Part VIII.
(Bonus)
LR Parsing

LR-Parser

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

- LR-parser is a EPDA, M, with states $Q = \{q_0, q_1, \ldots, q_k\}$, 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:

α	a_1	…	a_j	…	a_m	$
q_0						
…						
q_i						
…						
q_k						

$\alpha[q_i , a_j]$ = **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:

β	A_1	…	A_j	…	A_n
q_0					
…					
q_i					
…					
q_k					

$\beta[q_i , A_j]$ = **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 α[*state*, a] **of:**
    - **s***q*: push(⟨a, q⟩) & read next a from input string & *state* := q;
    - **r***p*: replace the pushdown top ⟨?, q⟩⟨X_1,?⟩⟨X_2,?⟩…⟨X_n,?⟩ with ⟨A, *state*⟩ where p: $A \to X_1X_2\ldots X_n \in P$ **and** *state* := β[q, A] & write p to output;
    - ☺: **success**
    - **blank**: **error**

  until success or error

LR-Parser: Example 1/2

$K = (N, T, P, S)$, where $N = \{S, A\}$, $T = \{i, o, (,)\}$,

$P = \{1: S \to SoA,\ 2: S \to A,\ 3: A \to i,\ 4: A \to (S)\}$

LR-table for *K*:

α	*i*	*o*	(	)	$
0	s3		s4		
1		s6			☺
2		r2		r2	r2
3		r3		r3	r3
4	s3		s4		
5		s6		s8	
6	s3		s4		
7		r1		r1	r1
8		r4		r4	r4

Action part for *K*

Go-to part for *K*

β	*S*	*A*
0	1	2
1		
2		
3		
4	5	2
5		
6		7
7		
8		

LR-Parser: Example 2/2

Rules: 1: $S \to SoA$, 2: $S \to A$, 3: $A \to i$, 4: $A \to (S)$

Input string: $i\ o\ i\ \$$

Pushdown	St.	Input	Enter	Rule
$\langle \$, 0\rangle$	0	$ioi\$$	$\alpha[0, i] = s3$	
$\langle \$, 0\rangle\langle i, 3\rangle$	3	$oi\$$	$\alpha[3, o] = r3$	3: $A \to i$
			$\beta[0, A] = 2$	
$\langle \$, 0\rangle\langle A, 2\rangle$	2	$oi\$$	$\alpha[2, o] = r2$	2: $S \to A$
			$\beta[0, S] = 1$	
$\langle \$, 0\rangle\langle S, 1\rangle$	1	$oi\$$	$\alpha[1, o] = s6$	
$\langle \$, 0\rangle\langle S, 1\rangle\langle o, 6\rangle$	6	$i\$$	$\alpha[6, i] = s3$	
$\langle \$, 0\rangle\langle S, 1\rangle\langle o, 6\rangle\langle i, 3\rangle$	3	$\$$	$\alpha[3, \$] = r3$	3: $A \to i$
			$\beta[6, A] = 7$	
$\langle \$, 0\rangle\langle S, 1\rangle\langle o, 6\rangle\langle A, 7\rangle$	7	$\$$	$\alpha[7, \$] = r1$	1: $S \to SoA$
			$\beta[0, S] = 1$	
$\langle \$, 0\rangle\langle S, 1\rangle$	1	$\$$	$\alpha[1, \$] =$ ☺	

Success

Right parse: 3231

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:

$K = (N, T, P, S)$, where $N = \{S, A\}$, $T = \{i, o, (,)\}$,
$P = \{1\text{: } S \rightarrow SoA,\ 2\text{: } S \rightarrow A,\ 3\text{: } A \rightarrow i,\ 4\text{: } A \rightarrow (S)\}$

Extended grammar for *K*:
$H = (N, T, P, S')$, where $N = \{S', S, A\}$, $T = \{i, o, (,)\}$,
$P = \{0\text{: } S' \rightarrow S,\ 1\text{: } S \rightarrow SoA,\ 2\text{: } S \rightarrow A,\ 3\text{: } A \rightarrow i,\ 4\text{: } A \rightarrow (S)\}$

Construction of LR Table: Items

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

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

Example: Consider $\boldsymbol{S} \rightarrow \boldsymbol{SoA}$

All items for $\boldsymbol{S} \rightarrow \boldsymbol{SoA}$ are:

$\boldsymbol{S} \rightarrow \bullet\boldsymbol{SoA}$, $\boldsymbol{S} \rightarrow \boldsymbol{S}\bullet\boldsymbol{oA}$, $\boldsymbol{S} \rightarrow \boldsymbol{So}\bullet\boldsymbol{A}$, $\boldsymbol{S} \rightarrow \boldsymbol{SoA}\bullet$

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

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** $A \to y \bullet Bz \in$ *Closure*(I) **and** $B \to x \in P$ **then** add $B \to \bullet x$ to *Closure*(I)

Closure of Item: Example 1/2

$H = (N, T, P, S')$, where $N = \{S', S, A\}$, $T = \{i, o, (,)\}$,
$P = \{0\text{: } S' \rightarrow S,\ 1\text{: } S \rightarrow SoA,\ 2\text{: } S \rightarrow A,\ 3\text{: } A \rightarrow i,\ 4\text{: } A \rightarrow (S)\}$

Task: $Closure(I)$ for $I = S' \rightarrow \bullet S$

$Closure(I) := \{S' \rightarrow \bullet S\}$

---

1) $S' \rightarrow \bullet S \in Closure(I)$ **&** $S \rightarrow SoA \in P$**:**
add $S \rightarrow \bullet SoA$ **to** $Closure(I)$

$Closure(I)$ **=** $\{S' \rightarrow \bullet S, S \rightarrow \bullet SoA\}$

---

2) $S' \rightarrow \bullet S \in Closure(I)$ **&** $S \rightarrow A \in P$**:**
add $S \rightarrow \bullet A$ **to** $Closure(I)$

$Closure(I)$ **=** $\{S' \rightarrow \bullet S, S \rightarrow \bullet SoA, S \rightarrow \bullet A\}$

Closure of Item: Example 2/2

$H = (N, T, P, S')$, where $N = \{S', S, A\}$, $T = \{i, o, (,)\}$,
$P = \{0\text{: } S' \to S,\ 1\text{: } S \to SoA,\ 2\text{: } S \to A,\ 3\text{: } A \to i,\ 4\text{: } A \to (S)\}$

---

3) $S \to \bullet A \in Closure(I)$ **&** $A \to i \in P$**:**
add $A \to \bullet i$ **to** $Closure(I)$

$Closure(I) = \{S' \to \bullet S,\ S \to \bullet SoA,\ S \to \bullet A,\ A \to \bullet i\}$

---

4) $S \to \bullet A \in Closure(I)$ **&** $A \to (S) \in P$**:**
add $A \to \bullet(S)$ **to** $Closure(I)$

Summary:

$Closure(I) = \{S' \to \bullet S,\ S \to \bullet SoA,\ S \to \bullet A,\ A \to \bullet i,\ A \to \bullet(S)\}$

Set $\Theta_U(\boldsymbol{I})$ for grammar G

Gist: For a symbol U and set of items I, $\Theta_U(I)$ denotes union of all closures of the form $Closure(A \to yU\bullet z)$, where $A \to y\bullet Uz$ is in I.

Definition: Let $G = (N, T, P, S)$ be a CFG, $\boldsymbol{I}$ be a set of items, and $\boldsymbol{U} \in T \cup N$. Then,

$\Theta_U(\boldsymbol{I}) = \{j: j \in Closure(A \to yU\bullet z), A \to y\bullet Uz \in \boldsymbol{I}\}$

Example:

$H = (N, T, P, S')$, where $N = \{S', S, A\}$, $T = \{i, o, (,)\}$,

$P = \{0: S' \to S,\ 1: S \to SoA,\ 2: S \to A,\ 3: A \to i,\ 4: A \to (S)\}$,

$\boldsymbol{I} = \{S \to So\bullet A, S \to \bullet A, A \to \bullet(S)\}$

Task: $\Theta_A(\boldsymbol{I})$

$Closure(S \to SoA\bullet) \cup Closure(S \to A\bullet) = \{S \to SoA\bullet, S \to A\bullet\}$

Task: $\Theta_((\boldsymbol{I})$

$Closure(A \to (\bullet S)) = \{A \to (\bullet S), S \to \bullet SoA, S \to \bullet A, A \to \bullet i, A \to \bullet(S)\}$

Set Θ_G for Grammar G

Note: Set Θ_G for grammar G is the set of sets of items defined by the following algorithm:

- **Input:** Extended $G = (N, T, P, S')$
- **Output:** Θ_G for grammar G

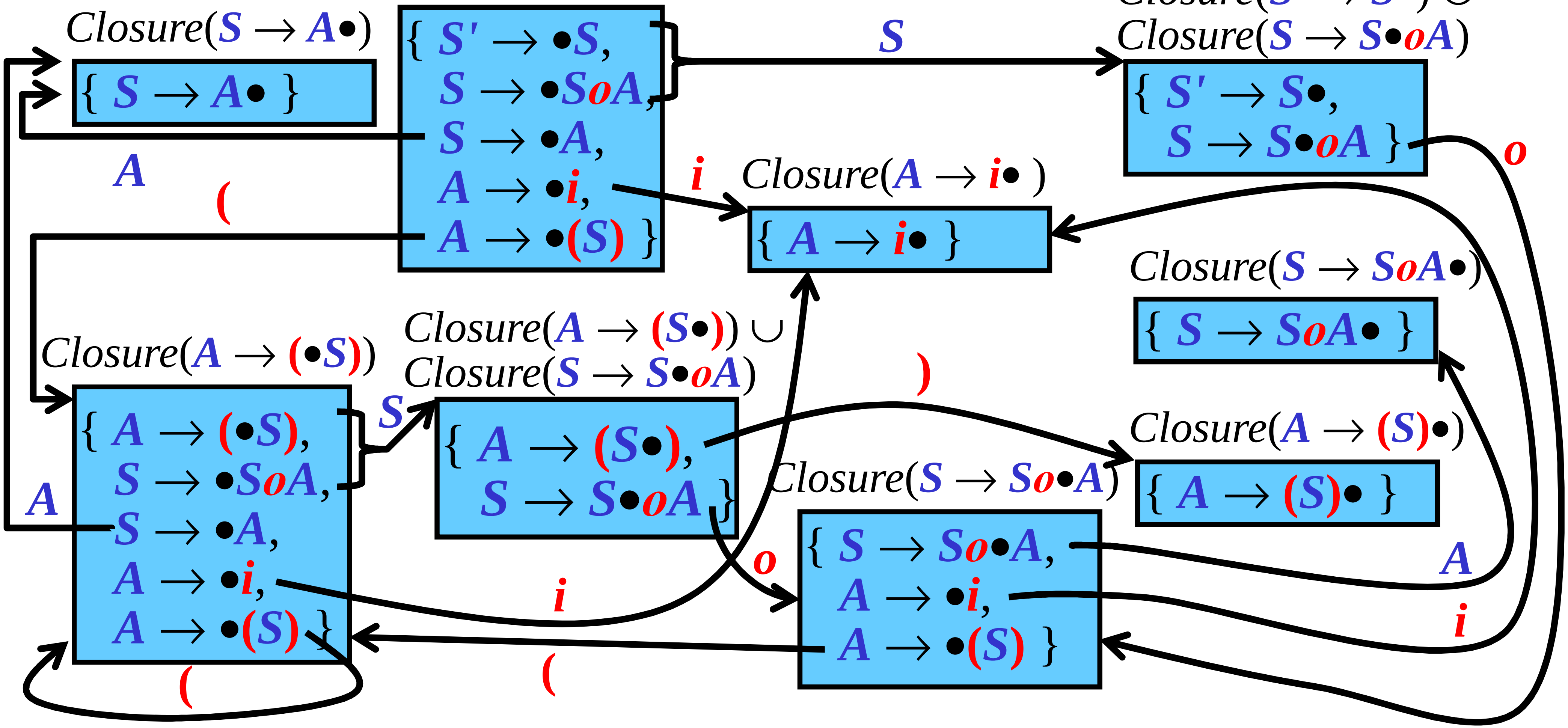
- **Method:**
- $\Theta_G := \{Closure(\boldsymbol{S'} \rightarrow \bullet \boldsymbol{S})\}$;
- **for each** $\boldsymbol{I} \in \Theta_G$ **and** $\boldsymbol{U} \in N \cup T$

  if $\Theta_U(\boldsymbol{I}) \neq \emptyset$ **then** include set $\Theta_U(\boldsymbol{I})$ into Θ_G

Set Θ_G: Example

$H = (N, T, P, S')$, where $N = \{S', S, A\}$, $T = \{i, o, (,)\}$,
$P = \{0: S' \to S,\ 1: S \to SoA,\ 2: S \to A,\ 3: A \to i,\ 4: A \to (S)\}$

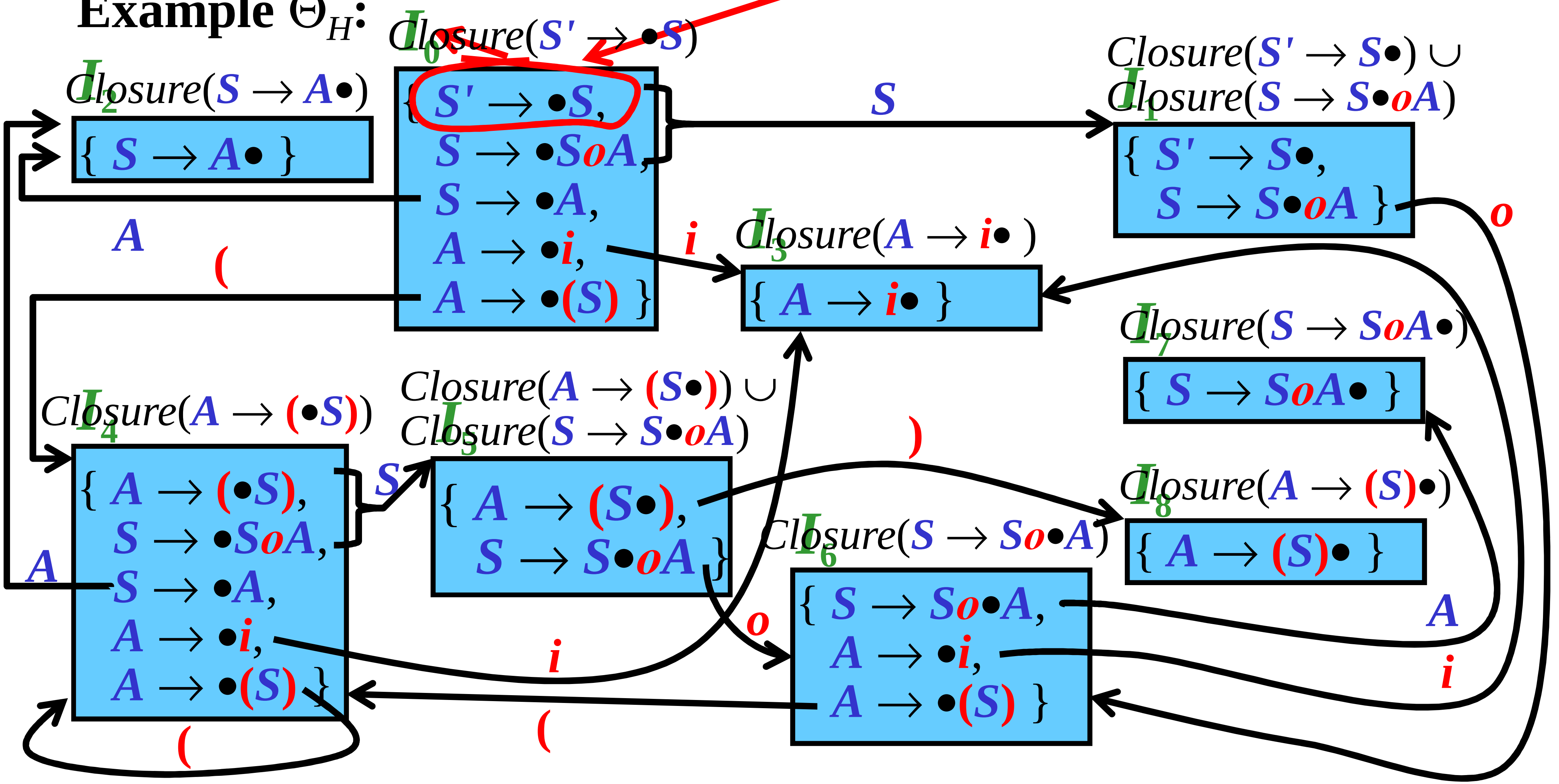
Initialization:



Naming of members in set Θ_G

Name the elements of Θ_G as I_0 to I_n, where n+1 is number of elements in Θ_G. The member with $S' \rightarrow \bullet S$ is I_0.

Example Θ_H:



Construction of LR-table: SLR Algorithm

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

- **Method:**
- $StatesOfTable := \Theta_G$; $StartState := Closure(\boldsymbol{S'} \rightarrow \bullet \boldsymbol{S})$;
- **<u>for each</u>** $x \in \Theta_G$ **<u>do</u>**
- **<u>for each</u>** $\boldsymbol{I} \in x$ **<u>do</u>**
    - **<u>case</u>** $\boldsymbol{I}$ **<u>of</u>**
        - $\boldsymbol{I} = \boldsymbol{A} \rightarrow \boldsymbol{y}\bullet\boldsymbol{X}\mathbf{z}$, where $\boldsymbol{X} \in N$:
          $\beta[x, \boldsymbol{X}] := \Theta_{X}(x)$
        - $\boldsymbol{I} = \boldsymbol{A} \rightarrow \boldsymbol{y}\bullet\boldsymbol{X}\mathbf{z}$, where $\boldsymbol{X} \in T$:
          $\alpha[x, \boldsymbol{X}] := \mathrm{s}\Theta_{X}(x)$
        - $\boldsymbol{I} = \mathbf{S'} \rightarrow \boldsymbol{S}\bullet$: $\alpha[x, \mathbf{\$}] :=$ ☺
        - $\boldsymbol{I} = \boldsymbol{A} \rightarrow \boldsymbol{y}\bullet$ $(\boldsymbol{A} \neq \mathbf{S'})$:
          <u>for each</u> $\boldsymbol{a} \in Follow(\boldsymbol{A})$ **<u>do</u>** $\alpha[x, \boldsymbol{a}] := \mathbf{r}\boldsymbol{p}$,
          where $\boldsymbol{p}$ is a label of rule $\boldsymbol{A} \rightarrow \boldsymbol{y}$

Construction of LR-table: Example 1/5

$\Theta_H = \{I_0{:}\{S' \to \bullet S,\ S \to \bullet SoA,\ S \to \bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_1{:}\{S' \to S\bullet,\ S \to S\bullet oA\},\ I_2{:}\{S \to A\bullet\},\ I_3{:}\{A \to i\bullet\},$
$I_4{:}\{A \to (\bullet S),\ S \to \bullet SoA,\ S \to \bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_5{:}\{A \to (S\bullet),\ S \to S\bullet oA\},\ I_6{:}\{S \to So\bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_7{:}\{S \to SoA\bullet\},\ I_8{:}\{A \to (S)\bullet\}\}$

Task: LR-table for *K*

	α					β	
	i	*o*	(	)	$	*S*	*A*
I_0	sI_3		sI_4			I_1	I_2

$S' \to \bullet S \in I_0,\ S \in N$: $\beta[I_0, S] := \Theta_S(I_0) = I_1$

$S \to \bullet SoA \in I_0,\ S \in N$: $\beta[I_0, S] := \Theta_S(I_0) = I_1$

$S \to \bullet A \in I_0,\ A \in N$: $\beta[I_0, A] := \Theta_A(I_0) = I_2$

$A \to \bullet i \in I_0,\ i \in T$: $\alpha[I_0, i] := \mathrm{s}\Theta_i(I_0) = \mathrm{s}I_3$

$A \to \bullet(S) \in I_0,\ (\in T$: $\alpha[I_0, (] := \mathrm{s}\Theta_((I_0) = \mathrm{s}I_4$

Construction of LR-table: Example 2/5

$\Theta_H = \{I_0\!:\!\{S' \to \bullet S,\ S \to \bullet SoA,\ S \to \bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_1\!:\!\{S' \to S\bullet,\ S \to S\bullet oA\},\ I_2\!:\!\{S \to A\bullet\},\ I_3\!:\!\{A \to i\bullet\},$
$I_4\!:\!\{A \to (\bullet S),\ S \to \bullet SoA,\ S \to \bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_5\!:\!\{A \to (S\bullet),\ S \to S\bullet oA\},\ I_6\!:\!\{S \to So\bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_7\!:\!\{S \to SoA\bullet\},\ I_8\!:\!\{A \to (S)\bullet\}\}$

Task: LR-table for *K*

	α					β	
	i	*o*	(	)	$	*S*	*A*
I_0	sI_3		sI_4			I_1	I_2
I_1		sI_6			☺		

$S' \to S\bullet \in I_1$: $\alpha[I_1, \$] :=$ ☺

$S \to S\bullet oA \in I_1$, $o \in T$: $\alpha[I_1, o] := s\Theta_o(I_1) = sI_6$

Construction of LR-table: Example 3/5

$\Theta_H = \{I_0{:}\{S' \to \bullet S,\ S \to \bullet SoA,\ S \to \bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_1{:}\{S' \to S\bullet,\ S \to S\bullet oA\},\ I_2{:}\{S \to A\bullet\},\ I_3{:}\{A \to i\bullet\},$
$I_4{:}\{A \to (\bullet S),\ S \to \bullet SoA,\ S \to \bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_5{:}\{A \to (S\bullet),\ S \to S\bullet oA\},\ I_6{:}\{S \to So\bullet A,\ A \to \bullet i,\ A \to \bullet(S)\},$
$I_7{:}\{S \to SoA\bullet\},\ I_8{:}\{A \to (S)\bullet\}\}$

Task: LR-table for *K*

	α					β	
	i	o	(	)	$	S	A
I_0	sI_3		sI_4			I_1	I_2
I_1		sI_6			☺		
I_2		r2		r2	r2		

$S \to A\bullet \in I_2$, $Follow(S) = \{o,), \$\}$:

$\alpha[I_2, o] = \alpha[I_2,)] = \alpha[I_2, \$] := r2$

Construction of LR-table: Example 4/5

$\Theta_H = \{I_0:\{S' \to \bullet S, S \to \bullet SoA, S \to \bullet A, A \to \bullet i, A \to \bullet(S)\},$
$I_1:\{S' \to S\bullet, S \to S\bullet oA\}, I_2:\{S \to A\bullet\}, I_3:\{A \to i\bullet\},$
$I_4:\{A \to (\bullet S), S \to \bullet SoA, S \to \bullet A, A \to \bullet i, A \to \bullet(S)\},$
$I_5:\{A \to (S\bullet), S \to S\bullet oA\}, I_6:\{S \to So\bullet A, A \to \bullet i, A \to \bullet(S)\},$
$I_7:\{S \to SoA\bullet\}, I_8:\{A \to (S)\bullet\}\}$

Task: LR-table for K

	α					β	
	i	o	(	)	\$	S	A
I_0	sI_3		sI_4			I_1	I_2
I_1		sI_6			☺		
I_2		r2		r2	r2		
I_3		r3		r3	r3		

Construct the rest analogically.

$A \to i\bullet \in I_3$, $Follow(A) = \{o,), \$\}$:
$\alpha[I_3, o] = \alpha[I_3,)] = \alpha[I_3, \$] :=$ r3

Construction of LR-table: Example 5/5

Final LR-table for *K*

	α					β	
	i	***o***	**(**	**)**	**$**	***S***	***A***
I_0	sI_3		sI_4			I_1	I_2
I_1		sI_6			☺		
I_2		r2		r2	r2		
I_3		r3		r3	r3		
I_4	sI_3		sI_4			I_5	I_2
I_5		sI_6		sI_8			
I_6	sI_3		sI_4				I_7
I_7		r1		r1	r1		
I_8		r4		r4	r4		

Renaming the states

Rename the states:

Old	New
I_0	0
I_1	1
I_2	2
I_3	3
I_4	4
I_5	5
I_6	6
I_7	7
I_8	8

LR-table for *K* with the renamed states:

α	*i*	*o*	(	)	$
0	s3		s4		
1		s6			☺
2		r2		r2	r2
3		r3		r3	r3
4	s3		s4		
5		s6		s8	
6	s3		s4		
7		r1		r1	r1
8		r4		r4	r4

β	*S*	*A*
0	1	2
1		
2		
3		
4	5	2
5		
6		7
7		
8		