دانشگاه صنعتی شاهرود
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

درس اصول طراحی کامپایلر

تجزیه بالا به‌پایین

مدرس:
علیرضا تجری

تجزیه بالابه‌پایین با کمک تابع First و Follow

- رشته id*(id+id) را تجزیه کنید.

$$E \longrightarrow T\ E'$$
$$E' \longrightarrow +\ T\ E' \mid \lambda$$
$$T \longrightarrow F\ T'$$
$$T' \longrightarrow *\ F\ T' \mid \lambda$$
$$F \longrightarrow (E) \mid id$$

First(F)={(, id}
First(T)={(, id}
First(T')={*, λ}
First(E)={(, id}
First(E')={+, λ}
First((E))={(}
First(id)={id}
First(*FT)={*}
First(λ)={λ}
First(FT')={(, id}
First(+TE')={+}
First(TE')={(, id}
Follow(F)={+, *,), $}
Follow(T)={+,), $}
Follow(T')={+,), $}
Follow(E)={), $}
Follow(E')={), $}

تجزیه بالابه‌پایین با کمک تابع First و Follow

رشته id*(id+id) را تجزیه کنید.

id*(id+id)$

E ⟶ T E’
E’⟶ + T E’ | λ
T⟶ F T’
T’⟶ * F T’ |λ
F⟶ (E) | **id**

First(F)={(, id}
First(T)={(, id}
First(T’)={*, λ}
First(E)={(, id}
First(E’)={+, λ}
First((**E**))={(}
First(id)={id}
First(*FT)={*}
First(λ)={λ}
First(FT’)={(, id}
First(+TE’)={+}
First(TE’)={(, id}
Follow(F)={+, *,), $}
Follow(T)={+,), $}
Follow(T’)={+,), $}
Follow(E)={), $}
Follow(E’)={), $}

E⇒
TE’⇒
FT’E’⇒
idT’E’⇒
id*FT’E’⇒
id*(E)T’E’⇒
id*(TE’)T’E’⇒
id*(FT’E’)T’E’⇒
id*(idT’E’)T’E’⇒
id*(idE’)T’E’⇒

id*(id+TE’)T’E’⇒
id*(id+FT’E’)T’E’⇒
id*(id+idT’E’)T’E’⇒
id*(id+idE’)T’E’⇒
id*(id+id)T’E’⇒
id*(id+id)E’⇒
id*(id+id)

تجزیه بالابه‌پایین با کمک تابع First و Follow

- فرض کنید
  - غیرپایانه A، سمت چپ‌ترین غیرپایانه در بسط باشد.
  - در گرامر، برای غیرپایانه A، چندین قاعده وجود دارد.
  - سمبل بعدی ورودی b باشد.
- در چه صورت از قاعده $A \longrightarrow \alpha$ استفاده می‌کنیم؟
  - اگر $b \in first(\alpha)$
  - اگر $b \in follow(A)$ و $\lambda \in first(\alpha)$

می‌توان یک جدول ایجاد کرد.

برچسب سطرهای جدول: غیرپایانه‌ها

برچسب ستون‌های جدول: پایانه‌ها و $

محتوای خانه‌های جدول: قاعده گرامری قابل استفاده

جدول پارسر پیشگو

جدول پارسر پیشگو

- برچسب سطرها: غیرپایانه‌ها
- برچسب ستون‌ها: پایانه‌ها و $
  - سمبل‌های ورودی
- محتوای خانه‌ها: قاعده گرامری قابل استفاده
  - سمت‌چپ‌ترین غیرپایانه: برچسب سطر
  - سمبل ورودی: برچسب ستون

NON-TERMINAL	INPUT SYMBOL					
	id	+	*	(	)	$
E	$E \to TE'$			$E \to TE'$		
E'		$E' \to +TE'$			$E' \to \epsilon$	$E' \to \epsilon$
T	$T \to FT'$			$T \to FT'$		
T'		$T' \to \epsilon$	$T' \to *FT'$		$T' \to \epsilon$	$T' \to \epsilon$
F	$F \to \mathbf{id}$			$F \to (E)$		

نحوه ساخت جدول پارسر پیشگو

- جدول تجزیه خالی را ایجاد کنید.
- برای هر قاعده به شکل $A \longrightarrow \alpha$ کارهای زیر را انجام دهید:

1) برای هر پایانه b در $first(\alpha)$ قاعده $A \longrightarrow \alpha$ را به $M[A, b]$ اضافه کنید.

2) اگر $\lambda \in first(\alpha)$، برای هر سمبل b در $Follow(A)$، قاعده $A \longrightarrow \alpha$ را به $M[A, b]$ اضافه کنید.

- اگر پس از انجام اعمال بالا، خانه‌ای از جدول خالی بود، در آن خانه Error قرار دهید.

$E \longrightarrow T\ E'$
$E' \longrightarrow +\ T\ E' \mid \lambda$
$T \longrightarrow F\ T'$
$T' \longrightarrow *\ F\ T' \mid \lambda$
$F \longrightarrow (E) \mid id$

	id	+	*	(	)	$
E						
E '						
T						
T '						
F						

نمونه جدول پارسر پیشگو

E ⟶ T E’
E’⟶ + T E’ | λ
T⟶ F T’
T’⟶ * F T’ | λ
F⟶ (E) | **id**

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	(	)	$
E	$E \to TE'$			$E \to TE'$		
E'		$E' \to +TE'$			$E' \to \epsilon$	$E' \to \epsilon$
T	$T \to FT'$			$T \to FT'$		
T'		$T' \to \epsilon$	$T' \to *FT'$		$T' \to \epsilon$	$T' \to \epsilon$
F	$F \to$ **id**			$F \to (E)$		

ساخت جدول پارسر پیشگو برای یک گرامر غیر LL(1)

$S \longrightarrow iEtSS' \mid a$

$S' \longrightarrow eS \mid \lambda$

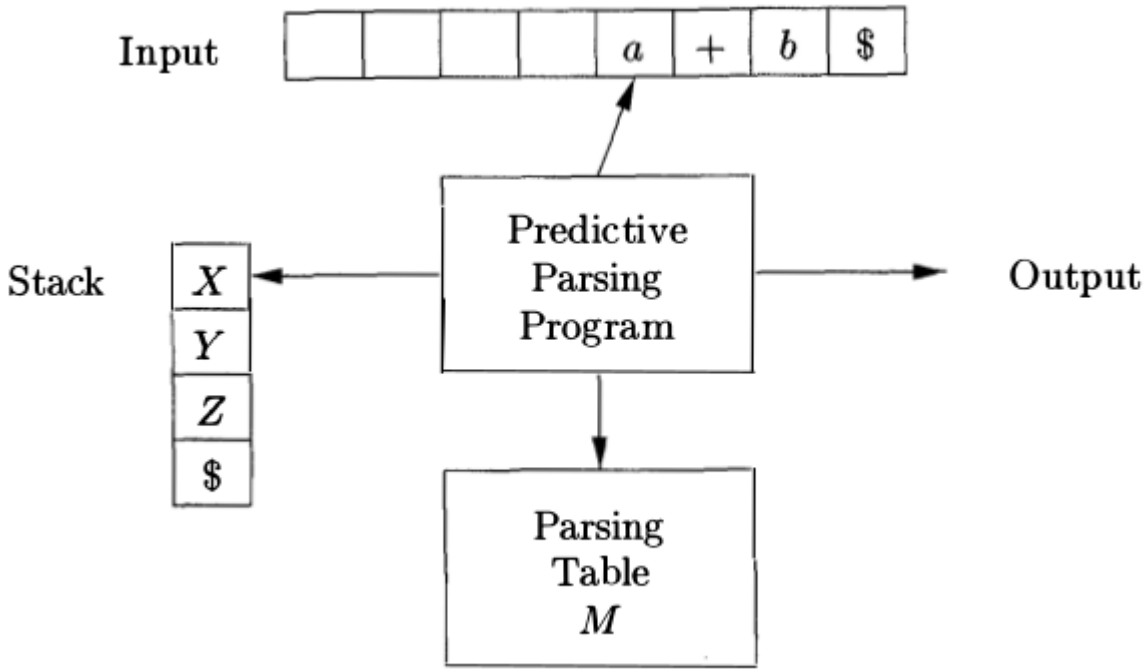
$E \longrightarrow b$

در یک خانه جدول، بیش از یک قاعده گرامری قرار می‌گیرد.

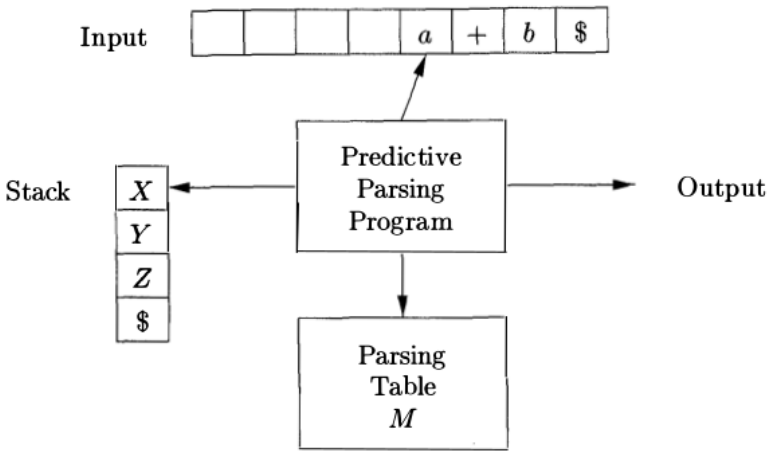
رفع ابهام با اولویت دادن به یک قاعده گرامری

NON - TERMINAL	INPUT SYMBOL					
	a	b	e	i	t	$
S	$S \to a$			$S \to iEtSS'$		
S'			$S' \to \epsilon$ $S' \to eS$			$S' \to \epsilon$
E		$E \to b$				

پارسر پیشگوی غیربازگشتی



پارسر پیشگوی غیربازگشتی



- ورودی: دنباله توکن‌های رشته ورودی
  - در انتهای ورودی، سمبل $ قرار دارد.
- پشته: یک پشته معمولی
  - در ابتدا $ به پشته اضافه می‌شود.
  - سپس S (غیر پایانه شروع) به پشته اضافه می‌شود.
- جدول تجزیه: جدول پارسر پیشگو
- برنامه تجزیه: الگوریتمی که با توجه به **سمبل بالای پشته** و **توکن ورودی** و **جدول تجزیه**، **خروجی** را تعیین می‌کند.
- خروجی: دنباله قواعد استفاده شده برای تجزیه رشته ورودی

برنامه تجزیه

a: the input symbol that *ip* points to it

set *ip* to point to the first symbol of w;
set X to the top stack symbol;
while ($X \neq \$$) { /* stack is not empty */
    if (X is a) pop the stack and advance *ip*;
    else if (X is a terminal) *error*();
    else if ($M[X, a]$ is an error entry) *error*();
    else if ($M[X, a] = X \rightarrow Y_1 Y_2 \cdots Y_k$) {
        output the production $X \rightarrow Y_1 Y_2 \cdots Y_k$;
        pop the stack;
        push $Y_k, Y_{k-1}, \ldots, Y_1$ onto the stack, with Y_1 on top;
    }
    set X to the top stack symbol;
}

نمونه تجزیه

MATCHED	STACK	INPUT	ACTION

نمونه تجزیه

MATCHED	STACK	INPUT	ACTION
	$E\$$	$\mathbf{id} + \mathbf{id} * \mathbf{id}\$$	
	$TE'\$$	$\mathbf{id} + \mathbf{id} * \mathbf{id}\$$	output $E \to TE'$
	$FT'E'\$$	$\mathbf{id} + \mathbf{id} * \mathbf{id}\$$	output $T \to FT'$
	$\mathbf{id}\ T'E'\$$	$\mathbf{id} + \mathbf{id} * \mathbf{id}\$$	output $F \to \mathbf{id}$
$\mathbf{id}$	$T'E'\$$	$+\,\mathbf{id} * \mathbf{id}\$$	match **id**
$\mathbf{id}$	$E'\$$	$+\,\mathbf{id} * \mathbf{id}\$$	output $T' \to \epsilon$
$\mathbf{id}$	$+\ TE'\$$	$+\,\mathbf{id} * \mathbf{id}\$$	output $E' \to +\ TE'$
$\mathbf{id} +$	$TE'\$$	$\mathbf{id} * \mathbf{id}\$$	match $+$
$\mathbf{id} +$	$FT'E'\$$	$\mathbf{id} * \mathbf{id}\$$	output $T \to FT'$
$\mathbf{id} +$	$\mathbf{id}\ T'E'\$$	$\mathbf{id} * \mathbf{id}\$$	output $F \to \mathbf{id}$
$\mathbf{id} + \mathbf{id}$	$T'E'\$$	$* \mathbf{id}\$$	match **id**
$\mathbf{id} + \mathbf{id}$	$*\ FT'E'\$$	$* \mathbf{id}\$$	output $T' \to *\ FT'$
$\mathbf{id} + \mathbf{id} *$	$FT'E'\$$	$\mathbf{id}\$$	match $*$
$\mathbf{id} + \mathbf{id} *$	$\mathbf{id}\ T'E'\$$	$\mathbf{id}\$$	output $F \to \mathbf{id}$
$\mathbf{id} + \mathbf{id} * \mathbf{id}$	$T'E'\$$	$\$$	match **id**
$\mathbf{id} + \mathbf{id} * \mathbf{id}$	$E'\$$	$\$$	output $T' \to \epsilon$
$\mathbf{id} + \mathbf{id} * \mathbf{id}$	$\$$	$\$$	output $E' \to \epsilon$

رسم درخت تجزیه

رسم قاعده‌های گرامری

به ترتیب

بسط سمت‌چپ‌ترین

ACTION
output $E \rightarrow TE'$
output $T \rightarrow FT'$
output $F \rightarrow \mathbf{id}$
match **id**
output $T' \rightarrow \epsilon$
output $E' \rightarrow + TE'$
match $+$
output $T \rightarrow FT'$
output $F \rightarrow \mathbf{id}$
match **id**
output $T' \rightarrow * FT'$
match $*$
output $F \rightarrow \mathbf{id}$
match **id**
output $T' \rightarrow \epsilon$
output $E' \rightarrow \epsilon$