

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



اصول طراحی کامپایلر

درس ۱۰

تحلیل نحوی (۵)

تجزیه‌ی پایین به بالا : روش تقدم-عملگر

Syntax Analysis (5)

Bottom-Up Parsing – Operator-Precedence Method

کاظم فولادی قلعه

دانشکده مهندسی، پردیس فارابی

دانشگاه تهران

<http://courses.fouladi.ir/compiler>

تحلیل نحوی
تجزیه‌ی پایین به بالا
روش تقدم-عملگر

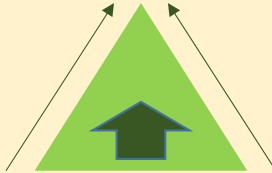


مقدمه:
روش‌های
تقدم

تجزیه‌ی پایین به بالا

روش‌های تجزیه
*Parsing Methods*روش‌های پایین به بالا
Bottom-Up Parsing Methods

ساخت درخت تجزیه از پایین به بالا
(از برگ‌ها به سمت ریشه)

روش‌های بالا به پایین
Top-Down Parsing Methods

ساخت درخت تجزیه از بالا به پایین
(از ریشه به سمت برگ‌ها)



از اشتقاق‌های راست‌ترین در جهت معکوس استفاده می‌شود.

روش‌های LR
*LR Methods*روش‌های تقدم
Precedence Methods

SLR

LALR

CLR

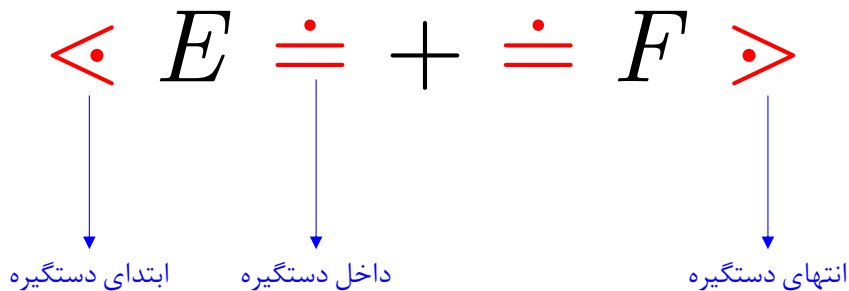
روش تقدم عملگر

روش تقدم ساده

روش‌های تقدم

در تجزیه‌ی پایین به بالا

اساس این روش‌ها،
استفاده از نمادهایی با نام «نمادهای تقدم»
برای مشخص کردن مرز دستگیره‌ها در هر فرم جمله‌ای است.



«نمادهای تقدم» در روشهای تقدم

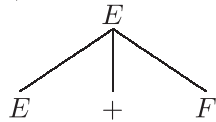
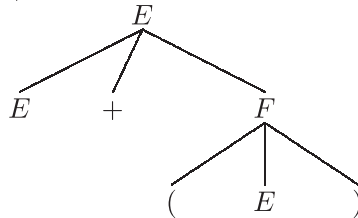
مثال (۱ از ۴)

 E $\langle E \rangle$

گرامر

$$E \rightarrow E + F \mid F$$

$$F \rightarrow (E) \mid \text{id}$$

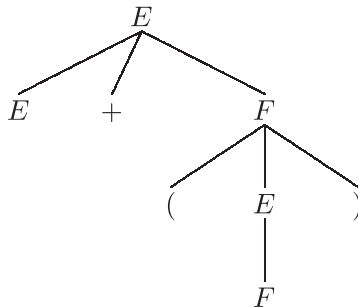
 $\Downarrow$  $\langle E \dot{=} + \dot{=} F \rangle$ $\Downarrow$  $\langle E \dot{=} + \langle \dot{=} E \dot{=} \rangle \rangle$ $\Downarrow$

رشته

 $\text{id} + (\text{id})$

«نمادهای تقدم» در روشهای تقدم

مثال (۲ از ۴)

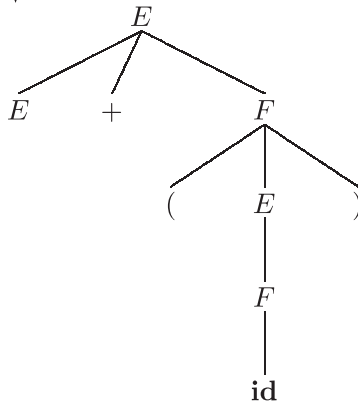


$$\langle E \dot{=} + \langle (\langle F \rangle) \rangle \rangle$$

گرامر

$$E \rightarrow E + F \mid F$$

$$F \rightarrow (E) \mid \text{id}$$

$$\Downarrow$$


$$\langle E \dot{=} + \langle (\langle \langle \text{id} \rangle \rangle) \rangle \rangle$$

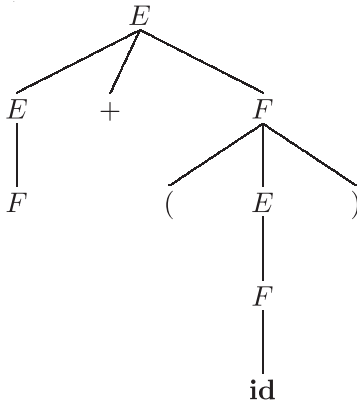
رشته

$$\text{id} + (\text{id})$$

$$\Downarrow$$

«نمادهای تقدم» در روشهای تقدم

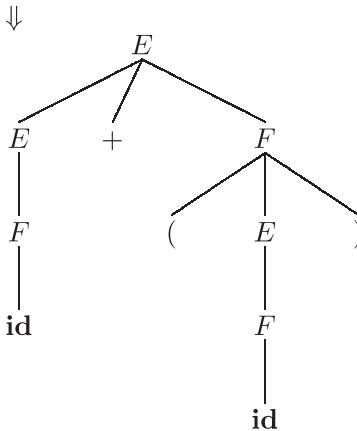
مثال (۳ از ۴)


 $\langle \langle F \rangle + \langle (\langle \langle id \rangle \rangle) \rangle \rangle$

گرامر

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

رشته

 $id + (id)$

 $\langle \langle \langle id \rangle \rangle + \langle (\langle \langle id \rangle \rangle) \rangle \rangle$

«نمادهای تقدم» در روش‌های تقدم

مثال (۴ از ۴)

$$E \rightarrow E + F \mid F$$

$$F \rightarrow (E) \mid \text{id}$$

$$E \Rightarrow \underline{E + F} \Rightarrow E + \underline{(E)} \Rightarrow E + \underline{(F)} \Rightarrow E + \underline{(\text{id})} \Rightarrow \underline{F} + \underline{(\text{id})} \Rightarrow \underline{\text{id}} + \underline{(\text{id})}$$

فرم جمله‌ای جاری
در اشتقاق راست‌تریننمایش درختی فرم جمله‌ای با
نمادهای تقدمتوکن‌های خوانده شده از
چپ به راستدستگیره‌ی جاری
(بالای پشته)

$\Rightarrow \underline{\text{id}} + (\text{id})$	$\langle \langle \langle \underline{\text{id}} \rangle \rangle + \langle (\langle \langle \text{id} \rangle \rangle) \rangle \rangle$	id	$\langle \langle \langle \underline{\text{id}} \rangle \rangle$
$\Rightarrow \underline{F} + (\text{id})$	$\langle \langle \underline{F} \rangle + \langle (\langle \langle \text{id} \rangle \rangle) \rangle \rangle$	id	$\langle \langle \underline{F} \rangle$
$\Rightarrow E + (\underline{\text{id}})$	$\langle E \dot{=} + \langle (\langle \langle \underline{\text{id}} \rangle \rangle) \rangle \rangle$	$\text{id} + (\text{id})$	$\langle E \dot{=} + \langle (\langle \underline{\text{id}} \rangle \rangle$
$\Rightarrow E + (\underline{F})$	$\langle E \dot{=} + \langle (\langle \underline{F} \rangle) \rangle \rangle$	$\text{id} + (\text{id})$	$\langle E \dot{=} + \langle (\langle \underline{F} \rangle$
$\Rightarrow E + (\underline{E})$	$\langle E \dot{=} + \langle (\underline{\dot{=} E \dot{=}}) \rangle \rangle$	$\text{id} + (\text{id})$	$\langle E \dot{=} + \langle (\underline{\dot{=} E \dot{=}})$
$\Rightarrow \underline{E + F}$	$\langle \underline{E \dot{=} + \dot{=} F} \rangle$	$\text{id} + (\text{id})$	$\langle \underline{E \dot{=} + \dot{=} F}$
E	$\langle E \rangle$	$\text{id} + (\text{id})$	$\langle E$

با پوشش رشته از چپ به راست، دستگیره همیشه سمت چپ‌ترین عبارت بین $\langle$ و $\rangle$ است (و درون آن دیگر این نمادها وجود ندارد).

تحلیل نحوی
تجزیه‌ی پایین به بالا
روش تقدم-عملگر

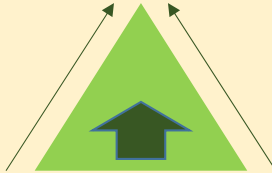
۲

روش
تقدم-عملگر

تجزیه‌ی پایین به بالا

روش‌های تجزیه
*Parsing Methods*روش‌های پایین به بالا
Bottom-Up Parsing Methods

ساخت درخت تجزیه از پایین به بالا
(از برگ‌ها به سمت ریشه)

روش‌های بالا به پایین
Top-Down Parsing Methods

ساخت درخت تجزیه از بالا به پایین
(از ریشه به سمت برگ‌ها)



از اشتقاق‌های راست‌ترین در جهت معکوس استفاده می‌شود.

روش‌های LR
*LR Methods*روش‌های تقدم
Precedence Methods

SLR LALR CLR

روش تقدم عملگر

روش تقدم ساده

گرامر عملگر

گرامر عملگر

Operator Grammar

گرامری که

(۱) قاعده‌ی تهی (ϵ) ندارد.

(۲) در سمت راست هیچ یک از قواعد آن دو ناپایانه‌ی مجاور وجود ندارد.

شرط لازم برای استفاده از روش تجزیه‌ی تقدم-عملگر این است که گرامر مورد استفاده، یک گرامر عملگر باشد.

گرامر عملگر

مثال

گرامر زیر یک گرامر عملگر نیست:

$$E \rightarrow EAE \mid (E) \mid \text{id}$$

$$A \rightarrow + \mid - \mid * \mid /$$

اما گرامر زیر یک گرامر عملگر معادل با گرامر فوق است:

$$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid \text{id}$$

روابط تقدم میان پایانه‌ها

چهار رابطه‌ی تقدم ممکن میان هر دو پایانه‌ی گرامر (a و b):

a مقدم بر b	$a \geq b$	(a باید قبل از b کاهش یابد.)
a مؤخر از b	$a \leq b$	(a باید بعد از b کاهش یابد.)
a هم‌تقدم با b	$a \doteq b$	(a و b هر دو در یک دستگیره قرار دارند.)
a بدون رابطه با b	$a \square b$	

- این روابط متقارن نیستند.
- ممکن است بین دو پایانه هیچ رابطه‌ی تقدمی موجود نباشد.
- ممکن است بین دو پایانه چند رابطه‌ی تقدم موجود باشد.

روابط تقدم میان پایانه‌ها

توابع لازم برای یافتن قواعد تقدم

نخستین پایانه‌ی فرم‌های جمله‌ای حاصل از ناپایانه‌ی A $FirstTerm(A)$

$$FirstTerm(A) = \{a : A \Rightarrow^+ a\alpha \vee A \Rightarrow^+ Ba\alpha\}$$

آخرین پایانه‌ی فرم‌های جمله‌ای حاصل از ناپایانه‌ی A $LastTerm(A)$

$$LastTerm(A) = \{a : A \Rightarrow^+ \alpha a \vee A \Rightarrow^+ \alpha aB\}$$

روابط تقدم میان پایانه‌ها

توابع لازم برای یافتن قواعد تقدم: مثال

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

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

$$F \rightarrow (E) \mid \mathbf{id}$$

$$FirstTerm(E) = \{+, *, (, \mathbf{id}\} \quad LastTerm(E) = \{+, *,), \mathbf{id}\}$$

$$FirstTerm(T) = \{*, (, \mathbf{id}\} \quad LastTerm(T) = \{*,), \mathbf{id}\}$$

$$FirstTerm(F) = \{(, \mathbf{id}\} \quad LastTerm(F) = \{), \mathbf{id}\}$$

روابط تقدم میان پایانه‌ها

قواعد یافتن روابط تقدم

با توجه به قواعد گرامر

بین a و b حداکثر یک ناپایانه باشد	
$a \doteq b$	$\Leftrightarrow \exists U(U \rightarrow \dots ab \dots \vee U \rightarrow \dots aWb \dots)$
$a \leq FirstTerm(W)$	a قبل از یک ناپایانه W
$a \leq b$	$\Leftrightarrow \exists U(U \rightarrow aW), b \in FirstTerm(W)$
$LastTerm(W) \geq b$	b بعد از یک ناپایانه W
$a \geq b$	$\Leftrightarrow \exists U(U \rightarrow Wb), a \in LastTerm(W)$
عدم وجود رابطه‌ی تقدم بین a و b	
$a \square b$	$\Leftrightarrow \neg(a \leq b \vee a \doteq b \vee a \geq b)$

تعیین رابطه‌ی $\$$ با سایر پایانه‌ها: با استفاده از قاعده‌ی افزوده $S' \rightarrow \$S\$$

روابط تقدم میان پایانه‌ها

یافتن روابط تقدم: مثال

$$E' \rightarrow \$E\$$$

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

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

$$F \rightarrow (E) \mid \text{id}$$

$$\$ \doteq \$$$

$$(\doteq)$$

$$\$ \triangleleft FirstTerm(E)$$

$$+ \triangleleft FirstTerm(T)$$

$$* \triangleleft FirstTerm(F)$$

$$(\triangleleft FirstTerm(E)$$

$$LastTerm(E) \triangleright \$$$

$$LastTerm(E) \triangleright +$$

$$LastTerm(T) \triangleright *$$

$$LastTerm(E) \triangleright)$$

جدول تجزیه‌ی تقدم-عملگر

جدول تجزیه‌ی تقدم-عملگر

یک جدول مربعی با ابعاد

$$(|T| + 1)(|T| + 1)$$

شامل همه‌ی روابط تقدم میان پایانه‌ها

$$P : (T \cup \{\$, \}\}) \times (T \cup \{\$, \}\}) \rightarrow \{<, \doteq, >, \square\}$$

جدول تجزیه‌ی تقدم-عملگر

مثال

$$E' \rightarrow \$E\$$$

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

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

$$F \rightarrow (E) \mid \text{id}$$

	+	*	(	)	id	\$
+	$\triangleright$	$\triangleleft$	$\triangleleft$	$\triangleright$	$\triangleleft$	$\triangleright$
*	$\triangleright$	$\triangleright$	$\triangleleft$	$\triangleright$	$\triangleleft$	$\triangleright$
(	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\doteq$	$\triangleleft$	
)	$\triangleright$	$\triangleright$		$\triangleright$		$\triangleright$
id	$\triangleright$	$\triangleright$		$\triangleright$		$\triangleright$
\$	$\triangleleft$	$\triangleleft$	$\triangleleft$		$\triangleleft$	$\doteq$

OPERATOR-PRECEDENCE PARSING PROCEDURE

push(\$)

repeat

$a \leftarrow \text{top-terminal}(\text{Stack})$

$b \leftarrow \text{lookahead}$

if $P[a, b] = \lessdot$ **then** *push*($\lessdot$); *shift*(b)

if $P[a, b] = \doteq$ **then** *push*($\doteq$); *shift*(b)

if $P[a, b] = \gtrdot$ **then** *reduce*()

if $P[a, b] = \square$ **then** *error*()

if $a = b = \$$ **then** *accept*()

روال تجزیه‌ی تقدم-عملگر

«عملیات کاهش» در روال تجزیه‌ی تقدم-عملگر

OPERATOR-PRECEDENCE PARSING PROCEDURE: REDUCE OPERATION

پایین رفتن در پشته تا رسیدن به اولین نماد $<$
 دستگیره = رشته‌ی میان $<$ و بالای پشته (و ناپایانه‌ی زیر $<$ در صورت وجود)

۱
یافتن دستگیره
 از بالای پشته

۲
حذف دستگیره
 از بالای پشته

۳
درج ناپایانه‌ی نوعی N
 در بالای پشته

کاهش
Reduce

روال تجزیه‌ی تقدم-عملگر

مثال

STACK	INPUT	ACTION
\$	id + id * id \$	shift
\$ <u><</u> id	+ id * id \$	reduce $F \rightarrow \mathbf{id}$
\$N	+ id * id \$	shift
\$N < +	id * id \$	shift
\$N < + < id	* id \$	reduce $F \rightarrow \mathbf{id}$
\$N < + N	* id \$	shift
\$N < + N < *	id \$	shift
\$N < + N < * < id	\$	reduce $F \rightarrow \mathbf{id}$
\$N < + <u>N < * N</u>	\$	reduce $T \rightarrow T * F$
\$ <u>N < + N</u>	\$	reduce $E \rightarrow E + T$
\$N	\$	accept

$E' \rightarrow \$E\$$
 $E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid \mathbf{id}$

	+	*	(	)	id	\$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(	<	<	<	=	<	
)	>	>		>		>
id	>	>		>		>
\$	<	<	<		<	=

وقوع خطا در تجزیه‌ی تقدم-عملگر

وقوع خطا در تجزیه‌ی تقدم-عملگر

عدم وجود رابطه میان a (پایانه‌ی بالای پشته) و b (توکن جاری)

۱

عدم تطبیق دستگیره‌ی واقع در بالای پشته با سمت راست هیچ یک از قواعد

۲

روش تجزیه‌ی تقدم-عملگر

مزایا و معایب

مزایا و معایب روش تجزیه‌ی تقدم-عملگر

معایب	مزایا
محدودیت زیاد	پیاده‌سازی ساده
غیر قابل استفاده برای گرامرهای شامل عملگرهایی یکسان با دو تقدم مختلف (مثل منهای دوتایی و تکی)	
امکان پیدا نشدن برخی از خطاهای نحوی	
امکان عدم تساوی زبان تجزیه‌گر با زبان گرامر	

گرامر تقدم-عملگر

گرامر تقدم-عملگر

Operator-Precedence Grammar

گرامری که جدول تجزیه‌ی تقدم-عملگر آن تداخلی نداشته باشد.

توابع تقدم

توابع تقدم، دو تابع f و g به صورت

$$f, g : T \cup \{\$ \} \rightarrow \{0, 1, 2, \dots\}$$

هستند که برای آنها داشته باشیم:

$f(a) > g(b)$	اگر و فقط اگر	$a > b$
$f(a) = g(b)$	اگر و فقط اگر	$a \doteq b$
$f(a) < g(b)$	اگر و فقط اگر	$a \leq b$

اشکال اساسی وارد به توابع تقدم:

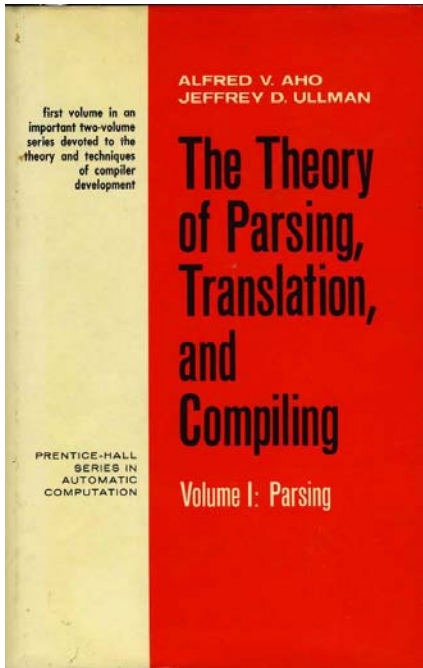
اگر بین دو پایانه، رابطه‌ی تقدم موجود نباشد، تشخیص آن ممکن نیست.

تحلیل نحوی
تجزیه‌ی پایین به بالا
روش تقدم-عملگر

۳

منابع

منبع اصلی



A. V. Aho, J. D. Ullman,
The Theory of Parsing, Translation, and Compiling,
Vol. I, Prentice-Hall, 1972.

Chapter 5 (5.4.3)