اصول طراحی کامپایلر

درس ۱۲

تحلیل نحوی (۷)

تجزیه‌ی پایین به بالا : روش SLR

Syntax Analysis (7)

Bottom-Up Parsing – Simple-LR (SLR) Method

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

http://courses.fouladi.ir/compiler

اصول طراحی کامپایلر

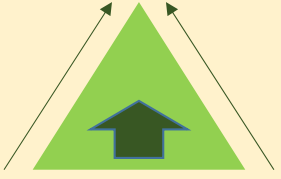
تحلیل نحوی
تجزیه‌ی پایین به بالا
روش SLR

۱

مقدمه

تجزیه‌ی پایین به بالا

روش‌های تجزیه
Parsing Methods

روش‌های بالا به پایین *Top-Down Parsing Methods*	**روش‌های پایین به بالا** *Bottom-Up Parsing Methods*
ساخت درخت تجزیه از بالا به پایین (از ریشه به سمت برگ‌ها)	ساخت درخت تجزیه از پایین به بالا (از برگ‌ها به سمت ریشه)
	
	از **اشتقاق‌های راست‌ترین** در جهت معکوس استفاده می‌شود.
روش‌های تقدم *Precedence Methods*	**روش‌های LR** *LR Methods*
روش تقدم عملگر روش تقدم ساده	SLR | LALR | CLR

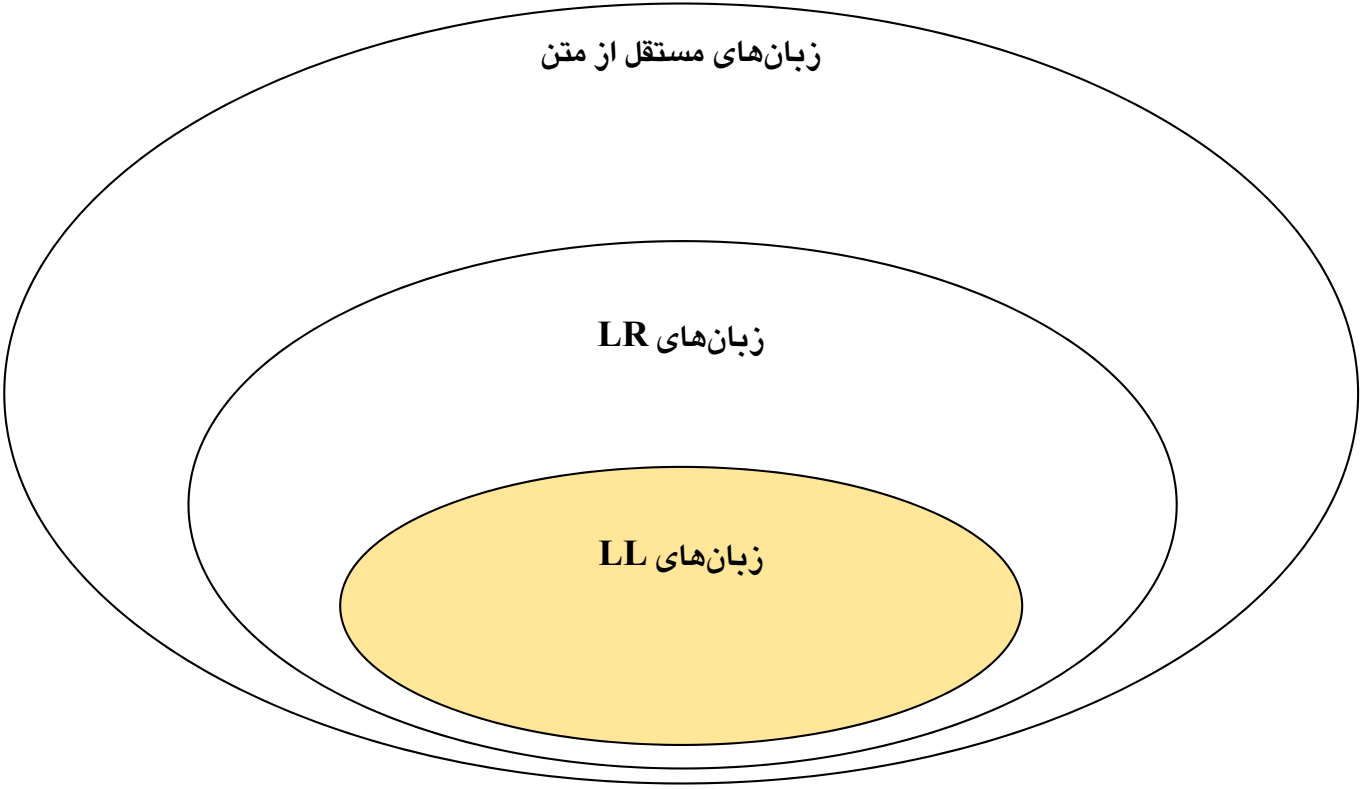
گرامر LR(k)

گرامر G یک گرامر LR(k) است
اگر
در هر گام از پویش رشته از سمت **چپ به راست**
تنها با نگاه کردن به k **توکن ورودی بعدی**،
بتوان قاعده‌ی استفاده شده در **اشتقاق راست‌ترین** در آن گام را تعیین کرد.

زبان LR(k) زبانی است که حداقل یک گرامر LR(k) داشته باشد.

گرامرهای LR عمومی‌ترین خانواده از گرامرها هستند که بدون عقب‌گرد قابل تجزیه اند.

نسبت زبان‌های LR با زبان‌های LL



اصول طراحی کامپایلر

تحلیل نحوی
تجزیه‌ی پایین به بالا
روش SLR

۲

روش SLR

آیتم LR(0)

LR(0) ITEM

یک آیتم $\mathrm{LR}(0)$ از گرامر G
یک قاعده‌ی تولید از G است که در یک مکان از سمت راست آن یک نقطه قرار گرفته است.

$$A \longrightarrow \alpha \bullet \beta$$

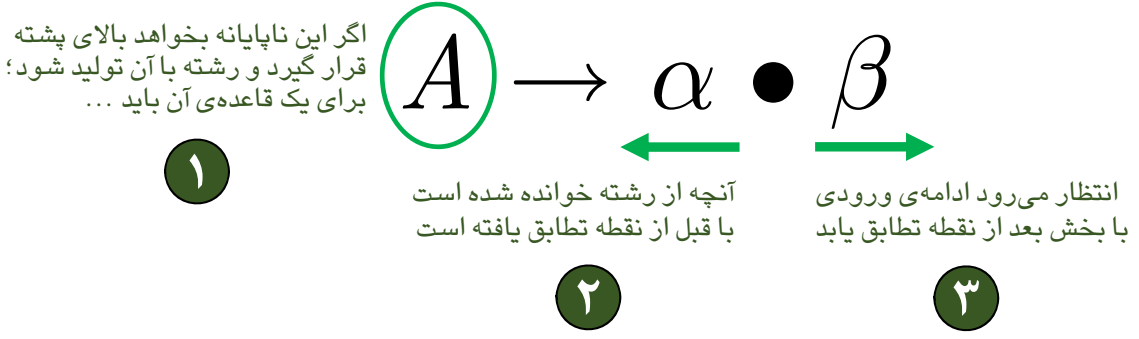
$$\eta = \alpha\beta, \quad A \to \eta \in P(G)$$

آیتم LR(0)

تعبیر

LR(0) ITEM

یک آیتم $\mathrm{LR}(0)$ از گرامر G
یک قاعده‌ی تولید از G است که در یک مکان از سمت راست آن یک نقطه قرار گرفته است.

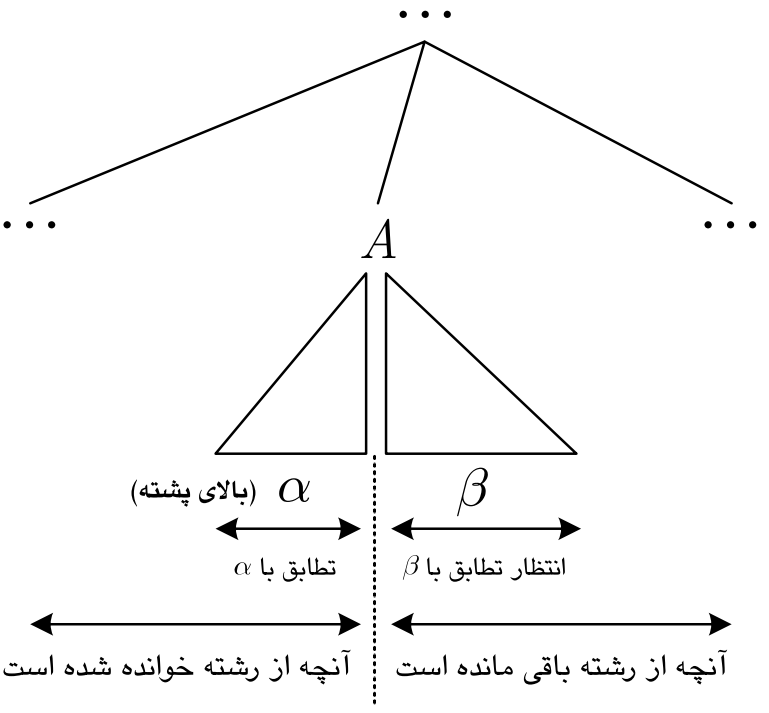


$$\eta = \alpha\beta, \quad A \rightarrow \eta \in P(G)$$

آیتم LR(0)

تعبیر شماتیک

LR(1) ITEM



$$A \rightarrow \alpha \bullet \beta$$

آیتم LR(0)

مثال

LR(0) ITEM

قاعده‌ی تولید

$$A \rightarrow XYZ$$

آیتم‌های LR(0)

$$A \rightarrow \bullet XYZ$$
$$A \rightarrow X \bullet YZ$$
$$A \rightarrow XY \bullet Z$$
$$A \rightarrow XYZ \bullet$$

نکته: قاعده‌ی تولید تهی $A \rightarrow \epsilon$ تنها یک آیتم $\mathrm{LR}(0)$ $A \rightarrow \bullet$ تولید می‌کند.

آتوماتون LR(0)

برای شناسایی دستگیره

LR(0) AUTOMATON

در روش‌های LR دستگیره در هر گام به کمک یک آتوماتون متناهی (DFA) شناسایی می‌شود.

حالت‌های این DFA از **آیتم‌های LR(0)** تشکیل می‌شوند:

آیتم‌هایی که نماینده‌ی وضعیت‌های یکسان هستند، در یک حالت دسته‌بندی می‌شوند.

عملیات بستار

CLOSURE OPERATION

الگوریتم محاسبه‌ی بستار یک مجموعه آیتم: CLOSURE(I)

۱) در ابتدا I به CLOSURE(I) اضافه می‌شود.

۲) **اگر** $A \to \alpha \bullet B\beta \in$ CLOSURE(I) (نقطه قبل از یک ناپایانه‌ی B) و
قاعده‌ی $B \to \gamma$ در G موجود بود،
آن‌گاه آیتم $B \to \bullet\gamma$ به CLOSURE(I) اضافه می‌شود.

۳) قاعده‌ی فوق را آن‌قدر تکرار می‌کنیم تا دیگر چیزی به CLOSURE(I) اضافه نشود.

اگر در یکی از آیتم‌های مجموعه‌ی I نقطه قبل از یک ناپایانه قرار داشت،
کلیه‌ی آیتم‌های حاصل از قواعد آن ناپایانه (با قرار گرفتن نقطه در ابتدای سمت راست آنها)
به CLOSURE(I) اضافه می‌شود.

عملیات بستار

منطق عملیات بستار

CLOSURE OPERATION

اگر در یکی از آیتم‌های مجموعه‌ی I نقطه قبل از یک ناپایانه قرار داشت،
کلیه‌ی آیتم‌های حاصل از قواعد آن ناپایانه (با قرار گرفتن نقطه در ابتدای سمت راست آنها)
به $\text{CLOSURE}(I)$ اضافه می‌شود.

اگر

$$A \rightarrow \alpha \bullet B\beta \in \text{CLOSURE}(I)$$

انتظار داریم که ادامه‌ی ورودی از $B\beta$ قابل اشتقاق باشد.

حال اگر

$$B \rightarrow \gamma$$

در گرامر موجود باشد،

بدیهی است که می‌توان انتظار داشت که ادامه‌ی ورودی از γ قابل اشتقاق باشد.

عملیات بستار

مثال

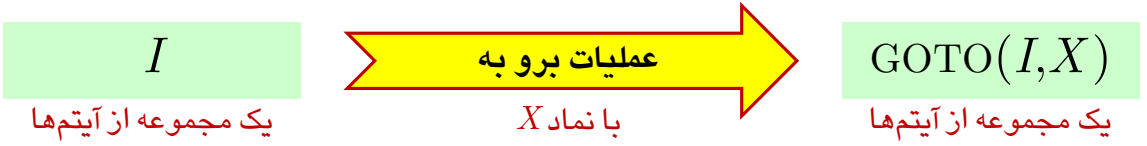
CLOSURE OPERATION

$$
\begin{aligned}
E' &\rightarrow E\$ \\
E &\rightarrow E + T \\
E &\rightarrow T \\
T &\rightarrow T * F \\
T &\rightarrow F \\
F &\rightarrow (E) \\
F &\rightarrow \mathbf{id}
\end{aligned}
$$

$$
\begin{aligned}
\text{CLOSURE}(\{E' \rightarrow \bullet E\$\}) = \{ & E' \rightarrow \bullet E\$, \\
& E \rightarrow \bullet E + T, \\
& E \rightarrow \bullet T, \\
& T \rightarrow \bullet T * F, \\
& T \rightarrow \bullet F, \\
& F \rightarrow \bullet (E), \\
& F \rightarrow \bullet \mathbf{id}, \}
\end{aligned}
$$

عملیات برو به

GOTO OPERATION



الگوریتم محاسبه‌ی برو به برای یک مجموعه آیتم: $\text{GOTO}(I,X)$

اگر I مجموعه‌ای از آیتم‌ها و X یک نماد گرامر باشد،
در این صورت $\text{GOTO}(I, X)$ بستار مجموعه‌ی همه‌ی آیتم‌ها به صورت $A \to \alpha X \bullet \beta$ است که در آن $A \to \alpha \bullet X\beta$ در I قرار دارد.

$$A \to \alpha \bullet X\beta \in I \quad \Rightarrow \quad \text{CLOSURE}(\{A \to \alpha X \bullet \beta\}) \subseteq \text{GOTO}(I, X)$$

کلیه‌ی آیتم‌های موجود در مجموعه‌ی I که در آنها نقطه قبل از یک نماد X قرار دارد،
با انتقال نقطه به بعد از X به $I' = \text{GOTO}(I,X)$ اضافه می‌شوند و
سپس $\text{CLOSURE}(I')$ محاسبه می‌شود.

عملیات برو به

منطق عملیات برو به

GOTO OPERATION

کلیه‌ی آیتم‌های موجود در مجموعه‌ی I که در آنها نقطه قبل از یک نماد X قرار دارد،
با انتقال نقطه به بعد از X به $I' = \mathrm{GOTO}(I,X)$ اضافه می‌شوند و
سپس $\mathrm{CLOSURE}(I')$ محاسبه می‌شود.

اگر I مجموعه‌ی آیتم‌های مجاز برای پیشوند γ از یک فرم جمله‌ای راست باشد،
در این صورت
$\mathrm{GOTO}(I,X)$
مجموعه‌ی آیتم‌های مجاز برای پیشوند γX است.

عملیات برو به

مثال

GOTO OPERATION

I (یک مجموعه از آیتم‌ها) → **عملیات برو به** با نماد X → $\text{GOTO}(I,X)$ (یک مجموعه از آیتم‌ها)

$$
\begin{aligned}
E' &\rightarrow E\$ \\
E &\rightarrow E + T \\
E &\rightarrow T \\
T &\rightarrow T * F \\
T &\rightarrow F \\
F &\rightarrow (E) \\
F &\rightarrow \mathbf{id}
\end{aligned}
$$

$$
\begin{aligned}
&\text{GOTO}(\{E' \rightarrow E \bullet \$, E \rightarrow E \bullet + T\}, +) = \\
&\text{CLOSURE}(\{E' \rightarrow E + \bullet T\}) = \{ \\
&E \rightarrow E + \bullet T, \\
&T \rightarrow \bullet T * F, \\
&T \rightarrow \bullet F, \\
&F \rightarrow \bullet (E), \\
&F \rightarrow \bullet \mathbf{id}\}
\end{aligned}
$$

روال ساخت آتوماتون LR(0)

LR(0) AUTOMATON

هر حالت آتوماتون متناهی (DFA) حاوی یک مجموعه آیتم است:

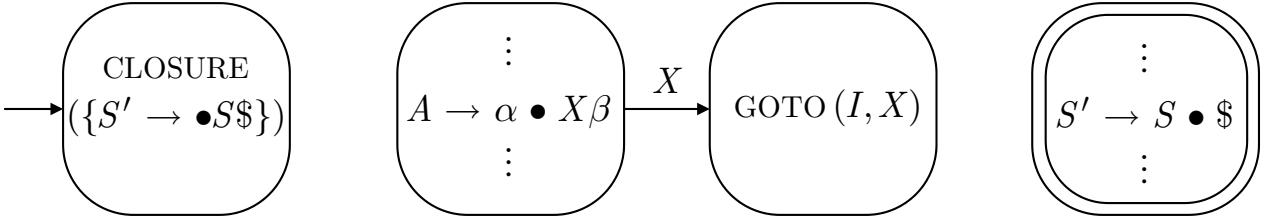
هر حالت DFA حاوی یک مجموعه آیتم است و به صورت زیر ساخته می‌شود:

- حالت شروع $\text{CLOSURE}(\{S' \to \bullet S\$\})$ است.
- به ازای هر نماد X که در یکی از آیتم‌های مجموعه‌ی مربوط به حالت I به صورت

$$A \to \alpha \bullet X\beta$$

قرار دارد، بین I و $\text{GOTO}(I, X)$ یک گذر با برچسب X ایجاد می‌شود.

- حالت نهایی DFA، حالتی است که در آن آیتم $S' \to S \bullet \$$ قرار دارد (نقطه به انتها رسیده است).



آتوماتون LR(0)

کاربرد

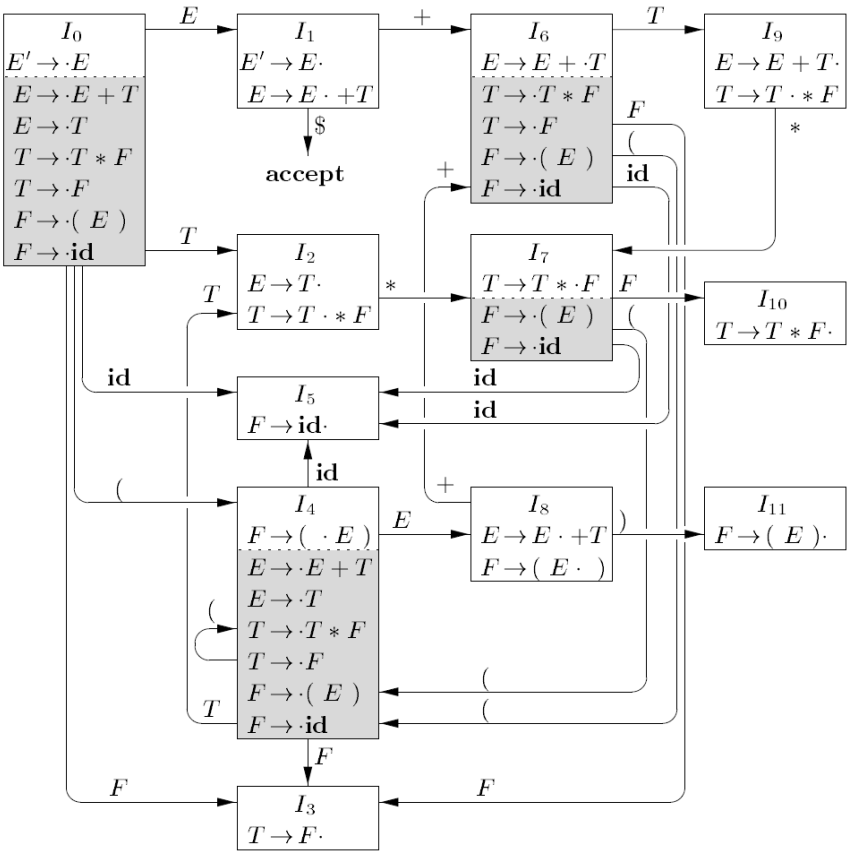
آتوماتون LR(0) پیشوندهای نموپذیر یک گرامر را نشان می‌دهد.

پیشوند نموپذیر *Viable Prefix*	پیشوند یک فرم جمله‌ای راست که می‌تواند در بالای پشته‌ی تجزیه‌گر ظاهر شود.

آتوماتون LR(0)

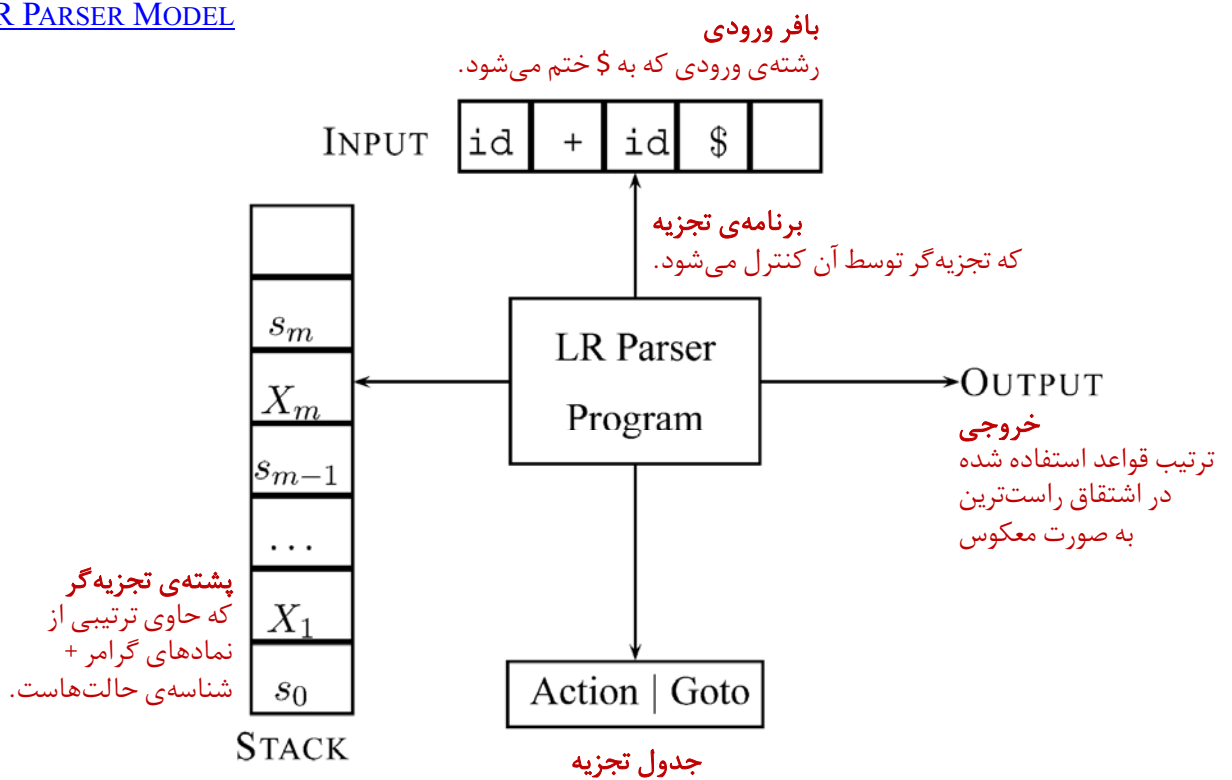
مثال

$$
\begin{aligned}
E' &\rightarrow E\$ \\
E &\rightarrow E + T \\
E &\rightarrow T \\
T &\rightarrow T * F \\
T &\rightarrow F \\
F &\rightarrow (E) \\
F &\rightarrow \mathbf{id}
\end{aligned}
$$



مدل تجزیه‌گر LR

LR Parser Model



الگوریتم تجزیه‌ی LR

LR Parsing Algorithm

هر گام از عملیات تجزیه‌گر بر اساس زوج زیر و با مراجعه به قسمت ACTION **جدول تجزیه** مشخص می‌شود:

$$(s_m, a_i)$$

s_m: بالای پشته: شناسه‌ی حالت

a_i: ورودی: توکن جاری

۱	**شیفت** *Shift*	انتقال توکن جاری از ورودی به بالای پشته	$\text{ACTION}[s_m, a_i] = \text{shift}(s_j)$
۲	**کاهش** *Reduce*	۱) یافتن دستگیره + pop کردن دستگیره از بالای پشته ۲) push کردن ناپایانه‌ی سمت چپ دستگیره به بالای پشته	$\text{ACTION}[s_m, a_i] = \text{reduce}(A \rightarrow \beta)$
۳	**پذیرش** *Accept*	پایان موفقیت‌آمیز تجزیه	$\text{ACTION}[s_m, \$] = \text{accept}$
۴	**خطا** *Error*	برخورد با خطا (فراخوانی روال اعلام و گذر از خطا)	$\text{ACTION}[s_m, a_i] = \text{error}$

در ابتدا شماره‌ی حالت 0 به بالای پشته درج می‌شود.

الگوریتم تجزیه‌ی LR

عمل شیفت

LR Parsing Algorithm

هر گام از عملیات تجزیه‌گر بر اساس زوج زیر و با مراجعه به قسمت ACTION **جدول تجزیه** مشخص می‌شود:

$$(s_m, a_i)$$

s_m: بالای پشته: شناسه‌ی حالت

a_i: ورودی: توکن جاری

۱	**شیفت** *Shift*	انتقال توکن جاری از ورودی به بالای پشته	$\text{ACTION}[s_m, a_i] = \text{shift}(s_j)$

$$\text{ACTION}[s_m, a_i] = \text{shift}(s_j)$$

شیفت a_i به بالای پشته و قرار دادن s_j در بالای آن:

$$(s_0 X_1 s_1 \dots s_{m-1} X_m s_m, a_i a_{i+1} \dots a_n \$) \mapsto (s_0 X_1 s_1 \dots s_{m-1} X_m s_m a_i s_j, a_{i+1} \dots a_n \$)$$

الگوریتم تجزیه‌ی LR

عمل کاهش

LR Parsing Algorithm

هر گام از عملیات تجزیه‌گر بر اساس زوج زیر و با مراجعه به قسمت ACTION **جدول تجزیه** مشخص می‌شود:

$$(s_m, a_i)$$

s_m: بالای پشته: شناسه‌ی حالت

a_i: ورودی: توکن جاری

۲ کاهش *Reduce*

۱) یافتن دستگیره + pop کردن دستگیره از بالای پشته
۲) push کردن ناپایانه‌ی سمت چپ دستگیره به بالای پشته

$$\text{ACTION}[s_m, a_i] = \text{reduce}(A \to \beta)$$

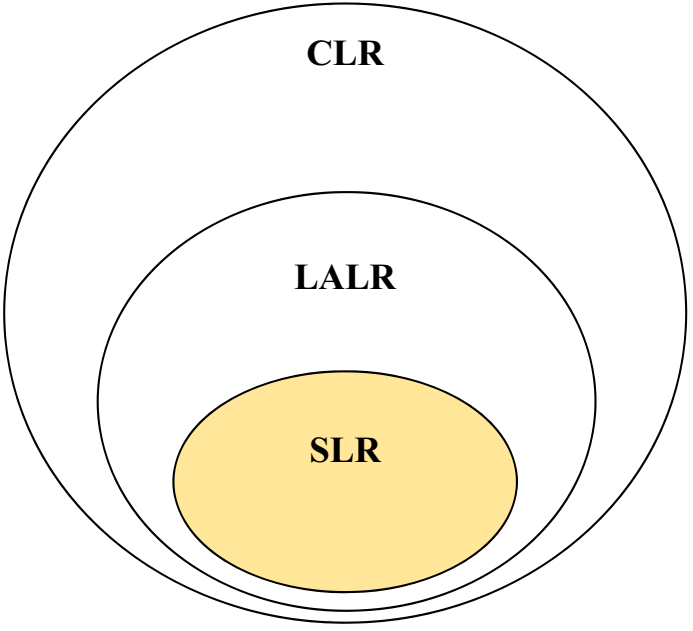
$$\text{ACTION}[s_m, a_i] = \text{reduce}(A \to \beta)$$

عمل کاهش انجام می‌شود: اگر $s = \text{GOTO}(s_{m-r}, A)$ و $r = |\beta|$، تجزیه‌گر $2r$ نماد از بالای پشته حذف می‌کند (r حالت و r نماد گرامر) و A و سپس s را در بالای پشته قرار می‌دهد:

$$(s_0 X_1 s_1 \dots s_{m-1} X_m s_m, a_i a_{i+1} \dots a_n \$) \mapsto (s_0 X_1 s_1 \dots s_{m-1} X_{m-r} s_{m-r} A s, a_i a_{i+1} \dots a_n \$)$$

روش‌های تجزیه‌ی LR

LR PARSING METHODS



الگوریتم تجزیه برای همه‌ی روش‌های LR یکسان است.
تفاوت آنها تنها در آتوماتون تشخیص‌دهنده‌ی دستگیره است.

ساخت جدول تجزیه‌ی SLR

SLR Parsing Table

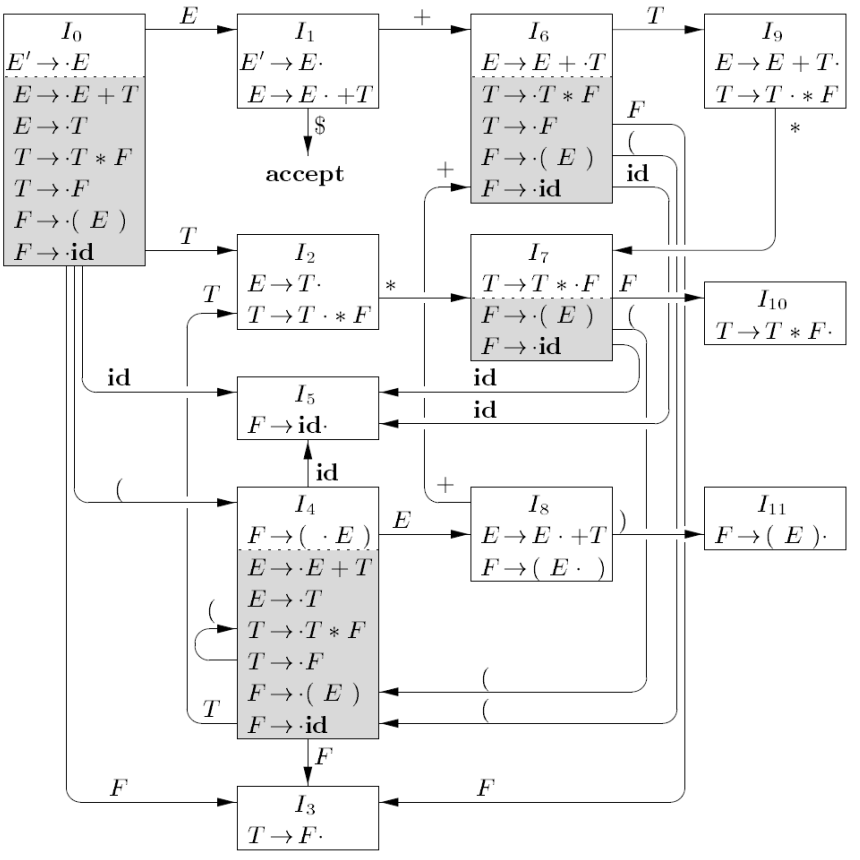
جدول تجزیه‌ی SLR همان عملکرد آتوماتون LR(0) را نشان می‌دهد.

- اگر $\text{GOTO}(I_i, a) = I_j$ در این صورت $\text{ACTION}[i, a] = \text{shift}(j)$.
- اگر $A \to \alpha\bullet \in I_i$، برای هر $a \in Follow(A)$ داریم $\text{ACTION}[i, a] = \text{reduce}(A \to \alpha)$.
- اگر $S' \to S \bullet \$ \in I_i$، داریم $\text{ACTION}[i, \$] = \text{accept}$.
- اگر $\text{GOTO}(I_i, A) = I_j$، داریم $\text{GOTO}[i, A] = j$.
- اگر خانه‌ای از جدول با قواعد فوق پر نشد، آن خانه حاوی error است.
- حالت آغازین، حالتی است که آیتم $S' \to \bullet S\$$ را در خود دارد.

آتوماتون LR(0)

مثال

$$
\begin{aligned}
E' &\rightarrow E\$ \\
E &\rightarrow E + T \\
E &\rightarrow T \\
T &\rightarrow T * F \\
T &\rightarrow F \\
F &\rightarrow (E) \\
F &\rightarrow \mathbf{id}
\end{aligned}
$$



جدول تجزیه‌ی SLR

مثال

	Action						Goto		
State	**id**	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6		s4		acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

$$E' \rightarrow E\$ \quad (0)$$
$$E \rightarrow E + T \quad (1)$$
$$E \rightarrow T \quad (2)$$
$$T \rightarrow T * F \quad (3)$$
$$T \rightarrow F \quad (4)$$
$$F \rightarrow (E) \quad (5)$$
$$F \rightarrow \mathbf{id} \quad (6)$$

$sn \equiv \text{shift}(n)$ $\qquad$ $rm \equiv \text{reduce}(m),\ m = \text{Rule-Number}(A \rightarrow \beta)$

تجزیه‌ی LR

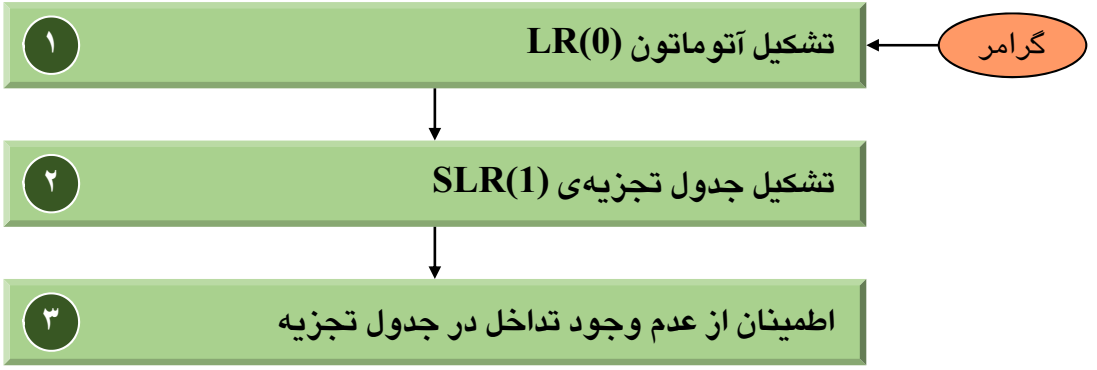
مثال

Stack	Input	Action
0	**id** $*$ **id** $+$ **id**\$	shift
0 **id** 5	$*$**id** $+$ **id**\$	reduce($F \rightarrow$ **id**)
0 F 3	$*$**id** $+$ **id**\$	reduce($T \rightarrow F$)
0 T 2	$*$**id** $+$ **id**\$	shift
0 T 2 $*$ 7	**id** $+$ **id**\$	shift
0 T 2 $*$ 7 **id** 5	$+$**id**\$	reduce($F \rightarrow$ **id**)
0 T 2 $*$ 7 F 10	$+$**id**\$	reduce($T \rightarrow T * F$)
0 T 2	$+$**id**\$	reduce($E \rightarrow T$)
0 E 1	$+$**id**\$	shift
0 E 1 $+$ 6	**id**\$	shift
0 E 1 $+$ 6 **id** 5	\$	reduce($F \rightarrow$ **id**)
0 E 1 $+$ 6 F 3	\$	reduce($T \rightarrow F$)
0 E 1 $+$ 6 T 9	\$	reduce($E \rightarrow E + T$)
0 E 1	\$	accept

	Action						Goto		
State	**id**	$+$	$*$	(	)	\$	E	T	F
0	s5			s4			1	2	3
1		s6		s4		acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

$E' \rightarrow E\$$ (0)
$E \rightarrow E + T$ (1)
$E \rightarrow T$ (2)
$T \rightarrow T * F$ (3)
$T \rightarrow F$ (4)
$F \rightarrow (E)$ (5)
$F \rightarrow$ **id** (6)

روال طراحی و ساخت تجزیه‌گر SLR



ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۱ از ۹)

$$S' \rightarrow S\$$$
$$S \rightarrow (L)$$
$$S \rightarrow \mathbf{id}$$
$$L \rightarrow S$$
$$L \rightarrow L, S$$

ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۲ از ۹)

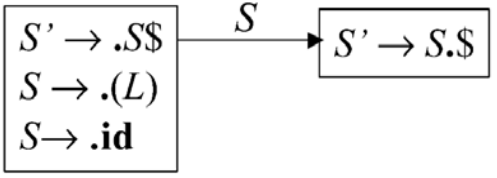
$$
\begin{aligned}
S' &\rightarrow S\$ \\
S &\rightarrow (L) \\
S &\rightarrow \mathbf{id} \\
L &\rightarrow S \\
L &\rightarrow L, S
\end{aligned}
$$

$S' \rightarrow .S\$$
$S \rightarrow .(L)$
$S \rightarrow .\mathbf{id}$

ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۳ از ۹)

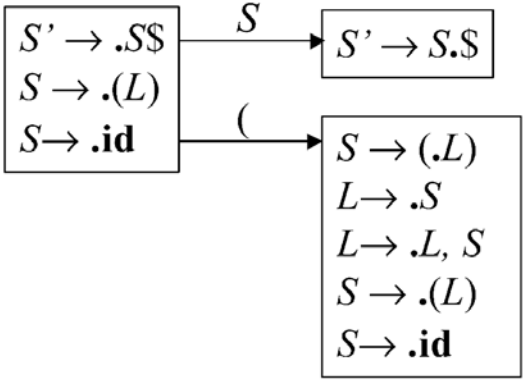
$$S' \rightarrow S\$$$
$$S \rightarrow (L)$$
$$S \rightarrow \mathbf{id}$$
$$L \rightarrow S$$
$$L \rightarrow L, S$$



ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۴ از ۹)

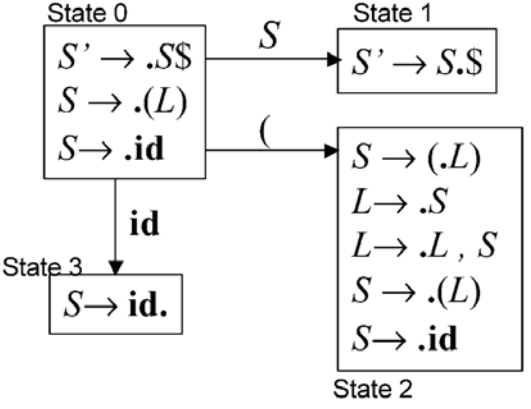
$$S' \rightarrow S\$$$
$$S \rightarrow (L)$$
$$S \rightarrow \mathbf{id}$$
$$L \rightarrow S$$
$$L \rightarrow L, S$$



ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۵ از ۹)

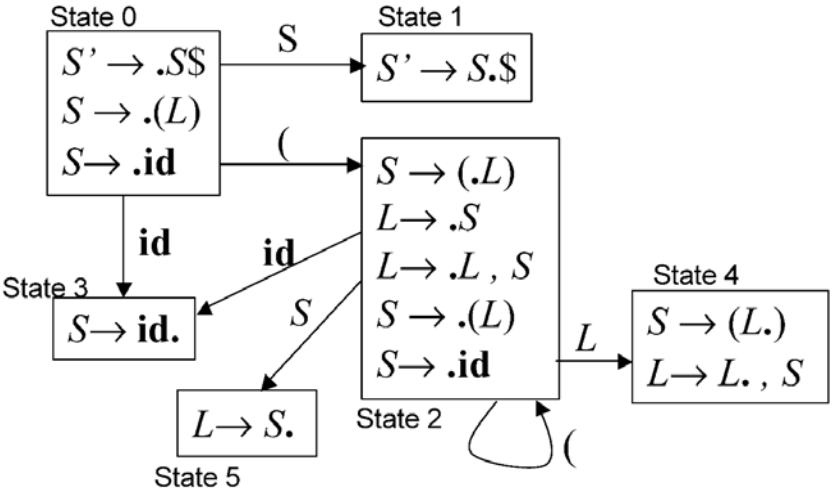
$$S' \rightarrow S\$$$
$$S \rightarrow (L)$$
$$S \rightarrow \mathbf{id}$$
$$L \rightarrow S$$
$$L \rightarrow L, S$$



ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۶ از ۹)

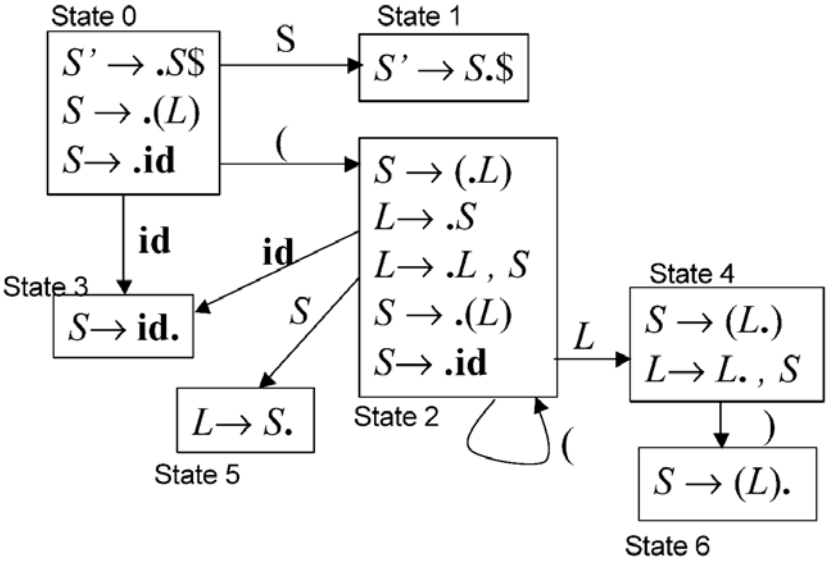
$$S' \rightarrow S\$$$
$$S \rightarrow (L)$$
$$S \rightarrow \mathbf{id}$$
$$L \rightarrow S$$
$$L \rightarrow L, S$$



ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۷ از ۹)

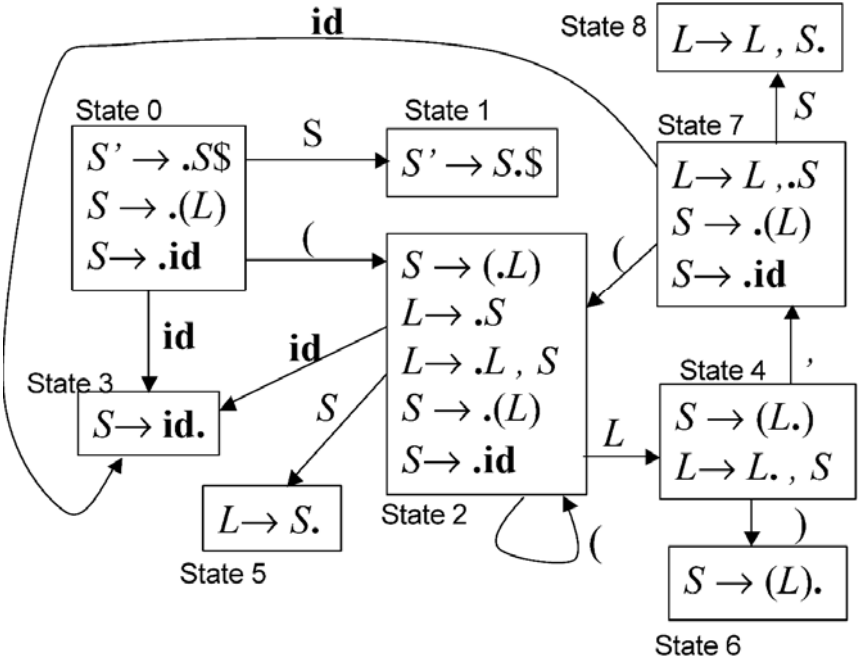
$$S' \rightarrow S\$$$
$$S \rightarrow (L)$$
$$S \rightarrow \mathbf{id}$$
$$L \rightarrow S$$
$$L \rightarrow L, S$$



ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۸ از ۹)

$$S' \rightarrow S\$$$
$$S \rightarrow (L)$$
$$S \rightarrow \mathbf{id}$$
$$L \rightarrow S$$
$$L \rightarrow L, S$$



ساخت آتوماتون LR(0) و جدول تجزیه‌ی SLR

مثال: (۹ از ۹)

	Action					Goto	
State	(	)	**id**	,	$	S	L
0	s2		s3			1	
1					acc		
2	s2		s3			5	4
3		r3		r3	r3		
4		s6		s7			
5		r4		r4			
6		r2		r2	r2		
7	s2		s3			8	
8		r5		r5			

$$S' \rightarrow S\$ \quad (1)$$
$$S \rightarrow (L) \quad (2)$$
$$S \rightarrow \mathbf{id} \quad (3)$$
$$L \rightarrow S \quad (4)$$
$$L \rightarrow L, S \quad (5)$$

گرامرهای غیر SLR

اگر جدول تجزیه‌ی SLR برای گرامری حاوی تداخل باشد، آن گرامر SLR نیست.

گرامرهای غیر SLR

مثال

Augmented Grammar

$S' \rightarrow S$

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow \text{id}$

$R \rightarrow L$

I_0 :
$S' \rightarrow .S$
$S \rightarrow .L = R$
$S \rightarrow .R$
$L \rightarrow .*R$
$L \rightarrow .\text{id}$
$R \rightarrow .L$

I_1 :
$S' \rightarrow S.$

I_2 :
$S \rightarrow L. = R$
$R \rightarrow L.$

I_3 :
$S \rightarrow R.$

I_4 :
$L \rightarrow *.R$
$R \rightarrow .L$
$L \rightarrow .*R$
$L \rightarrow .\text{id}$

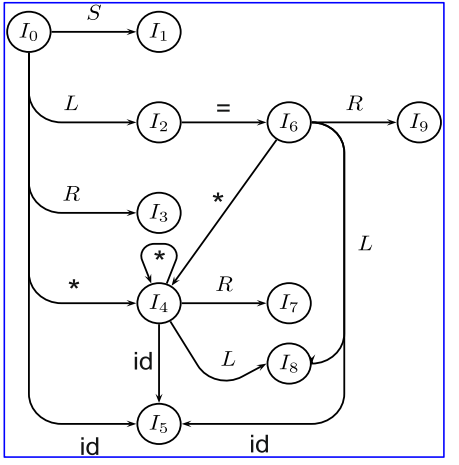
I_5 :
$L \rightarrow \text{id}.$

I_6 :
$S \rightarrow L = .R$
$R \rightarrow .L$
$L \rightarrow .*R$
$L \rightarrow .\text{id}$

I_7 :
$L \rightarrow *R.$

I_8 :
$R \rightarrow L.$

I_9 :
$S \rightarrow L = R.$



تداخل شیفت ـ کاهش

Shift-Reduce Conflict

I_2 :

$\text{ACTION}[2, =] = \text{shift}(6)$

$\text{ACTION}[2, =] = \text{reduce}(R \rightarrow L), \quad \text{since } = \in Follow(R)$

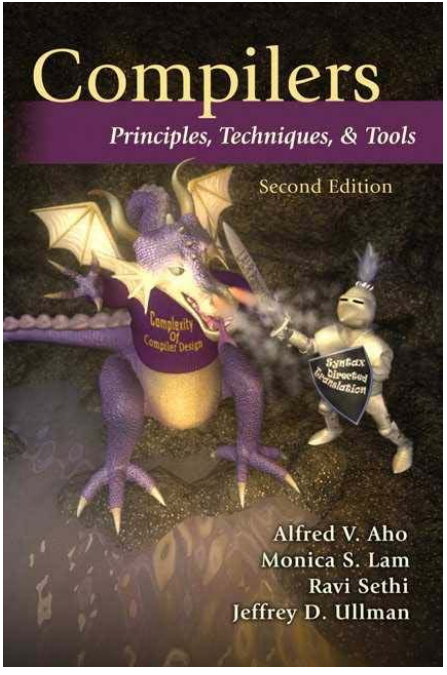
اصول طراحی کامپایلر

تحلیل نحوی
تجزیه‌ی پایین به بالا
روش SLR

۳

منابع

منبع اصلی



A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman,
Compilers: Principles, Techniques and Tools,
Second Edition, Addison-Wesley, 2007.

Chapter 4 (4.6)