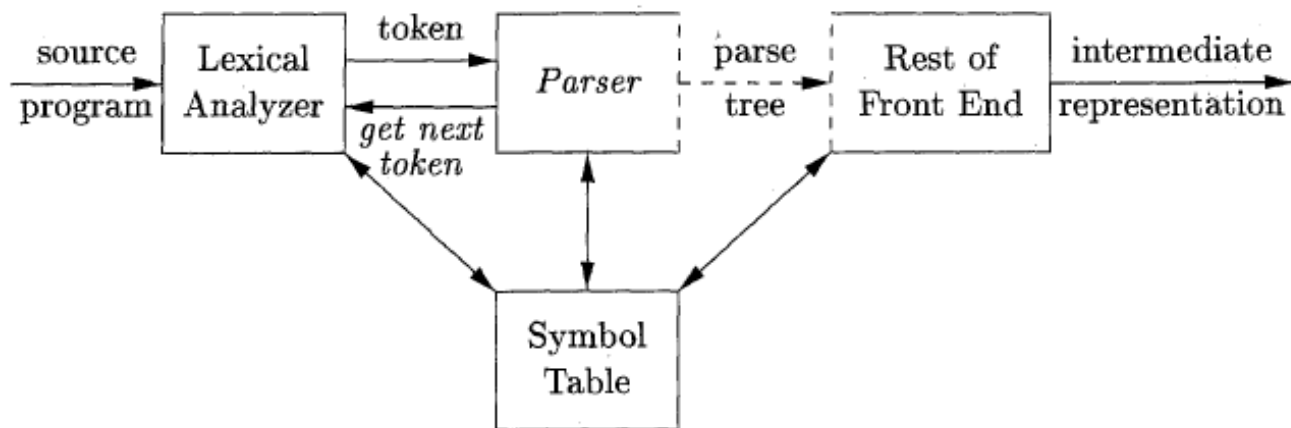

Syntax Analysis

التحليل النحوي

دور محلل القواعد Role of Syntactical Analyzer

- يستقبل محلل القواعد سلسلة tokens من محلل المفردات / يتأكد من إمكانية توليد السلسلة / يشير إلى الأخطاء / يتابع التنفيذ رغم وجودها.
- خوارزمية CYK لا تملك الكفاءة الكافية لاعتمادها في إنتاج المحلل القواعدي.
- Top Down Parsers تسمح سلسلة الرموز من اليسار إلى اليمين وتبني شجرة الإعراب من القمة (الجذر) نحو القاع (الأوراق)
- Bottom Up Parsers تسمح سلسلة الرموز من اليسار إلى اليمين وتبدأ بالأوراق وتعمل نحو الجذر.



Push-Down Automata & Context-Free Derivation

اشتقاق اللغات خارج السياق والأوتومات بمكدس

□ لدينا القواعد خارج السياق التالية $\{S \rightarrow aSa \mid bSb \mid \varepsilon\}$ ونريد إعراب الجملة $abba$:

□ قواعد الانتقال للأوتومات بمكدس المكافئ لها هي كالآتي:

□ $\delta(s_0, \varepsilon, \varepsilon) \rightarrow (f, S)$

□ $\delta(f, \varepsilon, S) \rightarrow (f, aSa)$

□ $\delta(f, \varepsilon, S) \rightarrow (f, bSb)$

□ $\delta(f, \varepsilon, S) \rightarrow (f, \varepsilon)$

□ $\delta(f, a, a) \rightarrow (f, \varepsilon)$

□ $\delta(f, b, b) \rightarrow (f, \varepsilon)$

state	input	stack	δ
s_0	\$abba	\$	1
f	\$abba	S\$	2
f	\$abba	aSa\$	5
f	\$abb	Sa\$	3
f	\$abb	bSba\$	6
f	\$ab	Sba\$	4
f	\$ab	ba\$	6
f	\$a	a\$	5
f	\$	\$	

Top-Down Parsing

تحليل من الأعلى للأسفل

□ التحليل من الأعلى للأسفل هو إيجاد الاشتقاق اليساري لسلسلة الدخل من خلال بناء شجرة الإعراب لسلسلة الدخل / انطلاقاً من الجذر وإنشاء لعقد الشجرة بالعمق أولاً (preorder).

□ Recursive-descend parsing يقوم بكل خطوة (حتى تنتهي سلسلة الدخل):

■ إما مطابقة Matching لل terminal في القواعد وسلسلة الدخل

■ أو إيجاد القاعدة المناسبة لتطبيقها (تحتاج أحياناً backtracking)

```
void A() {  
    Choose an A-production,  $A \rightarrow X_1 X_2 \dots X_k$ ;  
    for ( i = 1 to k ) {  
        if (  $X_i$  is a nonterminal )  
            call procedure  $X_i$  () ;  
        else if (  $X_i$  equals the current input symbol a )  
            advance the input to the next symbol;  
        else /* an error has occurred */;  
    }
```

}

Top-Down Parsing

تحليل من الأعلى للأسفل

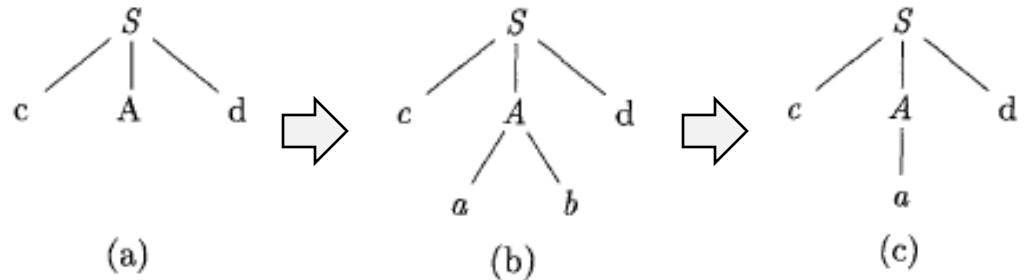
إعراب الجملة cad باستخدام القواعد

- $S \rightarrow c A d$
- $A \rightarrow a b \mid a$

انتقالات الأوتومات بمكدس

- $\delta(s_0, \epsilon, \epsilon) \rightarrow (f, S)$
- $\delta(f, \epsilon, S) \rightarrow (f, cAd)$
- $\delta(f, \epsilon, A) \rightarrow (f, ab)$
- $\delta(f, \epsilon, A) \rightarrow (f, a)$
- $\delta(f, a, a) \rightarrow (f, \epsilon)$
- $\delta(f, b, b) \rightarrow (f, \epsilon)$
- $\delta(f, c, c) \rightarrow (f, \epsilon)$
- $\delta(f, d, d) \rightarrow (f, \epsilon)$

state	input	stack	δ
s_0	cad	ϵ	1
f	cad	S	2
f	cad	cAd	7
f	ad	Ad	4
f	ad	ad	5
f	d	d	8
f	ϵ	ϵ	

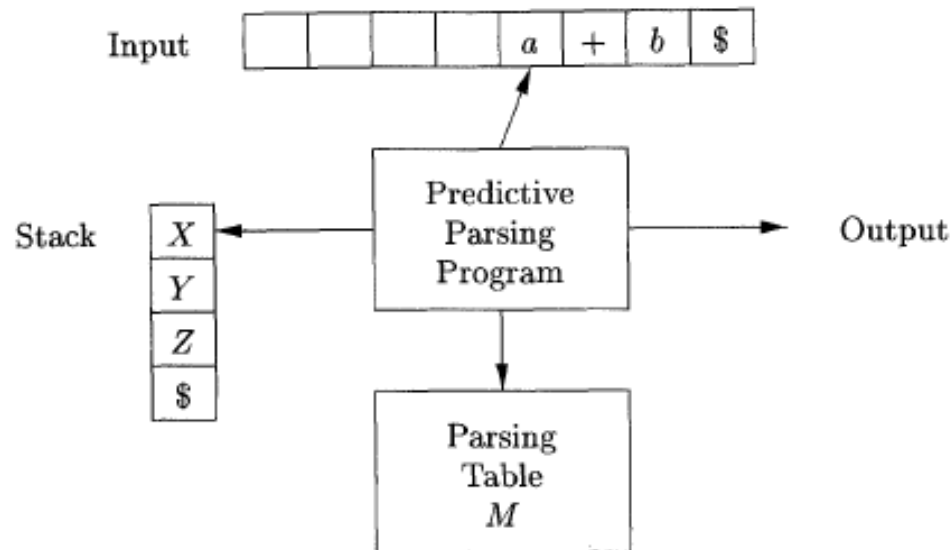


Top-Down Parsing

تحليل من الأعلى للأسفل

□ Predictive parsing يختار القاعدة الصحيحة من خلال النظر للأمام في سلسلة الدخل (بعدد محدد من الرموز)

□ Nonrecursive Predictive Parser يقوم بشكل صريح بإدارة المكس



Top-Down Parsing

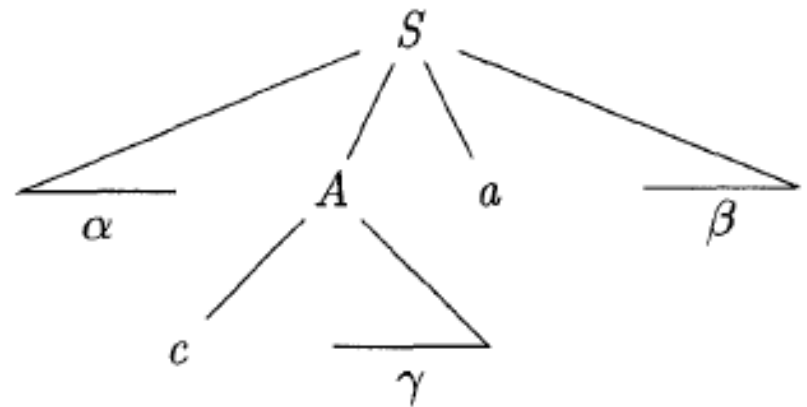
تحليل من الأعلى للأسفل

Nonrecursive Predictive Parser LL(1) □

```
set ip to point to the first symbol of w$;  
set X to the top stack symbol;  
set a the symbol pointed to it by ip  
while ( X != $ ) { /* 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] =  $X \rightarrow Y_1 Y_2 \dots Y_k$  ) {  
        output the production  $X \rightarrow Y_1 Y_2 \dots Y_k$ ;  
        pop the stack;  
        push  $Y_k, Y_{k-1}, \dots, Y_1$  onto the stack, with  $Y_1$  on top;  
    }  
    set X to the top stack symbol;  
}
```

First and Follow

- ❑ FIRST and FOLLOW allow us to choose which production to apply, based on the next input symbol.
- ❑ $\text{FIRST}(A)$ = the set of terminals that begin strings derived from A .
- ❑ $\text{FOLLOW}(A)$ = the set of terminals that can appear immediately to the right of A .



Compute FIRST

□ FIRST(X)

- If X is terminal $\rightarrow \text{FIRST}(X) = \{X\}$
- If $X \rightarrow^* \epsilon \rightarrow \text{FIRST}(X) = \text{FIRST}(X) \cup \{\epsilon\}$
- If $X \rightarrow \alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_n \rightarrow \text{FIRST}(X) = \text{FIRST}(\alpha_1) \cup \text{FIRST}(\alpha_2) \cup \dots \cup \text{FIRST}(\alpha_n)$
- If $X \rightarrow ABC \rightarrow \text{FIRST}(X) = \text{FIRST}(ABC)$
 - If ϵ doesn't in $\text{FIRST}(A) \rightarrow \text{FIRST}(X) = \text{FIRST}(A)$
 - If ϵ in $\text{FIRST}(A) \rightarrow \text{FIRST}(X) = \text{FIRST}(A) - \{\epsilon\} \cup \text{FIRST}(BC)$
and we compute $\text{FIRST}(BC)$ as above.

Compute FOLLOW

□ FOLLOW(X)

- If X is start symbol $\rightarrow \text{FOLLOW}(X) = \{\$ \}$
- If $A \rightarrow \alpha X \beta \rightarrow \text{FOLLOW}(X) = \text{FIRST}(\beta) \text{ except } \{\epsilon\}$
- If $A \rightarrow \alpha X \beta$ and $\beta \rightarrow \epsilon \rightarrow \text{add FOLLOW}(A) \text{ to FOLLOW}(X)$
- If $A \rightarrow \alpha X \rightarrow \text{add FOLLOW}(A) \text{ to FOLLOW}(X)$

Compute Predictive Parser Table

- Parser Table ...
- FOR each production $A \rightarrow \alpha$ DO
- Begin FOR
 - FOR each terminal a in $\text{FIRST}(\alpha)$ \rightarrow add $M[A, a] = A \rightarrow \alpha$
 - If ϵ in $\text{FIRST}(\alpha)$ \rightarrow add $A \rightarrow \alpha$ to $M[A, b]$ for each terminal b in $\text{FOLLOW}(A)$
 - If ϵ in $\text{FIRST}(\alpha)$ and $\$$ in $\text{FOLLOW}(A)$ \rightarrow add $A \rightarrow \alpha$ to $M[A, \$]$
- End FOR
- Make each undefined entry of M be error.

Full Example 1

productions

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow *FT' \mid \varepsilon$
- $F \rightarrow (E) \mid \text{id}$

first

- $\text{First}(E) = \{ (, \text{id} \}$
- $\text{First}(E') = \{ +, \varepsilon \}$
- $\text{First}(T) = \{ (, \text{id} \}$
- $\text{First}(T') = \{ *, \varepsilon \}$
- $\text{First}(F) = \{ (, \text{id} \}$

follow

- $\text{Follow}(E) = \{ \$,) \}$
- $\text{Follow}(E') = \{ \$,) \}$
- $\text{Follow}(T) = \{ \$,), + \}$
- $\text{Follow}(T') = \{ \$,), + \}$
- $\text{Follow}(F) = \{ *, \$,), + \}$

Full Example 1

productions

- ☐ $E \rightarrow TE'$
- ☐ $E' \rightarrow +TE' \mid \varepsilon$
- ☐ $T \rightarrow FT'$
- ☐ $T' \rightarrow *FT' \mid \varepsilon$
- ☐ $F \rightarrow (E) \mid \text{id}$

	+	*	()	id	\$
E			$E \rightarrow TE'$		$E \rightarrow TE'$	
E'	$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$		$E' \rightarrow \varepsilon$
T			$T \rightarrow FT'$		$T \rightarrow FT'$	
T'	$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$		$T' \rightarrow \varepsilon$
F			$F \rightarrow (E)$		$F \rightarrow \text{id}$	

Full Example 1

	+	*	()	id	\$
E			$E \rightarrow TE'$		$E \rightarrow TE'$	
E'	$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$		$E' \rightarrow \varepsilon$
T			$T \rightarrow FT'$		$T \rightarrow FT'$	
T'	$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$		$T' \rightarrow \varepsilon$
F			$F \rightarrow (E)$		$F \rightarrow id$	

Input	Stack	Action
id+id*id\$	E\$	
id+id*id\$	TE'\$	output $E \rightarrow TE'$
id+id*id\$	FT'E'\$	output $T \rightarrow FT'$
id+id*id\$	idT'E'\$	output $T \rightarrow id$
+id*id\$	T'E'\$	match id
+id*id\$	E'\$	output $T' \rightarrow \varepsilon$
+id*id\$	+TE' \$	output $E' \rightarrow +TE'$
id*id\$	TE' \$	match +

Full Example 1

	+	*	()	id	\$
E			$E \rightarrow TE'$		$E \rightarrow TE'$	
E'	$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$		$E' \rightarrow \varepsilon$
T			$T \rightarrow FT'$		$T \rightarrow FT'$	
T'	$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$		$T' \rightarrow \varepsilon$
F			$F \rightarrow (E)$		$F \rightarrow id$	

Input	Stack	Action
id*id\$	FT'E'\$	output $T \rightarrow FT'$
id*id\$	idT'E'\$	output $F \rightarrow id$
*id\$	T'E'\$	match id
*id\$	*FT'E'\$	output $T' \rightarrow *FT'$
id\$	FT'E'\$	match *
id\$	idT'E'\$	output $F \rightarrow id$
\$	T'E'\$	match id
\$	E'\$	output $T' \rightarrow \varepsilon$

Full Example 1

	+	*	()	id	\$
E			$E \rightarrow TE'$		$E \rightarrow TE'$	
E'	$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$		$E' \rightarrow \varepsilon$
T			$T \rightarrow FT'$		$T \rightarrow FT'$	
T'	$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$		$T' \rightarrow \varepsilon$
F			$F \rightarrow (E)$		$F \rightarrow id$	

Input	Stack	Action
\$	\$	output $E' \rightarrow \varepsilon$

Full Example 3

productions

- ☐ $S \rightarrow aBDh$
- ☐ $B \rightarrow cC$
- ☐ $C \rightarrow bC \mid \varepsilon$
- ☐ $D \rightarrow EF$
- ☐ $E \rightarrow g \mid \varepsilon$
- ☐ $F \rightarrow f \mid \varepsilon$

first

- ☐ $\text{First}(S) = \{a\}$
- ☐ $\text{First}(B) = \{c\}$
- ☐ $\text{First}(C) = \{b, \varepsilon\}$
- ☐ $\text{First}(D) = \{g, f, \varepsilon\}$
- ☐ $\text{First}(E) = \{g, \varepsilon\}$
- ☐ $\text{First}(F) = \{f, \varepsilon\}$

follow

- ☐ $\text{Follow}(S) = \{\$ \}$
- ☐ $\text{Follow}(B) = \{g, f \}$
- ☐ $\text{Follow}(C) = \{g, f \}$
- ☐ $\text{Follow}(D) = \{h\}$
- ☐ $\text{Follow}(E) = \{f, h\}$
- ☐ $\text{Follow}(F) = \{h\}$

Full Example 3

productions

☐ $S \rightarrow aBDh$

☐ $B \rightarrow cC$

☐ $C \rightarrow bC \mid \varepsilon$

☐ $D \rightarrow EF$

☐ $E \rightarrow g \mid \varepsilon$

☐ $F \rightarrow f \mid \varepsilon$

	a	b	c	f	g	h	\$
S	$S \rightarrow aBDh$						
B			$B \rightarrow cC$				
C		$C \rightarrow bC$			$C \rightarrow \varepsilon$	$C \rightarrow \varepsilon$	
D				$D \rightarrow EF$	$D \rightarrow EF$	$D \rightarrow \varepsilon$	
E				$E \rightarrow \varepsilon$	$E \rightarrow g$	$E \rightarrow \varepsilon$	
F				$F \rightarrow f$		$F \rightarrow \varepsilon$	

Full Example 3

	a	b	c	f	g	h	\$
S	$S \rightarrow aBDh$						
B			$B \rightarrow cC$				
C		$C \rightarrow bC$			$C \rightarrow \epsilon$	$C \rightarrow \epsilon$	
D				$D \rightarrow EF$	$D \rightarrow EF$	$D \rightarrow \epsilon$	
E				$E \rightarrow \epsilon$	$E \rightarrow g$	$E \rightarrow \epsilon$	
F				$F \rightarrow f$		$F \rightarrow \epsilon$	

Input	Stack	Action
acbgh\$	S\$	
acbgh\$	aBDh\$	$S \rightarrow aBDh$
cbgh\$	BDh\$	match a
cbgh\$	cCDh\$	$B \rightarrow cC$
bgh\$	CDh\$	match c
bgh\$	bCDh\$	$C \rightarrow bC$
gh\$	CDh\$	match b

Full Example 3

	a	b	c	f	g	h	\$
S	$S \rightarrow aBDh$						
B			$B \rightarrow cC$				
C		$C \rightarrow bC$			$C \rightarrow \epsilon$	$C \rightarrow \epsilon$	
D				$D \rightarrow EF$	$D \rightarrow EF$	$D \rightarrow \epsilon$	
E				$E \rightarrow \epsilon$	$E \rightarrow g$	$E \rightarrow \epsilon$	
F				$F \rightarrow f$		$F \rightarrow \epsilon$	

Input	Stack	Action
gh\$	CDh\$	match b
gh\$	Dh\$	$C \rightarrow \epsilon$
gh\$	EFh\$	$D \rightarrow EF$
gh\$	gFh\$	$E \rightarrow g$
h\$	Fh\$	match g
h\$	h\$	$F \rightarrow \epsilon$
\$	\$	match h