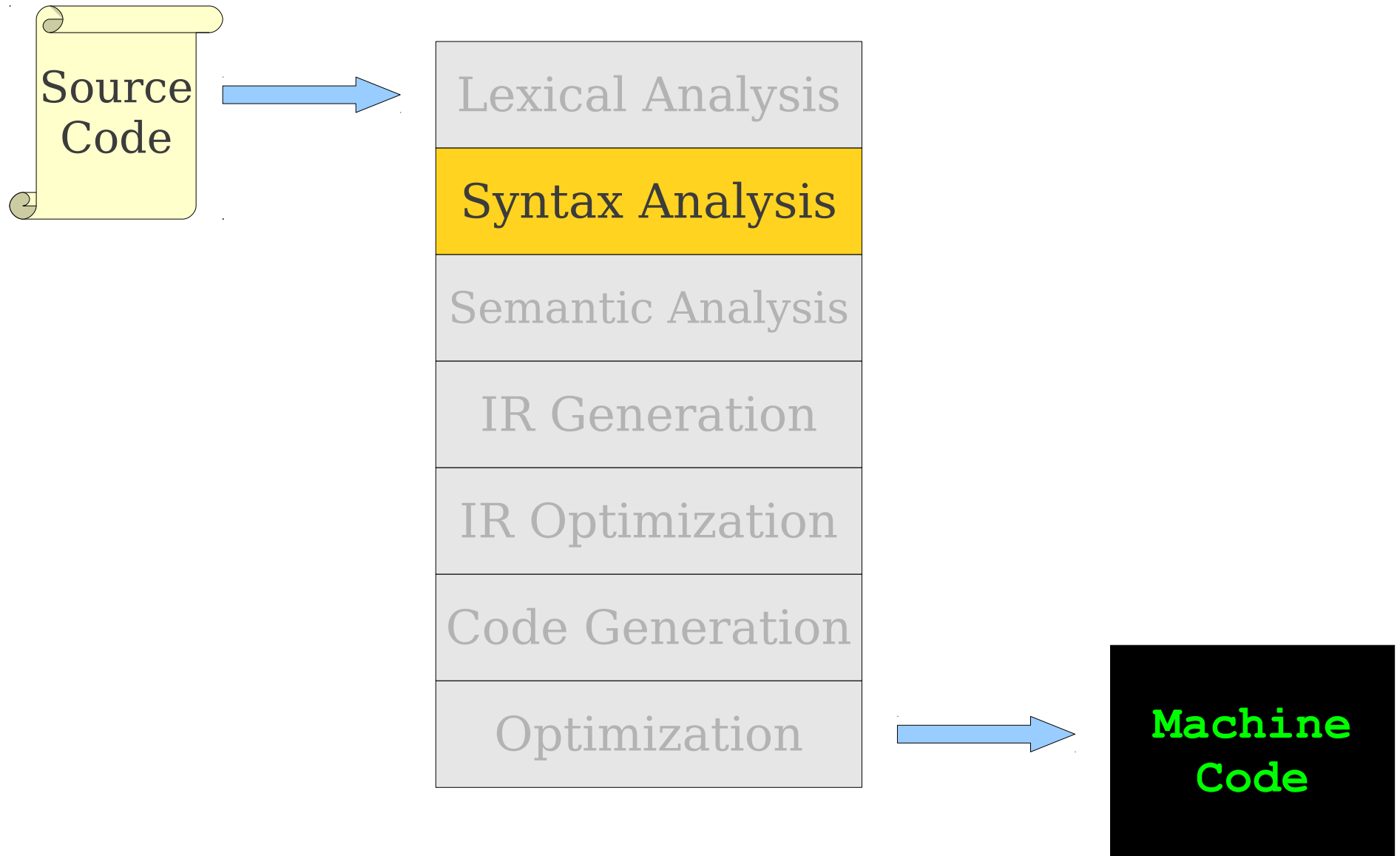


Top-Down Parsing II

Announcements

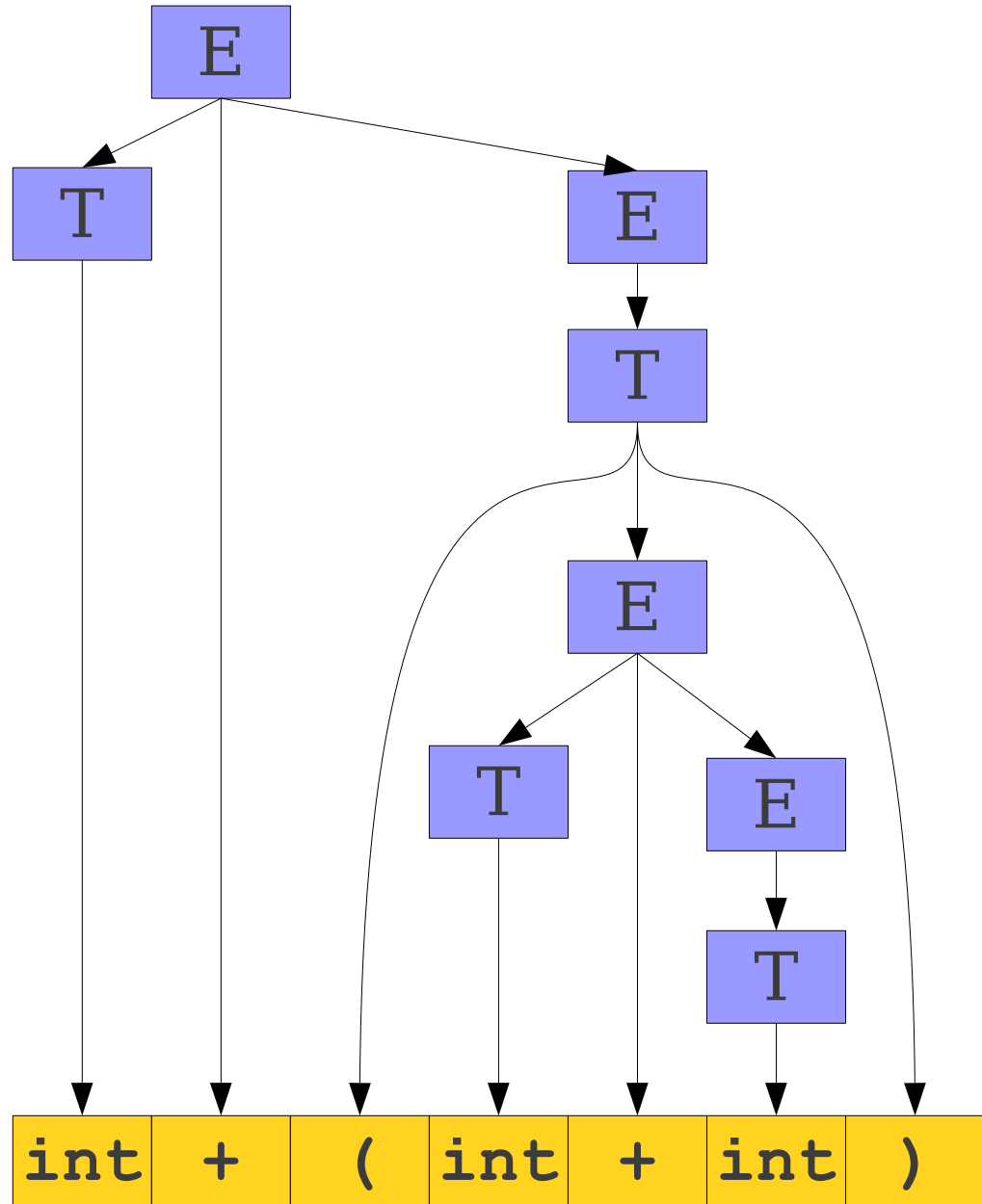
- Written Assignment 1 due this afternoon at 5PM.
 - Can submit electronically by emailing us at **cs143-sum1112-staff@lists.stanford.edu** with [WA1] somewhere in the subject line.
 - Can submit hard copies to the drop-off box in Gates (details in the problem set).
- C++ review session next Monday, time and place TBA.

Where We Are



Top-Down Parsing

$E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



LL(1) Parse Tables

E \rightarrow **int**

E \rightarrow (**E Op E**)

Op \rightarrow **+**

Op \rightarrow *****

	int	()	+	*
E	int	(E Op E)			
Op				+	*

FIRST Sets

- We want to tell if a particular nonterminal **A** derives a string starting with a particular nonterminal **t**.
- We can formalize this with **FIRST sets**.

$$\text{FIRST}(\mathbf{A}) = \{ \mathbf{t} \mid \mathbf{A} \Rightarrow^* \mathbf{t}\omega \text{ for some } \omega \}$$

- We also include **ε** in $\text{FIRST}(\mathbf{A})$ if A can produce the empty string.
- Intuitively, $\text{FIRST}(\mathbf{A})$ is the set of terminals that can be at the start of a string produced by **A**.
- We can generalize FIRST to strings with $\text{FIRST}^*(\omega)$ being the set of all terminals (or **ε**) that can appear at the start of a string derived from **ω**.

FIRST Computation with ε

- Initially, for all nonterminals A , set
$$\text{FIRST}(A) = \{ t \mid A \rightarrow t\omega \text{ for some } \omega \}$$
- For all nonterminals A where $A \rightarrow \varepsilon$ is a production, add ε to $\text{FIRST}(A)$.
- Repeat the following until no changes occur:
 - For each production $A \rightarrow \alpha$, set
$$\text{FIRST}(A) = \text{FIRST}(A) \cup \text{FIRST}^*(\alpha)$$

LL(1) Tables with ϵ

Num → **Sign Digits**
Sign → **+** | **-** | **ϵ**
Digits → **Digit More**
More → **Digits** | **ϵ**
Digit → **0** | **1** | ... | **9**

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

	+	-	#	\$
Num				
Sign				
Digits				
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**
Sign → **+ | - | ϵ**
Digits → **Digit More**
More → **Digits | ϵ**
Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num				
Sign				
Digits				
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+ | - | ϵ**

Digits → **Digit More**

More → **Digits | ϵ**

Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num				
Sign				
Digits				
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**
Sign → **+ | - | ϵ**
Digits → **Digit More**
More → **Digits | ϵ**
Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign				
Digits				
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+ | - | ϵ**

Digits → **Digit More**

More → **Digits | ϵ**

Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign				
Digits				
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+ | - | ϵ**

Digits → **Digit More**

More → **Digits | ϵ**

Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits				
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits				
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits			Digits More	
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits			Digits More	
More				
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits			Digits More	
More			Digits	
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits			Digits More	
More			Digits	
Digit				

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits			Digits More	
More			Digits	
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**
Sign → **+ | - | ϵ**
Digits → **Digit More**
More → **Digits | ϵ**
Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9								ϵ

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits			Digits More	
More			Digits	
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**
Sign → **+ | - | ϵ**
Digits → **Digit More**
More → **Digits | ϵ**
Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits		
Sign	+	-		
Digits			Digits More	
More			Digits	
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits	Sign Digits	
Sign	+	-		
Digits			Digits More	
More			Digits	
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+ | - | ϵ**

Digits → **Digit More**

More → **Digits | ϵ**

Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits	Sign Digits	
Sign	+	-		
Digits			Digits More	
More			Digits	
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits	Sign Digits	
Sign	+	-	ϵ	
Digits			Digits More	
More			Digits	
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits	Sign Digits	
Sign	+	-	ϵ	
Digits			Digits More	
More			Digits	
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits	Sign Digits	
Sign	+	-	ϵ	
Digits			Digits More	
More			Digits	
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits	Sign Digits	
Sign	+	-	ϵ	
Digits			Digits More	
More			Digits	ϵ
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**

Sign → **+** | **-** | **ϵ**

Digits → **Digit More**

More → **Digits** | **ϵ**

Digit → **0** | **1** | ... | **9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits	Sign Digits	
Sign	+	-	ϵ	
Digits			Digits More	
More			Digits	ϵ
Digit			#	

LL(1) Tables with ϵ

Num → **Sign Digits**
Sign → **+ | - | ϵ**
Digits → **Digit More**
More → **Digits | ϵ**
Digit → **0 | 1 | ... | 9**

Num		Sign		Digit		Digits		More	
+	-	+	-	0	5	0	5	0	5
0	5	ϵ		1	6	1	6	1	6
1	6			2	7	2	7	2	7
2	7			3	8	3	8	3	8
3	8			4	9	4	9	4	9
4	9							ϵ	

	+	-	#	\$
Num	Sign Digits	Sign Digits	Sign Digits	
Sign	+	-	ϵ	
Digits			Digits More	
More			Digits	ϵ
Digit			#	

FOLLOW Sets

- With ϵ -productions in the grammar, we may have to “look past” the current nonterminal to what can come after it.
- The **FOLLOW set** represents the set of terminals that might come after a given nonterminal.
- Formally:

$$\text{FOLLOW}(\mathbf{A}) = \{ \mathbf{t} \mid \mathbf{S} \Rightarrow^* \alpha \mathbf{A} \mathbf{t} \omega \text{ for some } \alpha, \omega \}$$

where \mathbf{S} is the start symbol of the grammar.

- Informally, every nonterminal that can ever come after \mathbf{A} in a derivation.

Computation of FOLLOW Sets

- Initially, for each nonterminal **A**, set
$$\text{FOLLOW}(\mathbf{A}) = \{ \mathbf{t} \mid \mathbf{B} \rightarrow \alpha \mathbf{A} \mathbf{t} \omega \text{ is a production} \}$$
- Add **\$** to FOLLOW(**S**), where **S** is the start symbol.
- Repeat the following until no changes occur:
 - If $\mathbf{B} \rightarrow \alpha \mathbf{A} \omega$ is a production, set
$$\text{FOLLOW}(\mathbf{A}) = \text{FOLLOW}(\mathbf{A}) \cup \text{FIRST}^*(\omega) - \{ \epsilon \}.$$
 - If $\mathbf{B} \rightarrow \alpha \mathbf{A} \omega$ is a production and $\epsilon \in \text{FIRST}^*(\omega)$, set
$$\text{FOLLOW}(\mathbf{A}) = \text{FOLLOW}(\mathbf{A}) \cup \text{FOLLOW}(\mathbf{B}).$$

The Final LL(1) Table Algorithm

- Compute $\text{FIRST}(\mathbf{A})$ and $\text{FOLLOW}(\mathbf{A})$ for all nonterminals \mathbf{A} .
- For each rule $\mathbf{A} \rightarrow \omega$, for each terminal $\mathbf{t} \in \text{FIRST}^*(\omega)$, set $T[\mathbf{A}, \mathbf{t}] = \omega$.
 - Note that ϵ is not a terminal.
- For each rule $\mathbf{A} \rightarrow \omega$, if $\epsilon \in \text{FIRST}^*(\omega)$, set $T[\mathbf{A}, \mathbf{t}] = \omega$ for each $\mathbf{t} \in \text{FOLLOW}(\mathbf{A})$.

An Egregious Abuse of Notation

- Compute $\text{FIRST}(\mathbf{A})$ and $\text{FOLLOW}(\mathbf{A})$ for all nonterminals \mathbf{A} .
- For each rule $\mathbf{A} \rightarrow \omega$, for each terminal $\mathbf{t} \in \text{FIRST}^*(\omega \text{FOLLOW}(\mathbf{A}))$, set $T[\mathbf{A}, \mathbf{t}] = \omega$.

Example LL(1) Construction

The Limits of LL(1)

A Grammar that is Not LL(1)

- Consider the following (left-recursive) grammar:

$$A \rightarrow Ab \mid c$$

- $\text{FIRST}(A) = \{c\}$
- However, we cannot build an LL(1) parse table.
- **Why?**

A Grammar that is Not LL(1)

- Consider the following (left-recursive) grammar:

$$A \rightarrow Ab \mid c$$

- $\text{FIRST}(A) = \{c\}$
- However, we cannot build an LL(1) parse table.
- Why?**

	b	c
A		$A \rightarrow Ab$ $A \rightarrow c$

A Grammar that is Not LL(1)

- Consider the following (left-recursive) grammar:

$$A \rightarrow Ab \mid c$$

- $\text{FIRST}(A) = \{c\}$
- However, we cannot build an LL(1) parse table.
- Why?**

	b	c
A		$A \rightarrow Ab$ $A \rightarrow c$

- Cannot uniquely predict production!**
- This is called a **FIRST/FIRST conflict**.

Eliminating Left Recursion

- In general, left recursion can be converted into **right recursion** by a mechanical transformation.
- Consider the grammar

$$\mathbf{A} \rightarrow \mathbf{A}\omega \mid \alpha$$

- This will produce α followed by some number of ω 's.
- Can rewrite the grammar as

$$\mathbf{A} \rightarrow \alpha \mathbf{B}$$

$$\mathbf{B} \rightarrow \epsilon \mid \omega \mathbf{B}$$

Another Non-LL(1) Grammar

- Consider the following grammar:

$$\mathbf{E} \rightarrow \mathbf{T}$$

$$\mathbf{E} \rightarrow \mathbf{T} + \mathbf{E}$$

$$\mathbf{T} \rightarrow \text{int}$$

$$\mathbf{T} \rightarrow (\mathbf{E})$$

- $\text{FIRST}(\mathbf{E}) = \{ \text{int}, (\}$
- $\text{FIRST}(\mathbf{T}) = \{ \text{int}, (\}$
- Why is this grammar not LL(1)?

Another Non-LL(1) Grammar

- Consider the following grammar:

$E \rightarrow T$

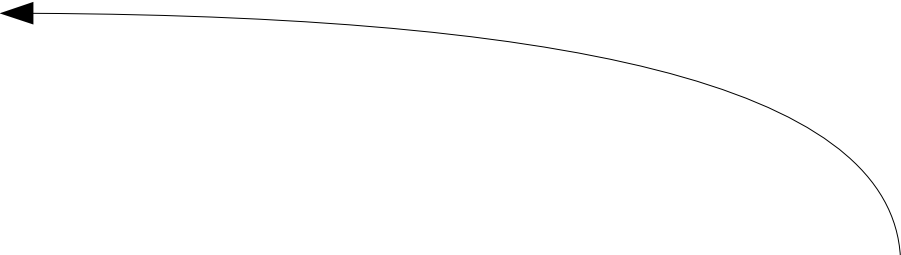
$E \rightarrow T + E$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

- $\text{FIRST}(E) = \{ \text{int}, (\}$
- $\text{FIRST}(T) = \{ \text{int}, (\}$
- Why is this grammar not LL(1)?

How do you
predict which of
these to use?



Left-Factoring

E → **T**

E → **T** + **E**

T → **int**

T → (**E**)

Left-Factoring

E \rightarrow **T** ϵ

E \rightarrow **T** + **E**

T \rightarrow **int**

T \rightarrow (**E**)

Left-Factoring

E → **TY**

T → **int**

T → **(E)**

Left-Factoring

E \rightarrow **T****Y**

T \rightarrow **int**

T \rightarrow (**E**)

Y \rightarrow **+** **E**

Y \rightarrow ϵ

Left-Factoring

E \rightarrow **T****Y** **1**

T \rightarrow **int** **2**

T \rightarrow (**E**) **3**

Y \rightarrow **+** **E** **4**

Y \rightarrow ϵ **5**

Left-Factoring

E	→	T Y	1
T	→	int	2
T	→	(E)	3
Y	→	+ E	4
Y	→	ε	5

Left-Factoring

E	→	T Y	1
T	→	int	2
T	→	(E)	3
Y	→	+ E	4
Y	→	ε	5

FIRST		
E	T	Y
FOLLOW		
E	T	Y

Left-Factoring

E	→	TY	1
T	→	int	2
T	→	(E)	3
Y	→	+ E	4
Y	→	ε	5

FIRST		
E	T	Y
	int (
FOLLOW		
E	T	Y

Left-Factoring

E	\rightarrow	TY	1
T	\rightarrow	int	2
T	\rightarrow	(E)	3
Y	\rightarrow	+ E	4
Y	\rightarrow	ϵ	5

FIRST		
E	T	Y
	int (+ ϵ
FOLLOW		
E	T	Y

Left-Factoring

E	→	T Y	1
T	→	int	2
T	→	(E)	3
Y	→	+ E	4
Y	→	ε	5

FIRST		
E	T	Y
int (int (+ ε
FOLLOW		
E	T	Y

Left-Factoring

E	\rightarrow	TY	1
T	\rightarrow	int	2
T	\rightarrow	(E)	3
Y	\rightarrow	+ E	4
Y	\rightarrow	ϵ	5

FIRST		
E	T	Y
int (int (+ ϵ
FOLLOW		
E	T	Y
\$		

Left-Factoring

E	\rightarrow	TY	1
T	\rightarrow	int	2
T	\rightarrow	(E)	3
Y	\rightarrow	+ E	4
Y	\rightarrow	ϵ	5

FIRST		
E	T	Y
int (int (+ ϵ
FOLLOW		
E	T	Y
\$)		

Left-Factoring

E	→	TY	1
T	→	int	2
T	→	(E)	3
Y	→	+ E	4
Y	→	ε	5

FIRST		
E	T	Y
int (int (+ ε
FOLLOW		
E	T	Y
\$)	+	

Left-Factoring

E	\rightarrow	TY	1
T	\rightarrow	int	2
T	\rightarrow	(E)	3
Y	\rightarrow	+ E	4
Y	\rightarrow	ϵ	5

FIRST		
E	T	Y
int (int (+ ϵ
FOLLOW		
E	T	Y
\$)	+	\$)

Left-Factoring

E	\rightarrow	TY	1
T	\rightarrow	int	2
T	\rightarrow	(E)	3
Y	\rightarrow	+ E	4
Y	\rightarrow	ϵ	5

FIRST		
E	T	Y
int (int (+ ϵ
FOLLOW		
E	T	Y
\$)	+ \$)	\$)

Left-Factoring

E → **TY** **1**
T → **int** **2**
T → **(E)** **3**
Y → **+ E** **4**
Y → **ε** **5**

FIRST		
E	T	Y
int (int (+ ε
FOLLOW		
E	T	Y
\$)	+ \$)	\$)

	int	()	+	\$
E					
T					
Y					

Left-Factoring

E → **TY** **1**
T → **int** **2**
T → **(E)** **3**
Y → **+ E** **4**
Y → **ε** **5**

FIRST		
E	T	Y
int (int (+ ε
FOLLOW		
E	T	Y
\$)	+ \$)	\$)

	int	()	+	\$
E	1	1			
T					
Y					

Left-Factoring

E \rightarrow **T****Y** **1**
T \rightarrow **int** **2**
T \rightarrow **(E)** **3**
Y \rightarrow **+** **E** **4**
Y \rightarrow **ϵ** **5**

FIRST		
E	T	Y
int (int (+ ϵ
FOLLOW		
E	T	Y
\$)	+ \$)	\$)

	int	()	+	\$
E	1	1			
T	2	3			
Y					

Left-Factoring

E → **TY** **1**
T → **int** **2**
T → **(E)** **3**
Y → **+ E** **4**
Y → **ε** **5**

FIRST		
E	T	Y
int (int (+ ε
FOLLOW		
E	T	Y
\$)	+ \$)	\$)

	int	()	+	\$
E	1	1			
T	2	3			
Y				4	

Left-Factoring

E → **TY** **1**
T → **int** **2**
T → **(E)** **3**
Y → **+ E** **4**
Y → **ε** **5**

FIRST		
E	T	Y
int (int (+ ε
FOLLOW		
E	T	Y
\$)	+ \$)	\$)

	int	()	+	\$
E	1	1			
T	2	3			
Y			5	4	5

A Formal Characterization of LL(1)

- A grammar G is LL(1) iff for any productions $\mathbf{A} \rightarrow \omega_1$ and $\mathbf{A} \rightarrow \omega_2$, the sets

$$\text{FIRST}(\omega_1 \text{ FOLLOW}(\mathbf{A}))$$

and

$$\text{FIRST}(\omega_2 \text{ FOLLOW}(\mathbf{A}))$$

are disjoint.

- This condition is equivalent to saying that there are no conflicts in the table.

The Strengths of LL(1)

LL(1) is Straightforward

- Can be implemented quickly with a table-driven design.
- Can be implemented by **recursive descent**:
 - Define a function for each nonterminal.
 - Have these functions call each other based on the lookahead token.
- See Handout #09 for more details.

LL(1) is Fast

- Both table-driven LL(1) and recursive-descent-powered LL(1) are fast.
- Can parse in $O(n |G|)$ time, where n is the length of the string and $|G|$ is the size of the grammar.

Summary

- **Top-down parsing** tries to derive the user's program from the start symbol.
- **Leftmost BFS** is one approach to top-down parsing; it is mostly of theoretical interest.
- **Leftmost DFS** is another approach to top-down parsing that is uncommon in practice.
- **LL(1)** parsing scans from left-to-right, using one token of lookahead to find a leftmost derivation.
- **FIRST sets** contain terminals that may be the first symbol of a production.
- **FOLLOW sets** contain terminals that may follow a nonterminal in a production.
- **Left recursion** and **left factorability** cause LL(1) to fail and can be mechanically eliminated in some cases.