Deterministic Bottom-Up Parsing

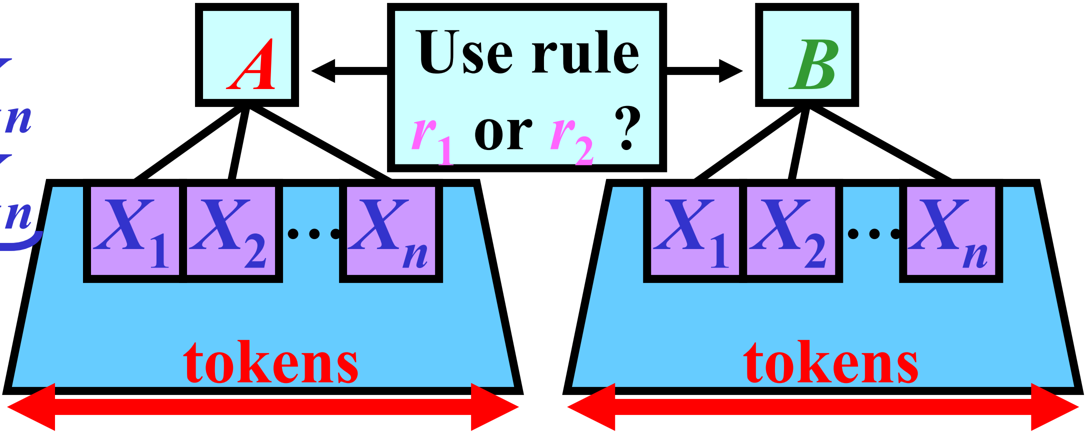
Chapter 5

Bottom-Up Parsing: Problems

1) Two or more rules have the same *handle*

r_1: $A \rightarrow X_1X_2\ldots X_n$

r_2: $B \rightarrow \underbrace{X_1X_2\ldots X_n}_{\text{handle}}$



Note: A *handle* is the right-hand side of a rule.

2) Ambiguous grammars

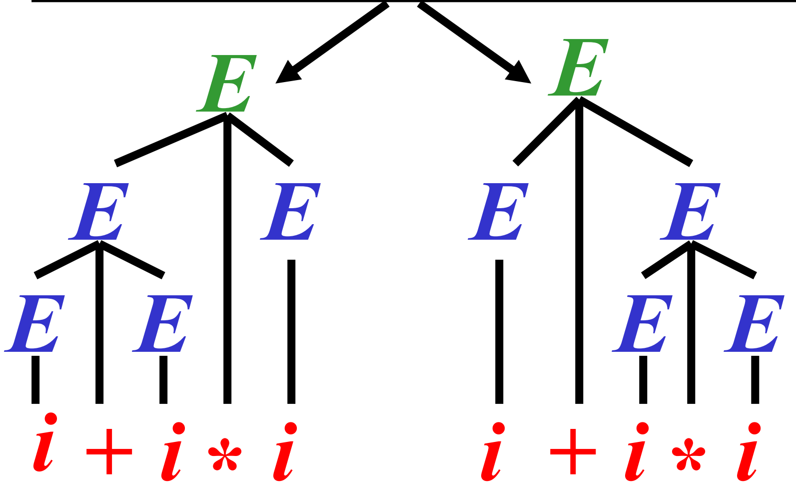
$G_{expr2} = (N, T, P, E)$, where

$N = \{E\}$, $T = \{i, +, *, (,)\}$,

$P = \{$ 1: $E \rightarrow E+E$, 2: $E \rightarrow E*E$,

3: $E \rightarrow (E)$, 4: $E \rightarrow i$ $\}$

Which of these tree to create?



Bottom-Up Parsers

1) Operator-precedence parser

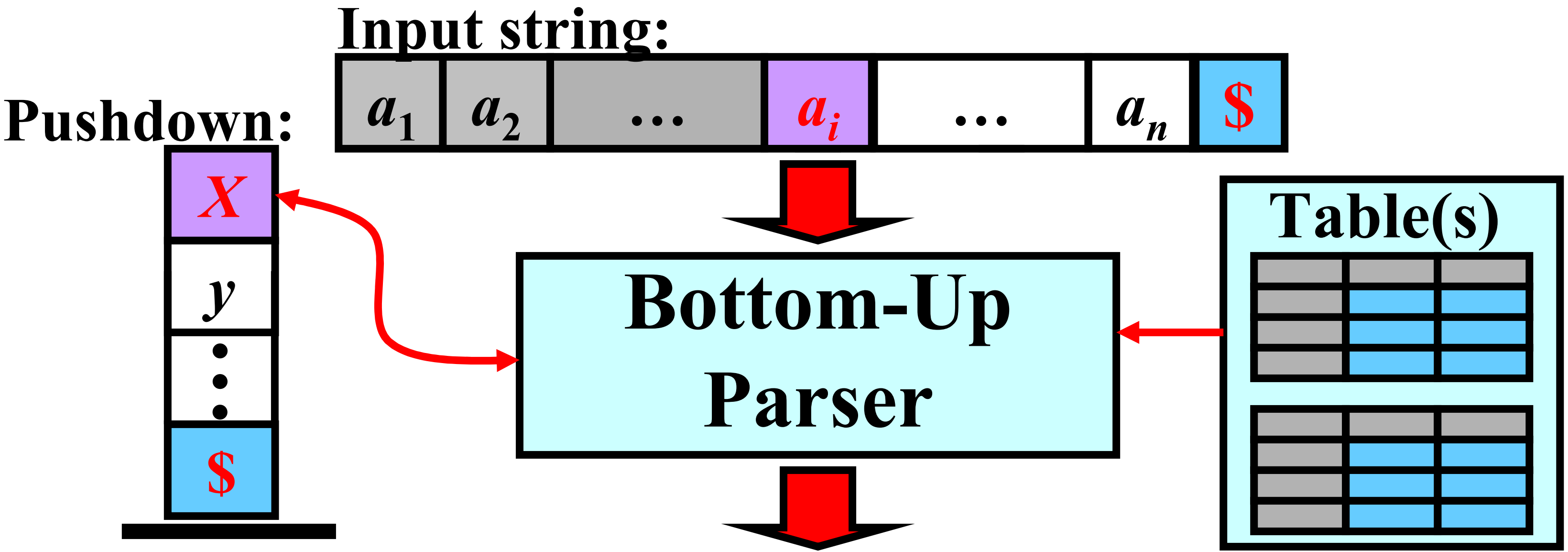
- the least powerful, but simple & easy-to-make

---

2) LR parser

- the most powerful

- **Model of Bottom-Up parser:**



Input string:

a_1 | a_2 | … | a_i | … | a_n | $

Pushdown:

X

y

⋮

$

Bottom-Up Parser

Table(s)

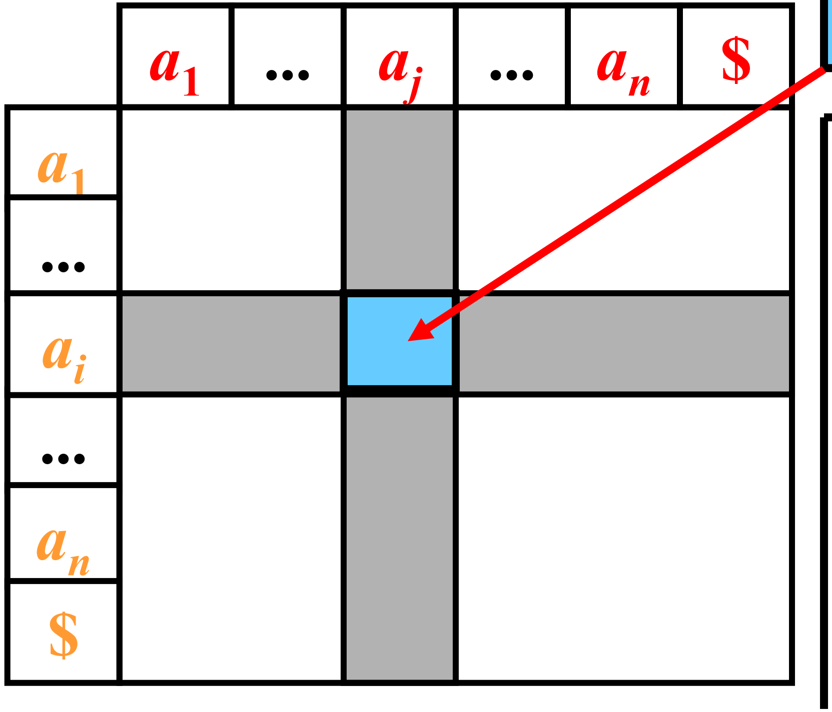
Right parse = **reverse** sequence of rules used in the **rightmost derivation** of the tokenized source program

Operator-Precedence Parser

- No two distinct nonterminals have the same handle
- No ε-rules.

- Let $G = (N, T, P, S)$ be CFG, where $T = \{a_1, a_2, \dots, a_n\}$

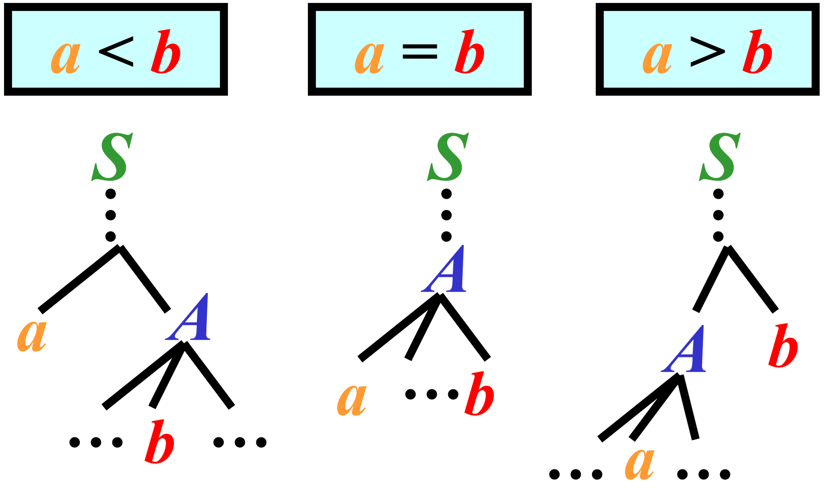
Precedence-table:



	a_1	…	a_j	…	a_n	\$
a_1						
…						
a_i						
…						
a_n						
\$						

Table[a_i , a_j] $\in$ {<, =, >, *blank*}

Illustration of meaning of <, =, >:



Operator-Precedence Parser: Algorithm

- **Input:** Precedence-table for $G = (N, T, P, S)$; $x \in T^*$
- **Output:** Right parse of x if $x \in L(G)$; otherwise, error

---

- **Method:**

```
• Push $ onto the pushdown;
• repeat
    • let a = the current token and
          b = the topmost terminal on the pushdown
    • case Table[b, a] of:
        • = : push(a) & read next a from input string
        • < : replace b with b< on the pushdown &
              push(a) & read next a from input string
        • > : if <y is the pushdown top string and r: A → y ∈ P
              then replace <y with A & write r to output
              else error
        • blank : error
  until a = $ and b = $
• success
```

Operator-Precedence Parser: Example

$G_{expr2} = (N, T, P, E)$, where $N = \{E\}$, $T = \{i, +, *, (,)\}$,

$P = \{\ 1: E \rightarrow E+E,\ 2: E \rightarrow E*E,\ 3: E \rightarrow (E),\ 4: E \rightarrow i\}$

Precedence-table for G_{expr2}:

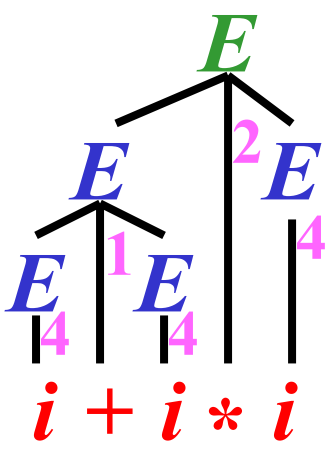
Input token

Pushdown topmost token

	+	*	(	)	i	$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(	<	<	<	=	<	
)	>	>		>		>
i	>	>		>		>
$	<	<	<		<	

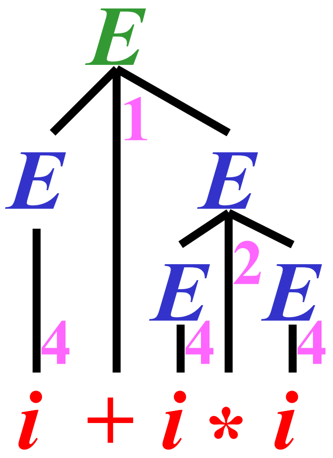
Note: Operator associativity and precedence rules underlie the precedence table:

☹ **Wrong tree:**



Right parse: 44142

☺ **Right tree:**



Right parse: 44421

Operator-Precedence Parsing: Example

	+	*	(	)	*i*	$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(	<	<	<	=	<	
)	>	>		>		>
i	>	>		>		>
$	<	<	<		<	

Rules:

1: *E* → *E*+*E*

2: *E* → *E***E*

3: *E* → (*E*)

4: *E* → *i*

Input string: *i* + *i* * *i* $

Pushdown	Op	Input	Rule
$	<	*i*+*i***i*$	
$<*i*	>	+*i***i*$	4: *E* → *i*
$*E*	<	+*i***i*$	
$<*E*+	<	*i***i*$	
$<*E*+<*i*	>	**i*$	4: *E* → *i*
$<*E*+*E*	<	**i*$	
$<*E*+<*E**	<	*i*$	
$<*E*+<*E**<*i*	>	$	4: *E* → *i*
$<*E*+<*E***E*	>	$	2: *E* → *E***E*
$<*E*+*E*	>	$	1: *E* → *E*+*E*
$*E*		$	

Success

Right parse: 44421

Construction of Precedence Table 1/5

- Let $G_{expr} = (N, T, P, E)$, where $N = \{E\}$,

$T = \{(,), \mathbf{id}_1, \mathbf{id}_2, \ldots, \mathbf{id}_m, \mathbf{op}_1, \mathbf{op}_2, \ldots \mathbf{op}_n\}$,

$P = \{ E \rightarrow (E), E \rightarrow \mathbf{id}_1, E \rightarrow \mathbf{id}_2, \ldots, E \rightarrow \mathbf{id}_m,$
$E \rightarrow E\ \mathbf{op}_1\ E, E \rightarrow E\ \mathbf{op}_2\ E, \ldots, E \rightarrow E\ \mathbf{op}_n\ E \}$

Note: $\mathbf{id}_1, \mathbf{id}_2, \ldots, \mathbf{id}_m$ are identifiers,
$\mathbf{op}_1, \mathbf{op}_2, \ldots \mathbf{op}_n$ are different operators

---

1) Precedence of operators:

- If $\mathbf{op}_i$ has **<u>higher precedence</u>** than $\mathbf{op}_j$ then

$$\boxed{\mathbf{op}_i > \mathbf{op}_j \text{ and } \mathbf{op}_j < \mathbf{op}_i}$$

Example: Precedence-table part derived from the precedence of operators in G_{expr2}

$* > +$
$+ < *$

	+	*
+		<
*	>	

Construction of Precedence Table 2/5

2) Associativity:

Note:

- $\mathbf{op}_i$ is left-associative ⇔ $a\ \mathbf{op}_i\ b\ \mathbf{op}_i\ c = (a\ \mathbf{op}_i\ b)\ \mathbf{op}_i\ c$
- $\mathbf{op}_i$ is right-associative ⇔ $a\ \mathbf{op}_i\ b\ \mathbf{op}_i\ c = a\ \mathbf{op}_i\ (b\ \mathbf{op}_i\ c)$

- Let $\mathbf{op}_i$ and $\mathbf{op}_j$ have **equal precedence**
  - If $\mathbf{op}_i$ and $\mathbf{op}_j$ are **left associative** then

$$\mathbf{op}_i > \mathbf{op}_j \text{ and } \mathbf{op}_j > \mathbf{op}_i$$

  - If $\mathbf{op}_i$ and $\mathbf{op}_j$ are **right associative** then

$$\mathbf{op}_i < \mathbf{op}_j \text{ and } \mathbf{op}_j < \mathbf{op}_i$$

Example: Precedence-table part derived from the associativity of operators in G_{expr2}

+ is left-associative

*** is left-associative**

	+	*
+	>	
*		>

Construction of Precedence Table 3/5

3) Identifiers:

- If $a \in T$ may **precede** $\mathbf{id}_i$, then $a < \mathbf{id}_i$
- If $a \in T$ may **follow** $\mathbf{id}_i$, then $\mathbf{id}_i > a$

Example: Precedence-table part for identifiers

$i * (i + i) * i

$, (, +, * may precede *i*

i * (*i* + *i*) * *i* $

*, +,), $ may follow *i*

	+	*	(	)	*i*	$
+					<	
*					<	
(					<	
)						
i	>	>		>		>
$					<	

Construction of Precedence Table 4/5

4) Parentheses:

- A pair of parentheses: $(=)$
- Let $a \in T - \{), \$\}$. Then, $(< a$
- Let $a \in T - \{(, \$\}$. Then, $a >)$
- Let $a \in T$ and a may **precede** (. Then, $a < ($
- Let $a \in T$ and a may **follow**). Then, $) > a$

Example: Precedence-table part for parentheses.

$\$(i + ((i * (i + (i + i)))))$

↓ ↓ ↓ ↓

$, (, *, + may precede (

$(((((i + i) + i) * i)) + i)\$$

↓ ↓ ↓ ↓

+, *,), $ may follow)

	+	*	(	)	i	$
+			<	>		
*			<	>		
(	<	<	<	=	<	
)	>	>		>		>
i				>		
$			<			

Construction of Precedence Table 5/5

5) End Marker $\$$

- Let $\mathbf{op}_i$ be any operator. Then:

$\$ < \mathbf{op}_i$ and $\mathbf{op}_i > \$$

Example: Precedence-table part for end-markers.

$\$ < +$
$\$ < *$
$+ > \$$
$* > \$$

	+	*	$
+			>
*			>
$	<	<	

Summary:

	+	*	(	)	*i*	$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(	<	<	<	=	<	
)	>	>		>		>
i	>	>		>		>
$	<	<	<		<	

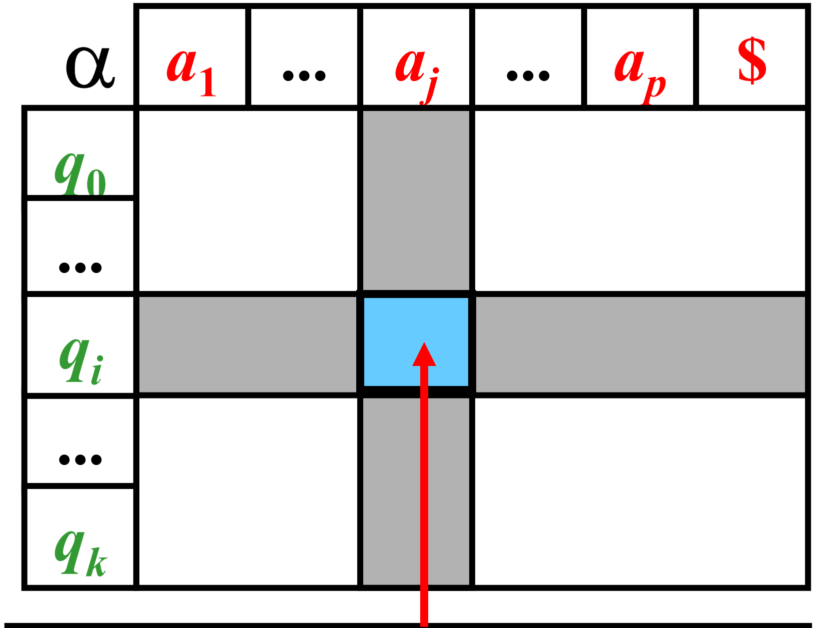
LR-Parser

- Let $G = (N, T, P, S)$ be a CFG, where $N = \{A_1, A_2, \dots, A_n\}$, $T = \{a_1, a_2, \dots, a_m\}$

- LR-parser is a EPDA, M, with states $Q = \{q_0, q_1, \dots, q_k\}$, where q_0 is the start state.
- M is based on LR table that has these two parts
  1) **Action part**
  2) **Go-to part**

Action Part & Go-to Part

Action Part:



$\alpha[q_i , a_j]$ = **1** or **2** or **3** or **4**

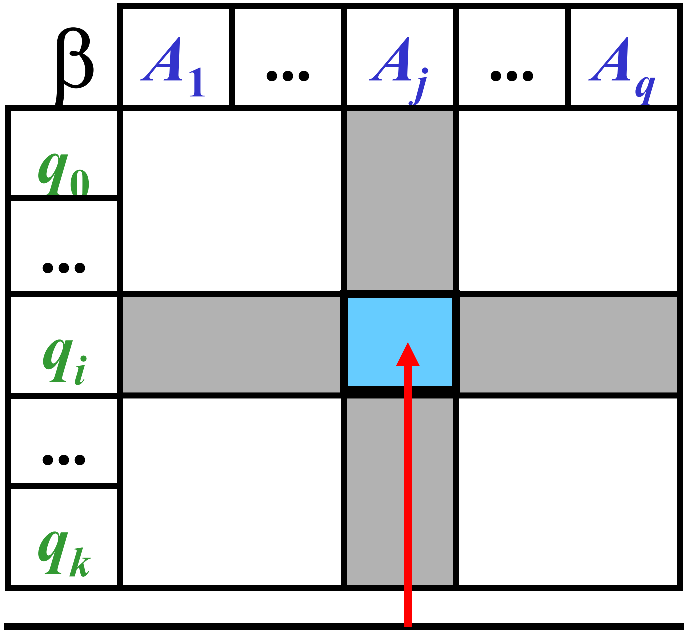
1) **sq**: **s** = **s**hift, $q \in Q$

2) **rp**: **r** = **r**educe, $p \in P$

3) ☺ : **success**

4) **blank**: **error**

Go-to Part:



$\beta[q_i , A_j]$ = **1** or **2**

1) q: $q \in Q$

2) **blank**

LR-Parser: Algorithm

- **Input:** LR-table for $G=(N, T, P, S)$; $x \in T^*$
- **Output:** Right parse of x if $x \in L(G)$; otherwise, error

---

- **Method:**

```
• push(<$, $q_0$>) onto pushdown; state := $q_0$;
• repeat
      • let a = the current token
      case $\alpha$[state, a] of:
          • sq: push(<a, q>) & read next a from input string &
                state := q;
          • rp: if p: $A \to X_1X_2\dots X_n \in P$ and
                <?, q><$X_1$,?><$X_2$,?>...<$X_n$,?> is pushdown top
                then state := $\beta$[q, A] &
                replace <$X_1$,?><$X_2$,?>...<$X_n$, ?> with <A, state>
                on the pushdown & write r to output
                else error
          • ☺: success
          • blank: error
  until success or error
```

!

LR-Parser: Example 1/2

$G_{expr1} = (N, T, P, E)$, where $N = \{E, F, T\}$, $T = \{i, +, *, (,)\}$,
$P = \{$ 1: $E \rightarrow E+T$, 2: $E \rightarrow T$, 3: $T \rightarrow T*F$,
4: $T \rightarrow F$, 5: $F \rightarrow (E)$, 6: $F \rightarrow i$ $\}$

LR-table for G_{expr1}:

α	*i*	+	*	(	)	$
0	s5			s4		
1		s6				☺
2		r2	s7		r2	r2
3		r4	r4		r4	r4
4	s5			s4		
5		r6	r6		r6	r6
6	s5			s4		
7	s5			s4		
8		s6			s11	
9		r1	s7		r1	r1
10		r3	r3		r3	r3
11		r5	r5		r5	r5

Action part for G_{expr1}

Go-to part for G_{expr1}

β	*E*	*T*	*F*
0	1	2	3
1			
2			
3			
4	8	2	3
5			
6		9	3
7			10
8			
9			
10			
11			

LR-Parser: Example 2/2

Rules: 1: *E* → *E*+*T*, 2: *E* → *T*, 3: *T* → *T***F*,
4: *T* → *F*, 5: *F* → (*E*), 6: *F* → *i*

Input string: *i* * *i* $

Pushdown	St.	Input	Enter	Rule
<$,0>	0	*i***i*$	α[0, *i*] = s5	
<$,0><*i*,5>	5	**i*$	α[5, *] = r6	6: *F* → *i*
			β[0, *F*] = 3	
<$,0><*F*,3>	3	**i*$	α[3, *] = r4	4: *T* → *F*
			β[0, *T*] = 2	
<$,0><*T*,2>	2	**i*$	α[2, *] = s7	
<$,0><*T*,2><*,7>	7	*i*$	α[2, *i*] = s5	
<$,0><*T*,2><*,7><*i*,5>	5	$	α[5, $] = r6	6: *F* → *i*
			β[7, *F*] = 10	
<$,0><*T*,2><*,7><*F*,10>	10	$	α[10, $]= r3	3: *T* → *T***F*
			β[0, *T*] = 2	
<$,0><*T*,2>	2	$	α[2, $] = r2	2: *E* → *T*
			β[0, *E*] = 1	
<$,0><*E*,1>	1	$	α[1, $] = ☺	

Success

Right parse: 64632

Construction of LR Table: Introduction

- **One parsing algorithm but many algorithms for the construction of LR table.**

Basic algorithms for the construction of LR table:

1) **Simple LR (SLR)**: the least powerful, but simple and few states

2) **Canonical LR**: more powerful, but many states

3) **Lookahead LR (LALR)**: the best because the most powerful and the same number of states as SLR

Extended Grammar with a "Dummy Rule"

Gist: Grammar with special "starting rule"

Definition: Let $G = (N, T, P, S)$ be a CFG, $S' \notin N$. *Extended grammar* for G is grammar $G' = (N \cup \{S'\}, T, P \cup \{S' \rightarrow S\}, S')$.

Why a dummy rule? When $S' \rightarrow S$ is used and the input token is endmarker, then **syntax analysis is successfully completed**.

Example:

$G_{expr1} = (N, T, P, E)$, where $N = \{E, F, T\}$, $T = \{i, +, *, (,)\}$,

$P = \{$ 1: $E \rightarrow E+T$, 2: $E \rightarrow T$, 3: $T \rightarrow T*F$,
4: $T \rightarrow F$, 5: $F \rightarrow (E)$, 6: $F \rightarrow i$ $\}$

Extended grammar for G_{expr1}:

$G'_{expr1} = (N, T, P, E')$, where $N = \{E', E, F, T\}$, $T = \{i, +, *, (,)\}$,

$P = \{$ 0: $E' \rightarrow E$, 1: $E \rightarrow E+T$, 2: $E \rightarrow T$, 3: $T \rightarrow T*F$,
4: $T \rightarrow F$, 5: $F \rightarrow (E)$, 6: $F \rightarrow i$ $\}$

Construction of LR Table: Items

Gist: Item is a rule of CFG with • in the right side of rule.

Definition: Let $G = (N, T, P, S)$ be a CFG, $A \to \boldsymbol{x} \in P$, $\boldsymbol{x} = \boldsymbol{yz}$. Then, $A \to \boldsymbol{y} \bullet \boldsymbol{z}$ is an *item*.

Example: Consider $\boldsymbol{E} \to \boldsymbol{E+T}$

All items for $\boldsymbol{E} \to \boldsymbol{E+T}$ are:

$\boldsymbol{E} \to \bullet\boldsymbol{E+T}$, $\boldsymbol{E} \to \boldsymbol{E}\bullet\boldsymbol{+T}$, $\boldsymbol{E} \to \boldsymbol{E+}\bullet\boldsymbol{T}$, $\boldsymbol{E} \to \boldsymbol{E+T}\bullet$

Meaning: $A \to y\bullet z$ means that if y appears on the pushdown top and a prefix of the input is eventually reduced to z, then yz $(= \boldsymbol{x})$ as a handle can be reduced to A according to $A \to \boldsymbol{x}$.

Closure of Item: Algorithm

Note: Closure(I) is the set of items defined by the following algorithm:

- **Input:** $G = (N, T, P, S)$; item I
- **Output:** $Closure(I)$

- **Method:**
- $Closure(I) := \{I\}$;
- **Apply the following rule until** $Closure(I)$ **cannot be changed:**
- **if** $A \rightarrow y\bullet Bz \in Closure(I)$ **and** $B \rightarrow x \in P$ **then** add $B \rightarrow \bullet x$ to $Closure(I)$

Closure of Item: Example 1/2

$G'_{expr1} = (N, T, P, E')$, where $N = \{E', E, F, T\}$, $T = \{i, +, *, (,)\}$,
$P = \{$ 0: $E' \to E$, 1: $E \to E+T$, 2: $E \to T$, 3: $T \to T*F$,
4: $T \to F$, 5: $F \to (E)$, 6: $F \to i$ $\}$

Task: $Closure(I)$ for $I = E' \to \bullet E$

$Closure(I) := \{E' \to \bullet E\}$

---

1) $E' \to \bullet E \in Closure(I)$ **and** $E \to E+T \in P$**:**
add $E \to \bullet E+T$ **to** $Closure(I)$

$Closure(I) = \{E' \to \bullet E, E \to \bullet E+T\}$

---

2) $E' \to \bullet E \in Closure(I)$ **and** $E \to T \in P$**:**
add $E \to \bullet T$ **to** $Closure(I)$

$Closure(I) = \{E' \to \bullet E, E \to \bullet E+T, E \to \bullet T\}$

---

3) $E \to \bullet T \in Closure(I)$ **and** $T \to T*F \in P$**:**
add $T \to \bullet T*F$ **to** $Closure(I)$

$Closure(I) = \{E' \to \bullet E, E \to \bullet E+T, E \to \bullet T, T \to \bullet T*F\}$

Closure of Item: Example 2/2

$G'_{expr1} = (N, T, P, E')$, where $N = \{E', E, F, T\}$, $T = \{i, +, *, (,)\}$,
$P = \{$ 0: $E' \rightarrow E$, 1: $E \rightarrow E+T$, 2: $E \rightarrow T$, 3: $T \rightarrow T*F$,
4: $T \rightarrow F$, 5: $F \rightarrow (E)$, 6: $F \rightarrow i$ $\}$

---

4) $E \rightarrow \bullet T \in Closure(I)$ **and** $T \rightarrow F \in P$:
add $T \rightarrow \bullet F$ **to** $Closure(I)$

$Closure(I) = \{E' \rightarrow \bullet E, E \rightarrow \bullet E+T, E \rightarrow \bullet T, T \rightarrow \bullet T*F,$
$T \rightarrow \bullet F \}$

---

5) $T \rightarrow \bullet F \in Closure(I)$ **and** $F \rightarrow (E) \in P$:
add $F \rightarrow \bullet(E)$ **to** $Closure(I)$

$Closure(I) = \{E' \rightarrow \bullet E, E \rightarrow \bullet E+T, E \rightarrow \bullet T, T \rightarrow \bullet T*F,$
$T \rightarrow \bullet F, F \rightarrow \bullet(E) \}$

---

6) $T \rightarrow \bullet F \in Closure(I)$ **and** $F \rightarrow i \in P$:
add $F \rightarrow \bullet i$ **to** $Closure(I)$

Summary:
$Closure(I) = \{E' \rightarrow \bullet E, E \rightarrow \bullet E+T, E \rightarrow \bullet T, T \rightarrow \bullet T*F,$
$T \rightarrow \bullet F, F \rightarrow \bullet(E), F \rightarrow \bullet i \}$

Set Θ_G for Grammar G 1/2

Gist: Θ_G is the set of all prefixes of the right-hand sides of rules from G.

Definition: Let $G = (N, T, P, S)$ be CFG.

$$\Theta_G = \{\langle y \rangle: A \rightarrow y \bullet z \text{ is an item in } G\}$$

Example:

$G'_{expr1} = (N, T, P, E')$, where $N = \{E', E, F, T\}$, $T = \{i, +, *, (,)\}$,
$P = \{$ 0: $E' \rightarrow E$, 1: $E \rightarrow E+T$, 2: $E \rightarrow T$, 3: $T \rightarrow T*F$, 4: $T \rightarrow F$, 5: $F \rightarrow (E)$, 6: $F \rightarrow i$ $\}$

Task: $\Theta_{G'expr1}$

1) Members of $\Theta_{G'expr1}$ of length 0: $\langle\varepsilon\rangle \in \Theta_{G'expr1}$

2) Members of $\Theta_{G'expr1}$ of length 1:

$$\underbrace{E' \rightarrow \underline{E},\ E \rightarrow \underline{E}+T}_{\langle E\rangle \in \Theta_{G'expr1}},\ \underbrace{E \rightarrow \underline{T},\ T \rightarrow \underline{T}*F}_{\langle T\rangle \in \Theta_{G'expr1}},\ \underbrace{T \rightarrow \underline{F},\ F \rightarrow \underline{(}E),\ F \rightarrow \underline{i}}_{\langle F\rangle, \langle(\rangle, \langle i\rangle \in \Theta_{G'expr1}}$$

Set Θ_G for Grammar G 2/2

$G'_{expr1} = (N, T, P, E')$, where $N = \{E', E, F, T\}$, $T = \{i, +, *, (,)\}$,

$P = \{$ 0: $E' \to E$, 1: $E \to E+T$, 2: $E \to T$, 3: $T \to T*F$,

4: $T \to F$, 5: $F \to (E)$, 6: $F \to i$ $\}$

---

3) Members of $\Theta_{G'expr1}$ of length 2:

$$E' \to E,\ \underbrace{E \to \underline{E+}T}_{\langle E+\rangle \in \Theta_{G'expr1}},\ E \to \underline{T},\ \underbrace{T \to \underline{T*}F}_{\langle T*\rangle \in \Theta_{G'expr1}},\ T \to F,\ \underbrace{F \to \underline{(E}\,)}_{\langle (E\rangle \in \Theta_{G'expr1}},\ F \to i$$

---

4) Members of $\Theta_{G'expr1}$ of length 3:

$$E' \to E,\ \underbrace{E \to \underline{E+T}}_{\langle E+T\rangle \in \Theta_{G'expr1}},\ E \to \underline{T},\ \underbrace{T \to \underline{T*F}}_{\langle T*F\rangle \in \Theta_{G'expr1}},\ T \to F,\ \underbrace{F \to \underline{(E)}}_{\langle (E)\rangle \in \Theta_{G'expr1}},\ F \to i$$

Summary:

$$\Theta_{G'expr1} = \{\ \langle\varepsilon\rangle, \langle E\rangle, \langle T\rangle, \langle F\rangle, \langle (\rangle, \langle i\rangle, \langle E+\rangle, \langle T*\rangle, \langle (E\rangle, \langle E+T\rangle, \langle T*F\rangle, \langle (E)\rangle\ \}$$

Contents(x): Algorithm

Note: For all $x \in \Theta_G$, *Contents*(x) is the set of items defined by the following algorithm:

- **Input:** Extended $G = (N, T, P, S')$; Θ_G
- **Output:** *Contents*(x) for all $x \in \Theta_G$

- **Method:**
- *Contents*($<\varepsilon>$) := *Closure*($S' \to \bullet S$);
- for each $x \in \Theta_G - \{<\varepsilon>\}$: *Contents*($x$) := $\varnothing$
- **Apply the following rule until no *Contents* set can be changed:**

  if $A \to y\bullet Xz \in$ *Contents*($<x>$), where $X \in N \cup T$
  and $<xX> \in \Theta_G$ **then**
  add *Closure*($A \to yX\bullet z$) to *Contents*($<xX>$)

Contents(*x*): Example 1/9

$G'_{expr1} = (N, T, P, E')$, where $N = \{E', E, F, T\}$, $T = \{i, +, *, (,)\}$,
$P = \{$ 0: $E' \rightarrow E$, 1: $E \rightarrow E+T$, 2: $E \rightarrow T$, 3: $T \rightarrow T*F$,
4: $T \rightarrow F$, 5: $F \rightarrow (E)$, 6: $F \rightarrow i$ $\}$

$\Theta_{G'expr1} = \{$ $<\varepsilon>$, $<E>$, $<T>$, $<F>$, $<(>$, $<i>$, $<E+>$,
$<T*>$, $<(E>$, $<E+T>$, $<T*F>$, $<(E)>$ $\}$

---

0) $Contents(<\varepsilon>) := Closure(E' \rightarrow \bullet E) =$
$\{E' \checkmark \rightarrow \bullet E,\ E \checkmark \rightarrow \bullet E+T,\ E \checkmark \rightarrow \bullet T,\ T \rightarrow \bullet T*F,\ T \rightarrow \bullet F,\ F \rightarrow \bullet (E),\ F \rightarrow \bullet i\ \}$

$E' \rightarrow \bullet E \in Contents(<\varepsilon>)$ & $<\varepsilon E> = <E> \in \Theta_{G'expr1}$:
add $Closure(E' \rightarrow E\bullet) = \{E' \rightarrow E\bullet\}$ **to** $Contents(<E>)$

$E \rightarrow \bullet E+T \in Contents(<\varepsilon>)$ & $<\varepsilon E> = <E> \in \Theta_{G'expr1}$:
add $Closure(E \rightarrow E\bullet+T) = \{E \rightarrow E\bullet+T\}$ **to** $Contents(<E>)$

$E \rightarrow \bullet T \in Contents(<\varepsilon>)$ & $<\varepsilon T> = <T> \in \Theta_{G'expr1}$:
add $Closure(E \rightarrow T\bullet) = \{E \rightarrow T\bullet\}$ **to** $Contents(<T>)$

$Contents(x)$: Example 2/9

⋮

$Contents(<\varepsilon>) =$

$\{E'\checkmark \rightarrow \bullet E,\ E\checkmark \rightarrow \bullet E+T,\ E\checkmark \rightarrow \bullet T,\ T\checkmark \rightarrow \bullet T{*}F,\ T\checkmark \rightarrow \bullet F,\ F\checkmark \rightarrow \bullet(E),\ F\checkmark \rightarrow \bullet i\ \}$

$T \rightarrow \bullet T{*}F \in Contents(<\varepsilon>)$ & $<\varepsilon T> = <T> \in \Theta_{G'expr1}$:
add $Closure(T \rightarrow T\bullet{*}F) = \{T \rightarrow T\bullet{*}F\}$ **to** $Contents(<T>)$

$T \rightarrow \bullet F \in Contents(<\varepsilon>)$ & $<\varepsilon F> = <F> \in \Theta_{G'expr1}$:
add $Closure(T \rightarrow F\bullet) = \{T \rightarrow F\bullet\}$ **to** $Contents(<F>)$

$F \rightarrow \bullet(E) \in Contents(<\varepsilon>)$ & $<\varepsilon(\ > = <\ (\ > \in \Theta_{G'expr1}$:
add $Closure(F \rightarrow (\bullet E)) = \{F \rightarrow (\bullet E),\ E \rightarrow \bullet E+T,\ E \rightarrow \bullet T,$
$T \rightarrow \bullet T{*}F,\ T \rightarrow \bullet F,\ F \rightarrow \bullet(E),\ F \rightarrow \bullet i\ \}$ **to** $Contents(<\ (\ >)$

$F \rightarrow \bullet i \in Contents(<\varepsilon>)$ & $<\varepsilon i> = <i> \in \Theta_{G'expr1}$:
add $Closure(F \rightarrow i\bullet) = \{F \rightarrow i\bullet\}$ **to** $Contents(<i>)$

Contents(*x*): Example 3/9

✓ $Contents(<\varepsilon>) = \{E' \to \bullet E, E \to \bullet E+T, E \to \bullet T, T \to \bullet T*F, T \to \bullet F, F \to \bullet (E), F \to \bullet i\}$

$Contents(<E>) = \{E' \to E\bullet, E \to E\bullet+T\}$

$Contents(<T>) = \{E \to T\bullet, T \to T\bullet*F\}$

$Contents(<F>) = \{T \to F\bullet\}$

$Contents(<(>) = \{F \to (\bullet E), E \to \bullet E+T, E \to \bullet T, T \to \bullet T*F, T \to \bullet F, F \to \bullet (E), F \to \bullet i\}$

$Contents(<i>) = \{F \to i\bullet\}$

$Contents(<E+>) = \emptyset$

$Contents(<T*>) = \emptyset$

$Contents(<(E>) = \emptyset$

$Contents(<E+T>) = \emptyset$

$Contents(<T*F>) = \emptyset$

$Contents(<(E)>) = \emptyset$

Contents(*x*): Example 4/9

1) *Contents*(<E>) = $\{E'\checkmark \rightarrow E\bullet,\ E\checkmark \rightarrow E\bullet+T\}$:

$E \rightarrow E\bullet$: *nothing*

$E \rightarrow E\bullet+T \in$ *Contents*(<E>) & <$E+$> $\in \Theta_{G'expr1}$:
add ***Closure***($E \rightarrow E+\bullet T$) = $\{E \rightarrow E+\bullet T,\ T \rightarrow \bullet T*F,\ T \rightarrow \bullet F,$
$F \rightarrow \bullet(E),\ F \rightarrow \bullet i\}$ **to** ***Contents***(<$E+$>)

---

2) *Contents*(<T>) = $\{E\checkmark \rightarrow T\bullet,\ T\checkmark \rightarrow T\bullet*F\}$:

$E \rightarrow T\bullet$: *nothing*

$T \rightarrow T\bullet*F \in$ *Contents*(<T>) & <$T*$> $\in \Theta_{G'expr1}$:
add ***Closure***($T \rightarrow T*\bullet F$) = $\{T \rightarrow T*\bullet F,\ F \rightarrow \bullet(E),\ F \rightarrow \bullet i\}$
to ***Contents***(<$T*$>)

---

3) *Contents*(<F>) = $\{T\checkmark \rightarrow F\bullet\}$:

$T \rightarrow F\bullet$: *nothing*

Contents(x): Example 5/9

4) $\mathit{Contents}(<\,(\,>) =$

$\{F \checkmark\!\to (\bullet E),\ E \checkmark\!\to \bullet E+T,\ E \checkmark\!\to \bullet T,\ T \checkmark\!\to \bullet T*F,\ T \checkmark\!\to \bullet F,\ F \checkmark\!\to \bullet(E),\ F \checkmark\!\to \bullet i\ \}$

$F \to (\bullet E) \in \mathit{Contents}(<\,(\,>)\ \&\ <\,(E\,> \in \Theta_{G'expr1}$:

add $\mathbf{Closure}(F \to (E\bullet)) = \{F \to (E\bullet)\}$ **to** $\mathbf{Contents}(<\,(E\,>)$

$F \to \bullet E+T \in \mathit{Contents}(<\,(\,>)\ \&\ <\,(E\,> \in \Theta_{G'expr1}$:

add $\mathbf{Closure}(F \to E\bullet+T) = \{F \to E\bullet+T\}$ **to** $\mathbf{Contents}(<\,(E\,>)$

$E \to \bullet T \in \mathit{Contents}(<\,(\,>)$ **but** $<\,(T\,> \notin \Theta_{G'expr1}$: *nothing*

$T \to \bullet T*F \in \mathit{Contents}(<\,(\,>)$ **but** $<\,(T\,> \notin \Theta_{G'expr1}$: *nothing*

$T \to \bullet F \in \mathit{Contents}(<\,(\,>)$ **but** $<\,(F\,> \notin \Theta_{G'expr1}$: *nothing*

$F \to \bullet(E) \in \mathit{Contents}(<\,(\,>)$ **but** $<\,((\,> \notin \Theta_{G'expr1}$: *nothing*

$T \to \bullet i \in \mathit{Contents}(<\,(\,>)$ **but** $<\,(i\,> \notin \Theta_{G'expr1}$: *nothing*

---

5) $\mathit{Contents}(<i>) = \{F \checkmark\!\to \bullet i\ \}$:

$F \to i\bullet$: *nothing*

Contents(*x*): Example 6/9

✓ $Contents(<\varepsilon>) = \{E' \to \bullet E, E \to \bullet E+T, E \to \bullet T, T \to \bullet T*F, T \to \bullet F, F \to \bullet(E), F \to \bullet i\}$

✓ $Contents(<E>) = \{E' \to E\bullet, E \to E\bullet+T\}$

✓ $Contents(<T>) = \{E \to T\bullet, T \to T\bullet*F\}$

✓ $Contents(<F>) = \{T \to F\bullet\}$

✓ $Contents(<(>) = \{F \to (\bullet E), E \to \bullet E+T, E \to \bullet T, T \to \bullet T*F, T \to \bullet F, F \to \bullet(E), F \to \bullet i\}$

✓ $Contents(<i>) = \{F \to i\bullet\}$

$Contents(<E+>) = \{E \to E+\bullet T, T \to \bullet T*F, T \to \bullet F, F \to \bullet(E), F \to \bullet i\}$

$Contents(<T*>) = \{T \to T*\bullet F, F \to \bullet(E), F \to \bullet i\}$

$Contents(<(E>) = \{F \to (E\bullet), E \to E\bullet+T\}$

$Contents(<E+T>) = \emptyset$

$Contents(<T*F>) = \emptyset$

$Contents(<(E)>) = \emptyset$

Contents(*x*): Example 7/9

6) $\mathit{Contents}(<E+>) =$
$\{E\checkmark \rightarrow E+\bullet T,\ T\checkmark \rightarrow \bullet T*F,\ T\checkmark \rightarrow \bullet F,\ F\checkmark \rightarrow \bullet(E),\ F\checkmark \rightarrow \bullet i\ \}$

$E \rightarrow E+\bullet T \in \mathit{Contents}(<E+>)\ \&\ <E+T> \in \Theta_{G'expr1}$:
add $\boldsymbol{Closure}(E \rightarrow E+T\bullet) = \{E \rightarrow E+T\bullet\}$ **to** $\boldsymbol{Contents}(<E+T>)$

$T \rightarrow \bullet T*F \in \mathit{Contents}(<E+>)\ \&\ <E+T> \in \Theta_{G'expr1}$:
add $\boldsymbol{Closure}(T \rightarrow T\bullet *F\ ) = \{T \rightarrow T\bullet *F\}$ **to** $\boldsymbol{Contents}(<E+T>)$

$T \rightarrow \bullet F \in \mathit{Contents}(<E+>)$ **but** $<E+F\ > \notin \Theta_{G'expr1}$: *nothing*
$F \rightarrow \bullet(E) \in \mathit{Contents}(<E+>)$ **but** $<E+\ (> \notin \Theta_{G'expr1}$: *nothing*
$T \rightarrow \bullet i \in \mathit{Contents}(<E+>)$ **but** $<E+\ i> \notin \Theta_{G'expr1}$: *nothing*

---

7) $\mathit{Contents}(<T*>) = \{T\checkmark \rightarrow T*\bullet F,\ F\checkmark \rightarrow \bullet(E),\ F\checkmark \rightarrow \bullet i\ \}$
$T \rightarrow T*\bullet F \in \mathit{Contents}(<T*>)\ \&\ <T*F> \in \Theta_{G'expr1}$:
add $\boldsymbol{Closure}(T \rightarrow T*F\bullet) = \{T \rightarrow T*F\bullet\}$ **to** $\boldsymbol{Contents}(<T*F>)$
$F \rightarrow \bullet(E) \in \mathit{Contents}(<T*>)$ **but** $<T*\ (> \notin \Theta_{G'expr1}$: *nothing*
$T \rightarrow \bullet i \in \mathit{Contents}(<T*>)$ **but** $<T*\ i> \notin \Theta_{G'expr1}$: *nothing*

Contents(*x*): Example 8/9

8) $\mathit{Contents}(<(E>) = \{F\checkmark \rightarrow (E\bullet),\ E\checkmark \rightarrow E\bullet+T\}$

$F \rightarrow (E\bullet) \in \mathit{Contents}(<(E>)\ \&\ <(E)> \in \Theta_{G'expr1}$:
add $\mathit{Closure}(E \rightarrow (E)\bullet) = \{F \rightarrow (E)\bullet\}$ **to** $\mathit{Contents}(<(E)>)$

$E \rightarrow E\bullet+T \in \mathit{Contents}(<(E>)$ **but** $<(E+> \notin \Theta_{G'expr1}$: *nothing*

---

9) $\mathit{Contents}(<E+T>) = \{E\checkmark \rightarrow E+T\bullet,\ T\checkmark \rightarrow T\bullet*F\}$

$E \rightarrow E+T\bullet$: *nothing*

$T \rightarrow T\bullet*F \in \mathit{Contents}(<E+T>)$ **but** $<E+T*> \notin \Theta_{G'expr1}$: *nothing*

---

10) $\mathit{Contents}(<E+T>) = \{T\checkmark \rightarrow T*F\bullet\}$

$T \rightarrow T*F\bullet$: *nothing*

---

11) $\mathit{Contents}(<(E)>) = \{F\checkmark \rightarrow (E)\bullet\}$

$F \rightarrow (E)\bullet$: *nothing*

Contents(*x*): Example 9/9

✓ $Contents(<\varepsilon>) = \{E' \rightarrow \bullet E,\ E \rightarrow \bullet E+T,\ E \rightarrow \bullet T,\ T \rightarrow \bullet T*F,\ T \rightarrow \bullet F,\ F \rightarrow \bullet (E),\ F \rightarrow \bullet i\ \}$

✓ $Contents(<E>) = \{E' \rightarrow E\bullet,\ E \rightarrow E\bullet +T\}$

✓ $Contents(<T>) = \{E \rightarrow T\bullet,\ T \rightarrow T\bullet *F\}$

✓ $Contents(<F>) = \{T \rightarrow F\bullet\}$

✓ $Contents(<(>) = \{F \rightarrow (\bullet E),\ E \rightarrow \bullet E+T,\ E \rightarrow \bullet T,\ T \rightarrow \bullet T*F,\ T \rightarrow \bullet F,\ F \rightarrow \bullet (E),\ F \rightarrow \bullet i\ \}$

✓ $Contents(<i>) = \{F \rightarrow i\bullet\}$

✓ $Contents(<E+>) = \{E \rightarrow E+\bullet T,\ T \rightarrow \bullet T*F,\ T \rightarrow \bullet F,\ F \rightarrow \bullet (E),\ F \rightarrow \bullet i\ \}$

✓ $Contents(<T*>) = \{T \rightarrow T*\bullet F,\ F \rightarrow \bullet (E),\ F \rightarrow \bullet i\ \}$

✓ $Contents(<(E>) = \{F \rightarrow (E\bullet),\ E \rightarrow E\bullet +T\ \}$

✓ $Contents(<E+T>) = \{E \rightarrow E+T\bullet,\ T \rightarrow T\bullet *F\ \}$

✓ $Contents(<T*F>) = \{T \rightarrow T*F\bullet\ \}$

✓ $Contents(<(E)>) = \{F \rightarrow (E)\bullet\ \}$

Construction of LR-table: Algorithm

- **Input:** Extended $G = (N, T, P, S')$; Θ_G;
  $Contents(x)$ for all $x \in \Theta_G$; $Follow(A)$ for all $A \in N$
- **Output:** LR-table for G (α = Action part, β = Go-to part)

- **Method:**

```
• $StatesOfTable := \Theta_G$; $StartingState := <\varepsilon>$
• for each $<x> \in \Theta_G$ do
• for each $I \in Contents(<x>)$ do
    • case $I$ of
        • $I = A \rightarrow y\bullet Xz$, where $X \in N$:
          if $A \rightarrow yX\bullet z \in Contents(<q>)$ then $\beta[<x>, X] := <q>$
        • $I = A \rightarrow y\bullet Xz$, where $X \in T$:
          if $A \rightarrow yX\bullet z \in Contents(<q>)$ then $\alpha[<x>, X] := s<q>$
        • $I = S' \rightarrow S\bullet$: $\alpha[<x>, \$] :=$ ☺
        • $I = A \rightarrow y\bullet$ $(A \neq S')$:
          for each $a \in Follow(A)$ do $\alpha[<x>, a] := rp$,
          where $p$ is a label of rule $A \rightarrow y$
```

Construction of LR-table: Example 1/5

Task: LR-table for G_{expr1}

	α						β		
	i	+	*	(	)	$	*E*	*T*	*F*
<ε>							<*E*>	<*T*>	<*F*>
<*E*>									
<*T*>									
<*F*>									
<(>									
<*i*>									
<*E*+>									
<*T**>									
<(*E*>									
<*E*+*T*>									
<*T***F*>									
<(*E*)>									

Contents(<ε>):

$I = E' \rightarrow \bullet E \in \mathit{Contents}(<\varepsilon>)$:

$E' \rightarrow E\bullet \in \mathit{Contents}(<E>)$: $\beta[<\varepsilon>, E] := <E>$

$I = E \rightarrow \bullet E{+}T \in \mathit{Contents}(<\varepsilon>)$:

$E \rightarrow E\bullet{+}T \in \mathit{Contents}(<E>)$: $\beta[<\varepsilon>, E] := <E>$

$I = E \rightarrow \bullet T \in \mathit{Contents}(<\varepsilon>)$:

$E \rightarrow T\bullet \in \mathit{Contents}(<T>)$: $\beta[<\varepsilon>, T] := <T>$

$I = E \rightarrow \bullet T{*}F \in \mathit{Contents}(<\varepsilon>)$:

$E \rightarrow T\bullet{*}F \in \mathit{Contents}(<T>)$: $\beta[<\varepsilon>, T] := <T>$

$I = E \rightarrow \bullet F \in \mathit{Contents}(<\varepsilon>)$:

$E \rightarrow F\bullet \in \mathit{Contents}(<F>)$: $\beta[<\varepsilon>, F] := <F>$

Construction of LR-table: Example 2/5

Task: LR-table for G_{expr1}

	α						β		
	i	+	*	(	)	$	*E*	*T*	*F*
<ε>	s<*i*>			s<(>			<*E*>	<*T*>	<*F*>
<*E*>									
<*T*>									
<*F*>									
<(>									
<*i*>									
<*E*+>									
<*T**>									
<(*E*>									
<*E*+*T*>									
<*T***F*>									
<(*E*)>									

$\mathit{Contents}(<\varepsilon>)$:

$I = F \rightarrow \bullet(E) \in \mathit{Contents}(<\varepsilon>)$:
$F \rightarrow (\bullet E) \in \mathit{Contents}(<(>)$: $\alpha[<\varepsilon>, (] := s<(>$

$I = F \rightarrow \bullet i \in \mathit{Contents}(<\varepsilon>)$:
$F \rightarrow i\bullet \in \mathit{Contents}(<i>)$: $\alpha[<\varepsilon>, E] := s<i>$

Construction of LR-table: Example 3/5

Task: LR-table for G_{expr1}

	α						β		
	i	+	*	(	)	$	*E*	*T*	*F*
<ε>	s<*i*>			s<(>			<*E*>	<*T*>	<*F*>
<*E*>		s<*E*+>				☺			
<*T*>									
<*F*>									
<(>									
<*i*>									
<*E*+>									
<*T**>									
<(*E*>									
<*E*+*T*>									
<*T***F*>									
<(*E*)>									

Contents(<*E*>):

$I = E' \rightarrow E\bullet \in$ *Contents*(<*E*>): α[<*E*>, $] := ☺

$I = E \rightarrow E\bullet{+}T \in$ *Contents*(<*E*>):

$E \rightarrow E{+}\bullet T \in$ *Contents*(<*E*+>): α[<*E*>, +] = s<*E*+>

Construction of LR-table: Example 4/5

Task: LR-table for G_{expr1}

	α						β		
	i	+	*	(	)	$	*E*	*T*	*F*
<ε>	s<*i*>			s<(>			<*E*>	<*T*>	<*F*>
<*E*>		s<*E*+>				☺			
<*T*>		r2	s<*T**>		r2	r2			
<*F*>									
<(>									
<*i*>									
<*E*+>									
<*T**>									
<(*E*>									
<*E*+*T*>									
<*T***F*>									
<(*E*)>									

$Contents(<T>)$:

$I = E \rightarrow T\bullet \in Contents(<T>)$: $Follow(E) = \{+,), \$\}$
$\alpha[<T>, +] = \alpha[<T>,)] = \alpha[<T>, \$] := r2$
Note: $E \rightarrow T$ is rule with label 2

$I = T \rightarrow T\bullet * F \in Contents(<T>)$:
$T \rightarrow T*\bullet F \in Contents(<T*>)$: $\alpha[<E>, +] = s<T*>$

Construct the rest analogically.

Construction of LR-table: Example 5/5

Final LR-table for G_{expr1}

	α						β		
	i	+	*	(	)	$	E	T	F
<ε>	s<i>			s<(>			<E>	<T>	<F>
<E>		s<E+>				☺			
<T>		r2	s<T*>		r2	r2			
<F>		r4	r4		r4	r4			
<(>	s<i>			s<(>			<(E>	<T>	<F>
<i>		r6	r6		r6	r6			
<E+>	s<i>			s<(>				<E+T>	<F>
<T*>	s<i>			s<(>					<T*F>
<(E>		s<E+>			s<(E)>				
<E+T>		r1	s<T*>		r1	r1			
<T*F>		r3	r3		r3	r3			
<(E)>		r5	r5		r5	r5			

Renaming the states

Rename the states:

Old	New
<ε>	*0*
<*E*>	*1*
<*T*>	*2*
<*F*>	*3*
<(>	*4*
<*i*>	*5*
<*E*+>	*6*
<*T**>	*7*
<(*E*>	*8*
<*E*+*T*>	*9*
<*T***F*>	*10*
<(*E*)>	*11*

LR table for G_{expr1} with the renamed states:

α	*i*	+	*	(	)	$
0	s5			s4		
1		s6				☺
2		r2	s7		r2	r2
3		r4	r4		r4	r4
4	s5			s4		
5		r6	r6		r6	r6
6	s5			s4		
7	s5			s4		
8		s6			s11	
9		r1	s7		r1	r1
10		r3	r3		r3	r3
11		r5	r5		r5	r5

β	*E*	*T*	*F*
0	1	2	3
1			
2			
3			
4	8	2	3
5			
6		9	3
7			10
8			
9			
10			
11			