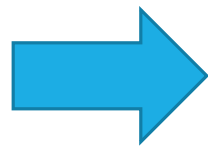
تجزیه چچ (۱) - ایجاد جدول تجزیه چچ (۱) - مثال

پیش از
حذف بازگشت چپ

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid id$$



پس از
حذف بازگشت چپ

$$E \rightarrow T E'$$

$$E' \rightarrow + T E' \mid \epsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T' \mid \epsilon$$

$$F \rightarrow (E) \mid id$$

تجزیہ چچ (۱) - ایجاد جدول تجزیہ چچ (۱) - مثال

$E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	آغاز	پیرو
$E \rightarrow T E'$	{(, id}	{\$,)}
$E' \rightarrow + T E' \mid \epsilon$	{+, \epsilon}	{\$,)}
$T \rightarrow F T'$	{id, (}	{+, \$,)}
$T' \rightarrow * F T' \mid \epsilon$	{*, \epsilon}	{+, \$,)}
$F \rightarrow (E) \mid id$	{id, (}	{*, +, \$,)}

تجزیه چچ (۱) - ایجاد جدول تجزیه چچ (۱) - مثال

	آغاز	پایان
$E \rightarrow T E'$	$\{(, id\}$	$\{ \$,)\}$
$E' \rightarrow + T E' \mid \epsilon$	$\{ +, \epsilon\}$	$\{ \$,)\}$
$T \rightarrow F T'$	$\{ id, (\}$	$\{ +, \$,)\}$
$T' \rightarrow * F T' \mid \epsilon$	$\{ *, \epsilon\}$	$\{ +, \$,)\}$
$F \rightarrow (E) \mid id$	$\{ id, (\}$	$\{ *, +, \$,)\}$

ایجاد جدول تجزیه

	ID	+	*	()	\$
E	$E \rightarrow T E'$			$E \rightarrow T E'$		
E'		$E' \rightarrow + T E'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow F T'$			$T \rightarrow F T'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow * F T'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

تجزیه چچ (۱) - ایجاد جدول تجزیه چچ (۱) - مثال ۲

$S \rightarrow A|a$
 $A \rightarrow a$

	آغاز	پيرو
$S \rightarrow A a$	{a}	{\$}
$A \rightarrow a$	{a}	{\$}

تجزیه چچ (۱) - ایجاد جدول تجزیه چچ (۱) - مثال ۲

	آغاز	پرو
$S \rightarrow A a$	{a}	{\\$}
$A \rightarrow a$	{a}	{\\$}

ایجاد جدول تجزیه

	a	\$
S	$S \rightarrow A, S \rightarrow a$	
A	$A \rightarrow a$	

تجزیة چچ (۱) - چند ویژگی سلبی

فاکتور چپ

بازگشت چپ

دستور مبهم

عبارت منظم لزوما چچ (۱) نیست

▪ ؟

تجزیه پیش‌بین - فاکتورگیری چپ جهت حذف پس‌رو

تصمیم‌ناپذیری اینکه دستور زبان مستقل از متنی پیش‌بین‌پذیر است یا نه

مثال - $\{a^n 0 b^n \mid n \geq 1\} \cup \{a n 1 b^{2n} \mid n \geq 1\}$

دستور پیش‌بین‌پذیر نیست

نزول بازگشتی

EPrime()

/ Expr' → + Term Expr' | - Term Expr' */*

if (word = + or word = -) then begin;

word ← NextWord();

if (Term())

then return EPrime();

else return false;

end;

else if (word = _ or word = eof) / Expr' → ε */*

then return true;

else begin;

/ no match */*

report a syntax error;

return false;

end;

بخشی از شبه کد مربوط به دستور

```
word ← NextWord();
push eof onto Stack;
push the start symbol, S, onto Stack;
focus ← top of Stack;
loop forever;
  if (focus = eof and word = eof)
    then report success and exit the loop;
  else if (focus ∈ T or focus = eof) then begin;
    if focus matches word then begin;
      pop Stack;
      word ← NextWord();
    end;
    else report an error looking for symbol at top of stack;
  end;
  else begin; /* focus is a nonterminal */
    if Table[focus,word] is  $A \rightarrow B_1 B_2 \dots B_k$  then begin;
      pop Stack;
      for i ← k to 1 by -1 do;
        if ( $B_i \neq \epsilon$ )
          then push  $B_i$  onto Stack;
        end;
      end;
    else report an error expanding focus;
  end;
  focus ← top of Stack;
end;
```

$$0 \ S \rightarrow E \$$$

$$1 \ E \rightarrow T E'$$

$$2 \ E' \rightarrow + T E'$$

$$3 \ E' \rightarrow - T E'$$

$$4 \ E' \rightarrow \epsilon$$

$$5 \ T \rightarrow F T'$$

$$6 \ T' \rightarrow * F T'$$

$$7 \ T' \rightarrow / F T'$$

$$8 \ T' \rightarrow \epsilon$$

$$9 \ F \rightarrow (E)$$

$$10 \ F \rightarrow num$$

$$11 \ F \rightarrow id$$

بخشی از شبکه کد مربوط به دستور

شماره قانون اجرائی	منطبق تاکنون	پشته	ورودی
0		E\$	$\uparrow x - 2 \times y$
1		T E'\$	$\uparrow x - 2 \times y$
5		F T' E'\$	$\uparrow x - 2 \times y$
11		id T' E'\$	$\uparrow x - 2 \times y$
→	id	T' E'\$	$x \uparrow - 2 \times y$
8	id	E'\$	$x \uparrow - 2 \times y$
3	id	- T E'\$	$x \uparrow - 2 \times y$
→	id -	T E'\$	$x - \uparrow 2 \times y$
5	id -	F T' E'\$	$x - \uparrow 2 \times y$
10	id -	num T' E'\$	$x - \uparrow 2 \times y$
→	id - num	T' E'\$	$x - 2 \uparrow \times y$
6	id - num	\times F T' E'\$	$x - 2 \uparrow \times y$
→	id - num \times	F T' E'\$	$x - 2 \times \uparrow y$
11	id - num \times	id T' E'\$	$x - 2 \times \uparrow y$
→	id - num \times id	T' E'\$	$x - 2 \times y \uparrow$
8	id - num \times id	E'\$	$x - 2 \times y \uparrow$
4	id - num \times id	\$	$x - 2 \times y \uparrow$

0 $S \rightarrow E \$$

1 $E \rightarrow T E'$

2 $E' \rightarrow + T E'$

3 $E' \rightarrow - T E'$

4 $E' \rightarrow \epsilon$

5 $T \rightarrow F T'$

6 $T' \rightarrow * F T'$

7 $T' \rightarrow / F T'$

8 $T' \rightarrow \epsilon$

9 $F \rightarrow (E)$

10 $F \rightarrow num$

11 $F \rightarrow id$

بازیابی خطا

0 $S \rightarrow E \$$

1 $E \rightarrow T E'$

2 $E' \rightarrow + T E'$

3 $E' \rightarrow - T E'$

4 $E' \rightarrow \epsilon$

5 $T \rightarrow F T'$

6 $T' \rightarrow * F T'$

7 $T' \rightarrow / F T'$

8 $T' \rightarrow \epsilon$

9 $F \rightarrow (E)$

10 $F \rightarrow num$

11 $F \rightarrow id$

```
void T(void) {switch (tok) {
    case ID:
    case NUM:
    case LPAREN: F(); Tprime(); break;
    default: error;
}}
void Tprime(void) {switch (tok) {
    case PLUS: break;
    case MINUS: break;
    case TIMES: eat(TIMES); F(); Tprime(); break;
    case DIVIDES: eat(DIVIDES); F(); Tprime(); break;
    case EOF: break;
    case LPAREN: break;
    default: error;
}}
```

بازیابی خطا

مدخل خالی در ردیف T و ستون x تجزیه چچ (۱)

- به معنای اینکه تابع تجزیه $T()$ منتظر دیدن x نیست
- خطا در صورت وقوع

چگونگی پرداختن به خطا؟

- ایمن ترین راه پیغام استثناء و خروج از تجزیه
- کاربرپسند نیست

▪ ایجاد پیام خطا و بازیابی خطا

- ممکن شدن یافتن دیگر خطاهای نحوی در همان کامپایل

جمله نبودن رشته ورودی دلیل رخداد خطا

بازیابی خطا

- روشی جهت یافتن جمله‌ای شبیه به رشته تکه‌ها

بازیابی خطا - ادامه

انواع بازیابی خطا

- حذف تکه [ها]
- جانشینی تکه [ها]
- درج تکه [ها]

مثال - بازیابی خطای T

▪ درج تکه `num`

▪ لزوماً مطابق با ورودی نیست

▪ کافی است فرض شود ورودی `num` بوده و پیام هشدار به کاربر و بازگشت نرمال

```
void T(void) {switch (tok) {
    case ID:
    case NUM:
    case LPAREN: F(); Tprime(); break;
    default: printf("expected id, num, or left-paren");
}}
```

بازیابی خطا - ادامه

بازیابی خطا با استفاده از درج

- خطرناک
- پیداری از خطاها به دلیل ایجاد خطای دیگر منجر به حلقه بی‌نهایت

```
int Tprime_follow [] = {PLUS, TIMES, RPAREN, EOF, -1};  
  
void Tprime(void) { switch (tok) {  
    case PLUS:    break;  
    case TIMES:  eat(TIMES); F(); Tprime(); break;  
    case RPAREN: break;  
    case EOF:    break;  
    default:    printf("expected +, *, right-paren,  
                    or end-of-file");  
                skipto(Tprime_follow);  
}}}
```

بازیابی خطا با استفاده از حذف

- نادیده گرفتن تکه تا رسیدن تکه‌ای در مجموعه پیرو
- ایمن‌تر به دلیل به انتها رسیدن حلقه

مثال - بازیابی خطای T'

بازیابی خطا - ادامه

سخن کوتاه

- نیاز به تطبیق سازوکار بازیابی خطا جهت جلوگیری از رشته مداومی از پیام‌های تصحیح خطا ناشی از نبود یک تکه در سرچایش
- گاهی اوقات با آزمایش و خطا

تجزیه بالا به پایین - روش نزول بازگشتی

```
token ← next_token
root ← start_symbol
node ← root

loop forever
  if node ∈ T & node ≅ token then
    advance node to next node on the fringe
    token ← next_token
  else if node ∈ T & node ≇ token then
    backtrack
  else if node ∈ NT then
    pick a rule "node → β"
    extend tree from node by building β
    node ← leftmost symbol in β

if node is empty & token = EOF then
  accept
else if node is empty & token ≠ EOF then
  backtrack
```

عدم انطباق

▪ دو دلیل

انتخاب قانون تولید اشتباه هنگام بسط درخت

▪ پس رو

رشته منطبق بر قواعد دستوری نباشد

▪ رشته نامعتبر

▪ شکست پس روی

▪ اعلام خطای نحوی به کاربر

نیاز به وجود تفاوت بین این دو عدم انطباق

تجزیه بالا به پائین تب پ

شروع از ریشه و

گسترش روش مند به سمت پائین

تا رسیدن به برگها

گسترش از متغیرها

عدم توانایی ادامه هنگام رسیدن به حروف الفباء

ادامه تا هنگامی که تمامی علامتها حروف الفبا باشند

▪ متغیری باقی نمانده باشد

تجزیه بالا به پائین - روش نزول بازگشتی

دارای رویه به ازای هر متغیر

شروع عملیات از متغیر آغاز

توقف و ختم در صورتی که رویه کل رشته ورودی را پوشش دهد

منابع

[بیرسبز]

[اژدرها]

[کوپر]

[فیشر]