Determining the Type of Grammar

Compilers course

Masters in Informatics and Computing Engineering (MIEIC), 3rd Year

João M. P. Cardoso
Email: jmpc@fe.up.pt

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

DEI DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Example

Given the following grammar, identify if the syntactic analyzer can be LL(1), LR(0), or LR(1)

- $S \rightarrow X\ \$$ (1)
- $X \rightarrow$ "(" | Y (2, 3)
- $Y \rightarrow$ "(" Y ")" | ε (4, 5)

Identify the Type of the Grammar

Let's assume X the type of the analyzer (LL(1), LR(0), etc.):

1. Build the parser table for X
   - For the LR grammars requires the construction of the DFA
2. In case of conflicts in the parser table, the grammar cannot be implemented by analyzer X

   In case of no conflict in the parser table, the grammar can be implemented by analyzer X

Determining the Type of the Grammar: LL(1)

LL(1)

- Parser table directly obtained:
  - For each production of the grammar determine the First set of the RHS
    - If RHS can derive ε then determine the Follow set for the LHS
  - Place each production in the columns corresponding to the terminal symbols belong to the sets determined in the step before and in the row that identifies the LHS

$S \rightarrow X$ (1)

$X \rightarrow$ "(" | Y (2, 3)

$Y \rightarrow$ "(" Y ")" | ε (4, 5)

LL(1)

- Parser Table
  - Production: $S \rightarrow X$
  - First(X) = {"("} $\Rightarrow$ place $S \rightarrow X$ in cell [S, "("]

Non-terminals	Terminals	
	"("	")"
S	$S \rightarrow X$	
X		
Y		

$S \rightarrow X$ (1)

$X \rightarrow$ "(" | Y (2, 3)

$Y \rightarrow$ "(" Y ")" | ε (4, 5)

LL(1)

- Parser Table
  - Production: S $\rightarrow$ X
  - First(X) = {"("} $\Rightarrow$ place S $\rightarrow$ X in cell [S, "("]
  - Since S can derive ε (via X) determine Follow(S) = { }

Non-terminals	Terminals	
	"("	")"
S	S $\rightarrow$ X	
X		
Y		

S $\rightarrow$ X (1)
X $\rightarrow$ "(" | Y (2, 3)
Y $\rightarrow$ "(" Y ")" | ε (4, 5)

LL(1)

- Parser table
  - Production: X → “(“ (2)
  - First(“(“) = {“(“} ⇒ place X → “(“ in cell [X, “(“]

Non-terminals	Terminals	
	“(”	“)”
S	S → X	
X	X → “(“	
Y		

S → X (1)
X → “(“ | Y (2, 3)
Y → “(“ Y “)” | ε (4, 5)

LL(1)

- Parser table
  - Production: $X \rightarrow Y$ (3)
  - First(Y) = {"("} $\Rightarrow$ place $X \rightarrow Y$ in cell [X, "("]

Non-terminals	Terminals	
	"("	")"
S	$S \rightarrow X$	
X	$X \rightarrow$ "(" $X \rightarrow Y$	
Y		

$S \rightarrow X$ (1)
$X \rightarrow$ "(" | Y (2, 3)
$Y \rightarrow$ "(" Y ")" | ε (4, 5)

LL(1)

- Parser table
  - Production: X → Y (3)
  - First(Y) = {"("} ⇒ place Y → "(" in cell [Y, "("]
  - Since X can derive ε (via Y) determine Follow(X) = { }

Non-terminals	Terminals	
	"("	")"
S	S → X	
X	X → "(" X → Y	
Y		

S → X (1)
X → "(" \| Y (2, 3)
Y → "(" Y ")" \| ε (4, 5)

LL(1)

- Parser table
  - Production: Y → "(" Y ")"   (4)
  - First("(" Y ")") = {"("} ⇒ place Y → "(" Y ")" in cell [Y, "("]

Non-terminals	Terminals	
	"("	")"
S	S → X	
X	X → "(" X → Y	
Y	Y → "(" Y ")"	

S → X   (1)
X → "(" \| Y   (2, 3)
Y → "(" Y ")" \| ε   (4, 5)

LL(1)

- Parser table
  - Production: $Y \rightarrow \varepsilon$ (5)
  - Follow(Y) = {“)“} $\Rightarrow$ place $Y \rightarrow \varepsilon$ in cell [Y, “)“]

Non-terminals	Terminals	
	“(”	“)”
S	$S \rightarrow X$	
X	$X \rightarrow$ “(“ $X \rightarrow Y$	
Y	$Y \rightarrow$ “(“ Y “)”	$Y \rightarrow \varepsilon$

$S \rightarrow X$ (1)

$X \rightarrow$ “(“ | Y (2, 3)

$Y \rightarrow$ “(“ Y “)” | ε (4, 5)

LL(1)

- Parser table
  - We have already analyzed all the productions of the grammar
  - Table is complete

Non-terminals	Terminals	
	"("	")"
S	$S \rightarrow X$	
X	$X \rightarrow$ "(" $X \rightarrow Y$	
Y	$Y \rightarrow$ "(" Y ")"	$Y \rightarrow \varepsilon$

$S \rightarrow X$ (1)

$X \rightarrow$ "(" | Y (2, 3)

$Y \rightarrow$ "(" Y ")" | ε (4, 5)

LL(1)

- Parser table directly obtained

Conflict between two productions to apply
$\Rightarrow$ not LL(1)

Non-terminals	Terminals	
	“(”	“)”
S	S → X	
X	X → “(“ X → Y	
Y	Y → “(“ Y “)”	Y → ε

S → X (1)
X → “(“ \| Y (2, 3)
Y → “(“ Y “)” \| ε (4, 5)

LL

- How to determine if it is LL(2), LL(3), LL(k)?

Determining the Type of the Grammar: LR(0)

LR(0)

1. Identify items LR(0)

$S \rightarrow X$ (1)
$X \rightarrow$ "(" | Y (2, 3)
$Y \rightarrow$ "(" Y ")" | ε (4, 5)

Items:

- $S \rightarrow \bullet X \$$
- $S \rightarrow X \bullet \$$
- $X \rightarrow \bullet$ "("
- $X \rightarrow$ "(" $\bullet$
- $X \rightarrow \bullet Y$
- $X \rightarrow Y \bullet$
- $Y \rightarrow \bullet$ "(" Y ")"
- $Y \rightarrow$ "(" $\bullet Y$ ")"
- $Y \rightarrow$ "(" $Y \bullet$ ")"
- $Y \rightarrow$ "(" Y ")" $\bullet$
- $Y \rightarrow \bullet$

LR(0)

1. Identify items LR(0)
2. Build DFA

$S \rightarrow X$ (1)
$X \rightarrow$ "(" | Y (2, 3)
$Y \rightarrow$ "(" Y ")" | ε (4, 5)

Items:

- $S \rightarrow \bullet\, X\ \$$
- $S \rightarrow X \bullet \$$
- $X \rightarrow \bullet$ "("
- $X \rightarrow$ "(" $\bullet$
- $X \rightarrow \bullet\, Y$
- $X \rightarrow Y \bullet$
- $Y \rightarrow \bullet$ "(" Y ")"
- $Y \rightarrow$ "(" $\bullet\, Y$ ")"
- $Y \rightarrow$ "(" $Y \bullet$ ")"
- $Y \rightarrow$ "(" Y ")" $\bullet$
- $Y \rightarrow \bullet$

LR(0)

2. Build DFA
   – Begin with 1st item and assign it a state

s0

$$\boxed{S \rightarrow \bullet X \$}$$

Items:

- $S \rightarrow \bullet X \$$
- $S \rightarrow X \bullet \$$
- X → • “(“
- X → “(“ •
- $X \rightarrow \bullet Y$
- $X \rightarrow Y \bullet$
- Y → • “(“ Y “)”
- Y → “(“ • Y “)”
- Y → “(“ Y • “)”
- Y → “(“ Y “)” •
- $Y \rightarrow \bullet$

S → X (1)
X → “(“ | Y (2, 3)
Y → “(“ Y “)” | ε (4, 5)

LR(0)

s0

$\boxed{S \rightarrow \bullet X \$}$

2. Build DFA

- Begin with first item and assign it a state (s0)
- Determine closure($S \rightarrow \bullet X \$$)
    - Add to state s0 all the items with X in the LHS and with RHS beginning with •

$S \rightarrow X$ (1)

$X \rightarrow$ "(" | Y (2, 3)

$Y \rightarrow$ "(" Y ")" | ε (4, 5)

Items:

- S → • X $
- S → X • $
- X → • "("
- X → "(" •
- X → • Y
- X → Y •
- Y → • "(" Y ")"
- Y → "(" • Y ")"
- Y → "(" Y • ")"
- Y → "(" Y ")" •
- Y → •

LR(0)

2. Build DFA

s0

$$\boxed{\begin{array}{l} S \rightarrow \bullet X \$ \\ X \rightarrow \bullet \text{“(“} \\ X \rightarrow \bullet Y \end{array}}$$

- Begin with first item and assign it a state (s0)
- Determine closure($S \rightarrow \bullet X \$$)
  - Add to state s0 all the items with X in the LHS and with RHS beginning with •

Items:

- $S \rightarrow \bullet X \$$
- $S \rightarrow X \bullet \$$
- $X \rightarrow \bullet$ “(“
- $X \rightarrow$ “(“ $\bullet$
- $X \rightarrow \bullet Y$
- $X \rightarrow Y \bullet$
- $Y \rightarrow \bullet$ “(“ Y “)”
- $Y \rightarrow$ “(“ $\bullet$ Y “)”
- $Y \rightarrow$ “(“ Y $\bullet$ “)”
- $Y \rightarrow$ “(“ Y “)” $\bullet$
- $Y \rightarrow \bullet$

$S \rightarrow X$ (1)

$X \rightarrow$ “(“ | Y (2, 3)

$Y \rightarrow$ “(“ Y “)” | ε (4, 5)

LR(0)

2. Build DFA

– Begin with first item and assign it a state (s0)

– Determine closure($S \to \bullet X\,\$$)

– For each item I added to state s0 determine closure(I)

s0

$S \to \bullet X\,\$$
$X \to \bullet$ "("
$X \to \bullet Y$

$S \to X$ (1)

$X \to$ "(" | Y (2, 3)

$Y \to$ "(" Y ")" | ε (4, 5)

Items:

- $S \to \bullet X\,\$$
- $S \to X \bullet \$$
- $X \to \bullet$ "("
- $X \to$ "(" $\bullet$
- $X \to \bullet Y$
- $X \to Y \bullet$
- $Y \to \bullet$ "(" Y ")"
- $Y \to$ "(" $\bullet$ Y ")"
- $Y \to$ "(" Y $\bullet$ ")"
- $Y \to$ "(" Y ")" $\bullet$
- $Y \to \bullet$

LR(0)

2. Build DFA

s0

$S \rightarrow \bullet X\,\$$
$X \rightarrow \bullet$ "("
$X \rightarrow \bullet Y$

- Begin with first item and assign it a state (s0)
- Determine closure($S \rightarrow \bullet X\,\$$)
- For each item I added to state s0 determine closure(I)
  - Closure($X \rightarrow \bullet$ "(") = { }

Items:

- S → • X $
- S → X • $
- X → • "("
- X → "(" •
- X → • Y
- X → Y •
- Y → • "(" Y ")"
- Y → "(" • Y ")"
- Y → "(" Y • ")"
- Y → "(" Y ")" •
- Y → •

S → X (1)

X → "(" | Y (2, 3)

Y → "(" Y ")" | ε (4, 5)

LR(0)

2. Build DFA

s0

$S \rightarrow \ \bullet X\,\$$
$X \ \rightarrow \bullet$ “(“
$X \ \rightarrow \bullet Y$

- Begin with first item and assign it a state (s0)
- Determine closure($S \rightarrow \ \bullet X\,\$$)
- For each item I added to state s0 determine closure(I)
  - Closure($X \ \rightarrow \bullet$ Y)

Items:

- S → • X $
- S → X • $
- X → • “(“
- X → “(“ •
- X → • Y
- X → Y •
- Y → • “(“ Y “)”
- Y → “(“ • Y “)”
- Y → “(“ Y • “)”
- Y → “(“ Y “)” •
- Y → •

S → X (1)

X → “(“ | Y (2, 3)

Y → “(“ Y “)” | ε (4, 5)

LR(0)

2. Build DFA

- Begin with first item and assign it a state (s0)
- Determine closure($S \rightarrow \bullet X \$$)
- For each item I added to state s0 determine closure(I)
  - Closure($X \rightarrow \bullet$ Y)

s0

$S \rightarrow \bullet X \$$
$X \rightarrow \bullet$ "("
$X \rightarrow \bullet$ Y
$Y \rightarrow \bullet$ "(" Y ")"
$Y \rightarrow \bullet$

Items:

- $S \rightarrow \bullet$ X \$
- $S \rightarrow$ X $\bullet$ \$
- $X \rightarrow \bullet$ "("
- $X \rightarrow$ "(" $\bullet$
- $X \rightarrow \bullet$ Y
- $X \rightarrow$ Y $\bullet$
- $Y \rightarrow \bullet$ "(" Y ")"
- $Y \rightarrow$ "(" $\bullet$ Y ")"
- $Y \rightarrow$ "(" Y $\bullet$ ")"
- $Y \rightarrow$ "(" Y ")" $\bullet$
- $Y \rightarrow \bullet$

S → X (1)
X → "(" | Y (2, 3)
Y → "(" Y ")" | ε (4, 5)

LR(0)

2. Build DFA

s0

$S \rightarrow \bullet X\$$
$X \rightarrow \bullet$ "("
$X \rightarrow \bullet Y$
$Y \rightarrow \bullet$ "(" Y ")"
$Y \rightarrow \bullet$

- Begin with first item and assign it a state (s0)
- Determine closure($S \rightarrow \bullet X\$$)
- For each item I added to state s0 determine closure(I)
- Stop when there aren't more items to add to s0

Items:

- $S \rightarrow \bullet X \$$
- $S \rightarrow X \bullet \$$
- $X \rightarrow \bullet$ "("
- $X \rightarrow$ "(" $\bullet$
- $X \rightarrow \bullet Y$
- $X \rightarrow Y \bullet$
- $Y \rightarrow \bullet$ "(" Y ")"
- $Y \rightarrow$ "(" $\bullet$ Y ")"
- $Y \rightarrow$ "(" Y $\bullet$ ")"
- $Y \rightarrow$ "(" Y ")" $\bullet$
- $Y \rightarrow \bullet$

$S \rightarrow X$		(1)
$X \rightarrow$ "(" | Y		(2, 3)
$Y \rightarrow$ "(" Y ")" | ε		(4, 5)

LR(0)

2. Build DFA

- After a state is complete
  - Determine possible transitions
  - Transition between states is related to the symbol that allows the movement of the point over that symbol

s0

$S \rightarrow \bullet X \$$
$X \rightarrow \bullet$ "("
$X \rightarrow \bullet Y$
$Y \rightarrow \bullet$ "(" Y ")"
$Y \rightarrow \bullet$

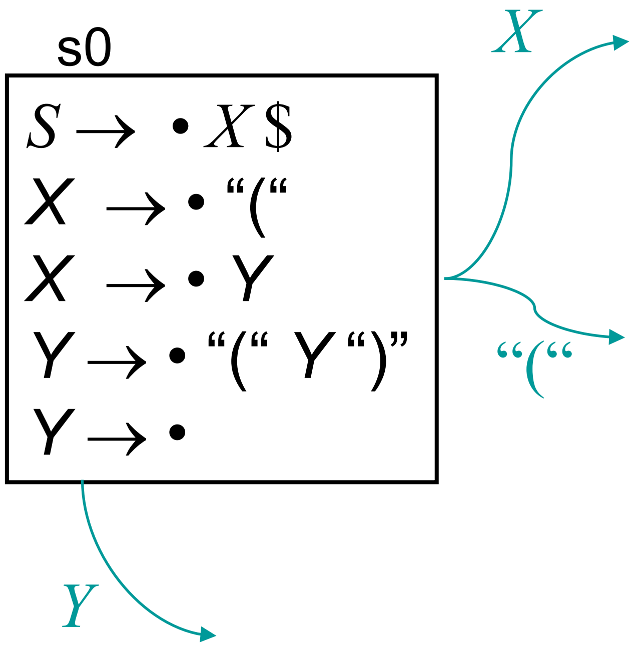
Items:

- $S \rightarrow \bullet X \$$
- $S \rightarrow X \bullet \$$
- $X \rightarrow \bullet$ "("
- $X \rightarrow$ "(" $\bullet$
- $X \rightarrow \bullet Y$
- $X \rightarrow Y \bullet$
- $Y \rightarrow \bullet$ "(" Y ")"
- $Y \rightarrow$ "(" $\bullet Y$ ")"
- $Y \rightarrow$ "(" $Y \bullet$ ")"
- $Y \rightarrow$ "(" Y ")" $\bullet$
- $Y \rightarrow \bullet$

$S \rightarrow X$ (1)
$X \rightarrow$ "(" | Y (2, 3)
$Y \rightarrow$ "(" Y ")" | ε (4, 5)

LR(0)

2. Build DFA



- After a state is complete
  - Determine possible transitions
  - Transition between states is related to the symbol that allows the movement of the point • over that symbol

$S \rightarrow X$ (1)

$X \rightarrow$ “(“ | Y (2, 3)

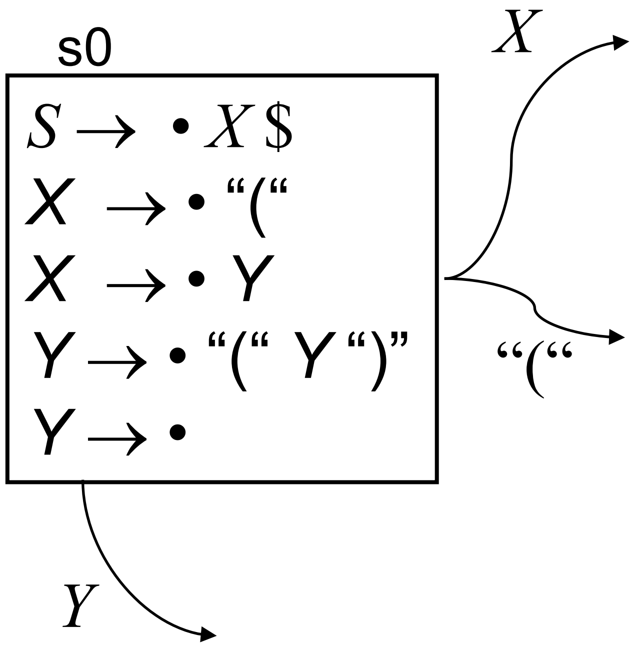
$Y \rightarrow$ “(“ Y “)” | ε (4, 5)

Items:

- $S \rightarrow \bullet X \$$
- $S \rightarrow X \bullet \$$
- $X \rightarrow \bullet$ “(“
- $X \rightarrow$ “(“ $\bullet$
- $X \rightarrow \bullet Y$
- $X \rightarrow Y \bullet$
- $Y \rightarrow \bullet$ “(“ Y “)”
- $Y \rightarrow$ “(“ $\bullet$ Y “)”
- $Y \rightarrow$ “(“ Y $\bullet$ “)”
- $Y \rightarrow$ “(“ Y “)” $\bullet$
- $Y \rightarrow \bullet$

LR(0)

2. Build DFA



$S \to X$ (1)

$X \to$ “(“ | Y (2, 3)

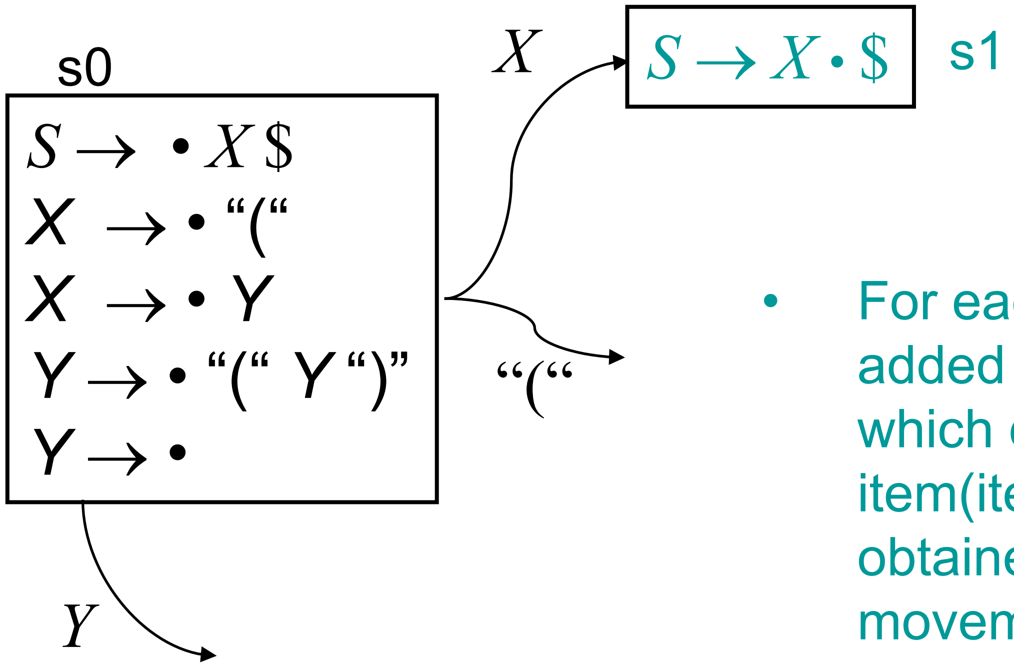
$Y \to$ “(“ Y “)” | ε (4, 5)

- After a state is complete
  - Determine possible transitions
  - For each transition it is added a new state which contains the item(items) directly obtained from the movement of the point

Items:

- $S \to \bullet X \$$
- $S \to X \bullet \$$
- $X \to \bullet$ “(“
- $X \to$ “(“ $\bullet$
- $X \to \bullet Y$
- $X \to Y \bullet$
- $Y \to \bullet$ “(“ Y “)”
- $Y \to$ “(“ $\bullet$ Y “)”
- $Y \to$ “(“ Y $\bullet$ “)”
- $Y \to$ “(“ Y “)” $\bullet$
- $Y \to \bullet$

LR(0)



- For each transition it is added a new state which contains the item(items) directly obtained from the movement of the point

$S \to X$ (1)

$X \to$ “(“ | Y (2, 3)

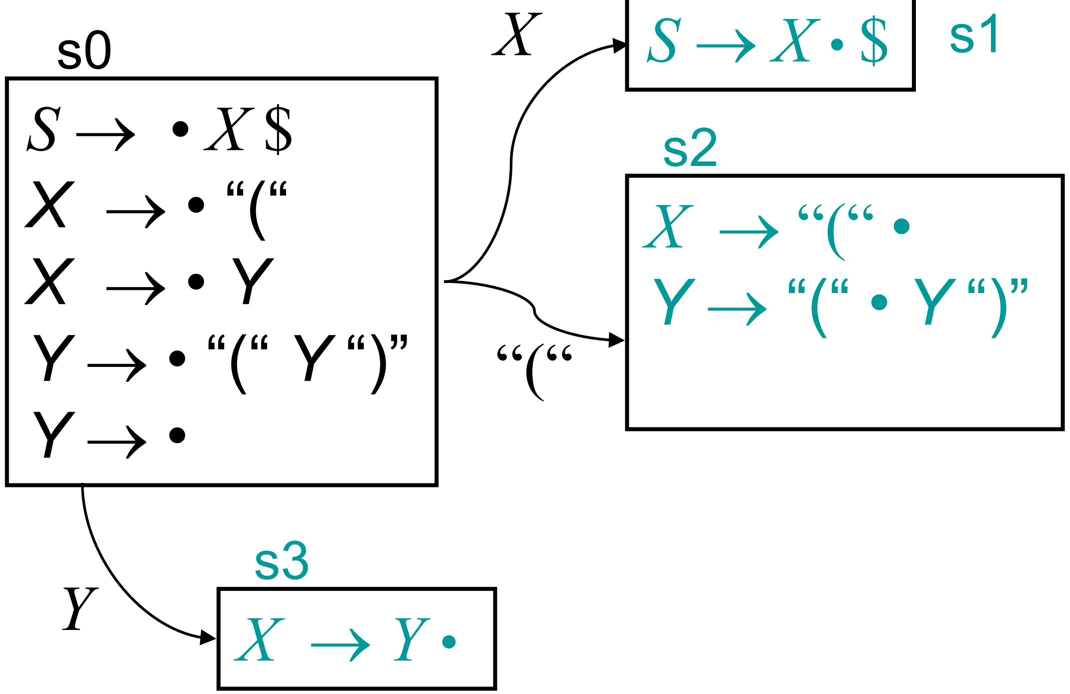
$Y \to$ “(“ Y “)” | ε (4, 5)

Items:

- $S \to \bullet X \$$
- $S \to X \bullet \$$
- $X \to \bullet$ “(“
- $X \to$ “(“ $\bullet$
- $X \to \bullet Y$
- $X \to Y \bullet$
- $Y \to \bullet$ “(“ Y “)”
- $Y \to$ “(“ $\bullet$ Y “)”
- $Y \to$ “(“ Y $\bullet$ “)”
- $Y \to$ “(“ Y “)” $\bullet$
- $Y \to \bullet$

LR(0)

- For each transition it is added a new state which contains the item(items) directly obtained from the movement of the point



$S \to X$ (1)

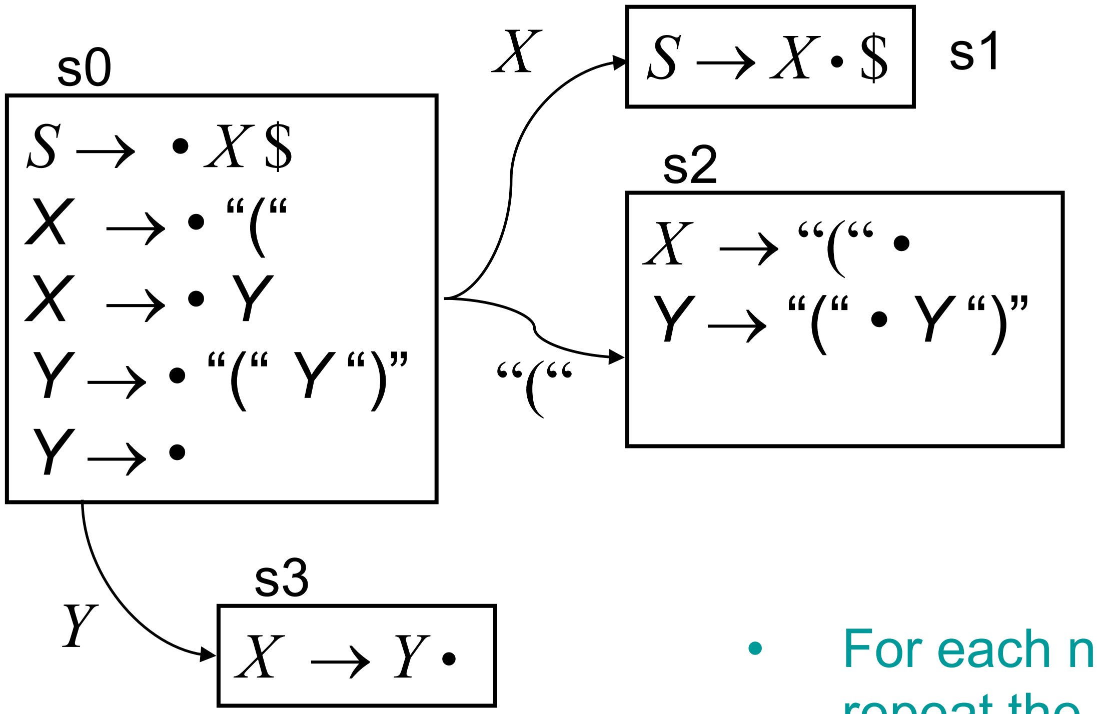
$X \to$ "(" | Y (2, 3)

$Y \to$ "(" Y ")" | ε (4, 5)

Items:

- $S \to \bullet X \$$
- $S \to X \bullet \$$
- $X \to \bullet$ "("
- $X \to$ "(" $\bullet$
- $X \to \bullet Y$
- $X \to Y \bullet$
- $Y \to \bullet$ "(" Y ")"
- $Y \to$ "(" $\bullet Y$ ")"
- $Y \to$ "(" $Y \bullet$ ")"
- $Y \to$ "(" Y ")" $\bullet$
- $Y \to \bullet$

LR(0)



S → X (1)

X → "(" | Y (2, 3)

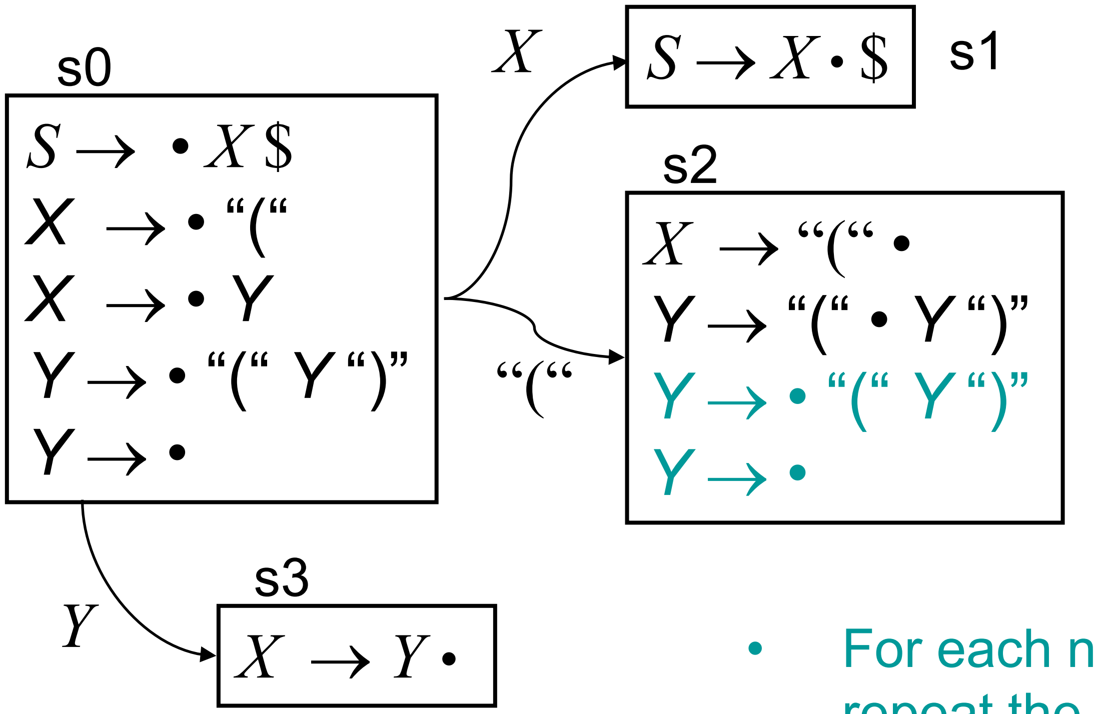
Y → "(" Y ")" | ε (4, 5)

- For each new state repeat the steps that were done to state s0

Items:

- S → • X $
- S → X • $
- X → • "("
- X → "(" •
- X → • Y
- X → Y •
- Y → • "(" Y ")"
- Y → "(" • Y ")"
- Y → "(" Y • ")"
- Y → "(" Y ")" •
- Y → •

LR(0)



$S \to X$ (1)

$X \to$ “(“ | Y (2, 3)

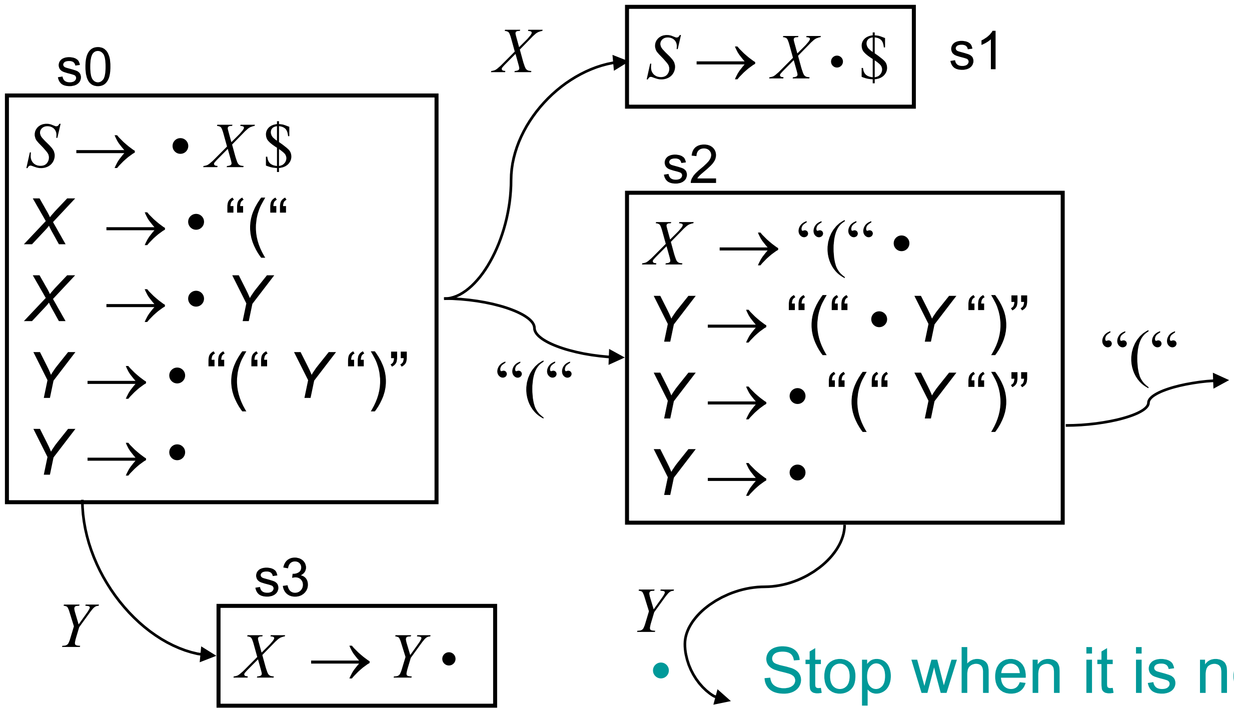
$Y \to$ “(“ Y “)” | ε (4, 5)

- For each new state repeat the steps that were done to state s0

Items:

- $S \to \bullet X \$$
- $S \to X \bullet \$$
- $X \to \bullet$ “(“
- $X \to$ “(“ $\bullet$
- $X \to \bullet Y$
- $X \to Y \bullet$
- $Y \to \bullet$ “(“ Y “)”
- $Y \to$ “(“ $\bullet$ Y “)”
- $Y \to$ “(“ Y $\bullet$ “)”
- $Y \to$ “(“ Y “)” $\bullet$
- $Y \to \bullet$

LR(0)



- Stop when it is not possible to add more states

$S \rightarrow X$ (1)

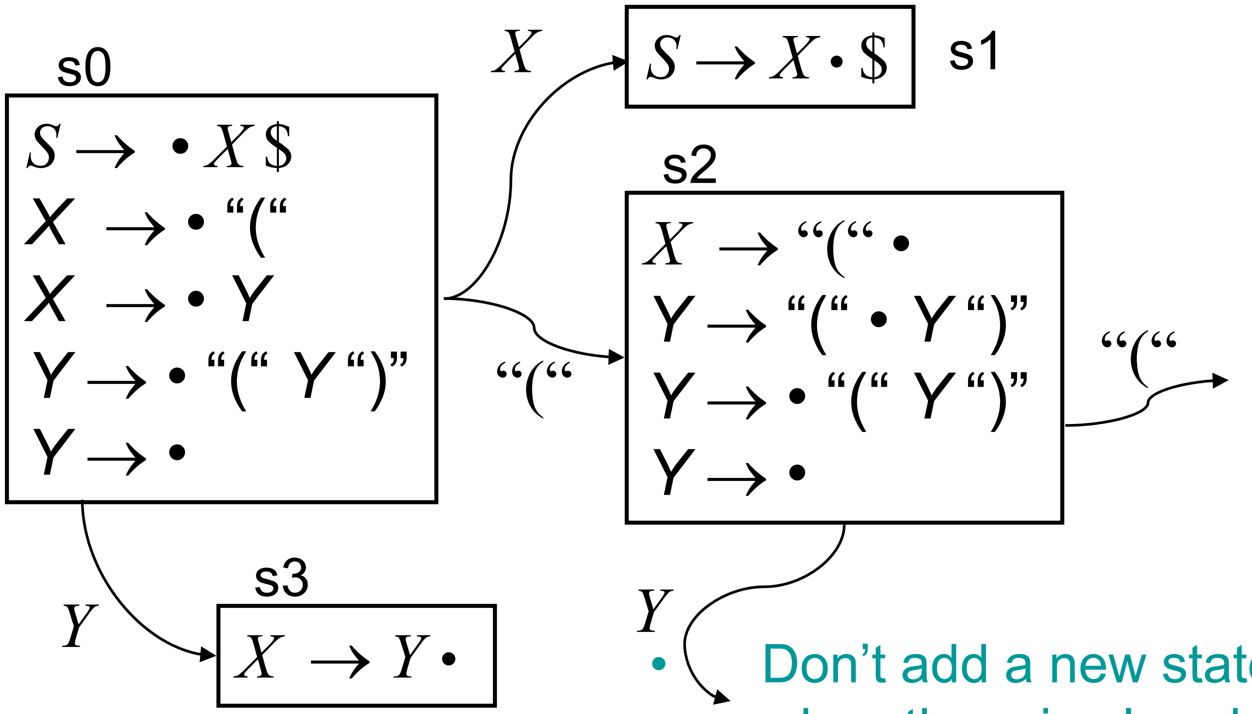
$X \rightarrow$ "(" | Y (2, 3)

$Y \rightarrow$ "(" Y ")" | ε (4, 5)

Items:

- $S \rightarrow \bullet X \$$
- $S \rightarrow X \bullet \$$
- $X \rightarrow \bullet$ "("
- $X \rightarrow$ "(" $\bullet$
- $X \rightarrow \bullet Y$
- $X \rightarrow Y \bullet$
- $Y \rightarrow \bullet$ "(" Y ")"
- $Y \rightarrow$ "(" $\bullet$ Y ")"
- $Y \rightarrow$ "(" Y $\bullet$ ")"
- $Y \rightarrow$ "(" Y ")" $\bullet$
- $Y \rightarrow \bullet$

LR(0)



- Don’t add a new state when there is already one state with the same items

$S \rightarrow X$ (1)

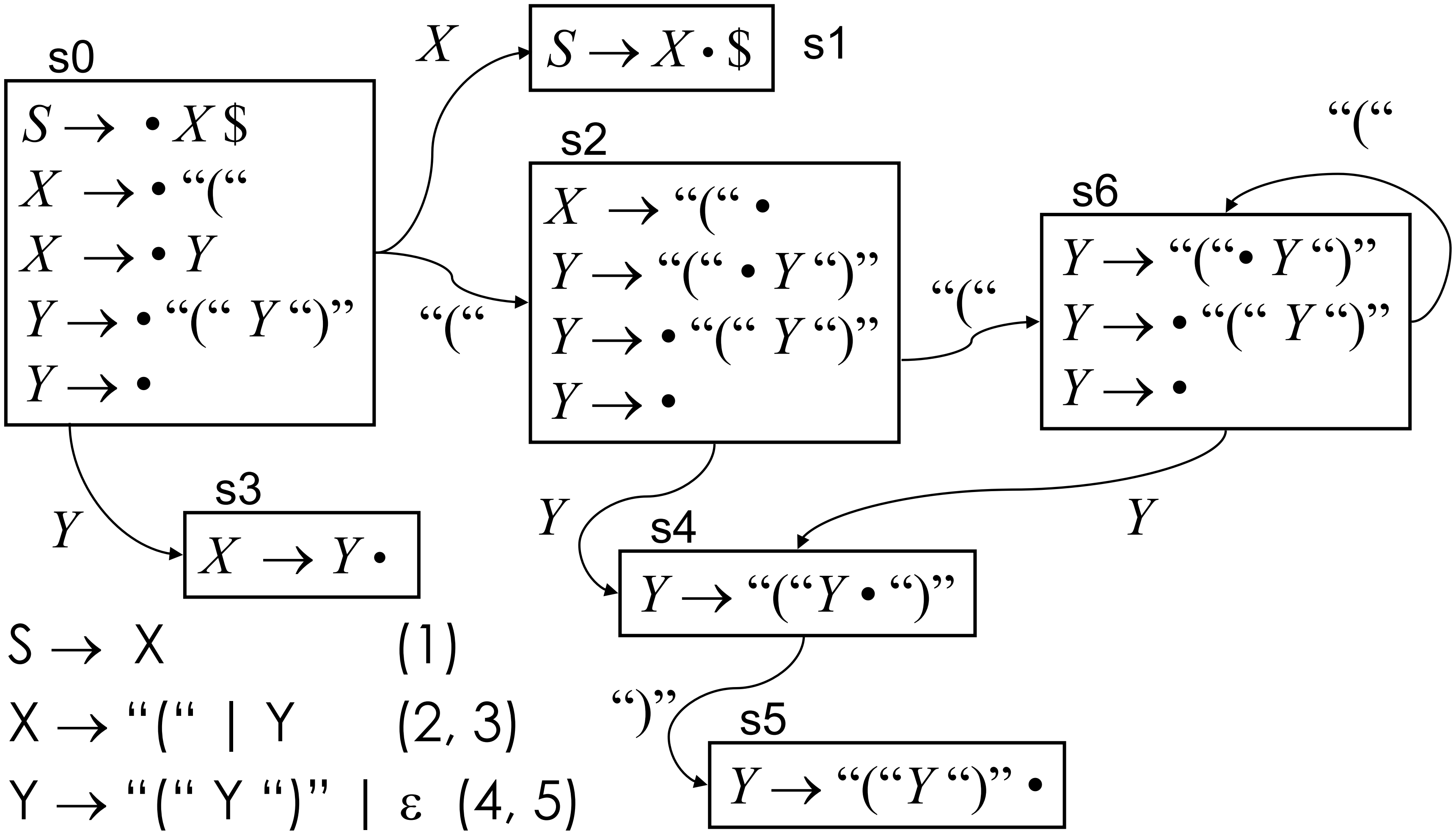
$X \rightarrow$ “(“ | Y (2, 3)

$Y \rightarrow$ “(“ Y “)” | ε (4, 5)

Items:

- $S \rightarrow \bullet X \$$
- $S \rightarrow X \bullet \$$
- $X \rightarrow \bullet$ “(“
- $X \rightarrow$ “(“ $\bullet$
- $X \rightarrow \bullet Y$
- $X \rightarrow Y \bullet$
- $Y \rightarrow \bullet$ “(“ Y “)”
- $Y \rightarrow$ “(“ $\bullet$ Y “)”
- $Y \rightarrow$ “(“ Y $\bullet$ “)”
- $Y \rightarrow$ “(“ Y “)” $\bullet$
- $Y \rightarrow \bullet$

LR(0)



S → X (1)

X → “(“ | Y (2, 3)

Y → “(“ Y “)” | ε (4, 5)

LR(0)

1. Identify LR(0) items
2. Build DFA
3. Build the parser table
4. Verify if there are conflicts

LR(0)

- Parser table

	“(“	“)”	$	X	Y
S0	Shift s2 Reduce (5)	Reduce (5)	Reduce (5)	Goto s1	goto s3
S1			Accept		
S2	Reduce (2) Reduce (5) Shift s6	Reduce (2) Reduce (5)	Reduce (2) Reduce (5)		Goto s4
S3	Reduce (3)	Reduce (3)	Reduce (3)		
S4		Shift s5			
S5	Reduce (4)	Reduce (4)	Reduce (4)		
S6	Reduce (5) Shift s6	Reduce (5)	Reduce (5)		Goto s4

reduce / reduce and shift/reduce conflicts

shift/reduce conflict

The gramar is not LR(0)!

Determining the Type of the Grammar: LR(1)

LR(1)

1. Identify LR(1) items

   Terminal symbols: “(“ e “)”

   Symbol of the end of the analysis: $

2. Build DFA

   - The same method as for LR(0) with differences in terms of the addition of items to a state

- S → X (1)
- X → “(“ | Y (2, 3)
- Y → “(“ Y “)” | ε (4, 5)

Items:

S → • X $	?
S → X • $	?
X → • “(“	“(“
X → • “(“	“)“
X → • “(“	$
X → “(“ •	“(“
X → “(“ •	“)“
X → “(“ •	$
X → • Y	“(“
X → • Y	“)“
X → • Y	$
X → Y •	“(“
X → Y •	“)“
X → Y •	$
Y → • “(“ Y “)”	“)“
Y → • “(“ Y “)”	“(“
Y → • “(“ Y “)”	$
Y → “(“ • Y “)”	“)“
Y → “(“ • Y “)”	“(“
Y → “(“ • Y “)”	$
Y → “(“ Y • “)”	“)“
Y → “(“ Y • “)”	“(“
Y → “(“ Y • “)”	$
Y → “(“ Y “)” •	“)“
Y → “(“ Y “)” •	“(“
Y → “(“ Y “)” •	$
Y → •	“)“
Y → •	“(“
Y → •	$

LR(1)

s0

$$\boxed{S \rightarrow \bullet X\$ \quad ?}$$

2. Build DFA
   - 1º item I in state s0

- $S \rightarrow X$ (1)
- $X \rightarrow$ "(" | Y (2, 3)
- $Y \rightarrow$ "(" Y ")" | ε (4, 5)

LR(1)

s0

$S \rightarrow \bullet X \$ \quad ?$
$X \rightarrow \bullet \text{“(“} \quad \$$
$X \rightarrow \bullet Y \quad \$$

2. Build DFA
   - 1º item I in state s0
   - Determine closure(I):
   - $X \rightarrow \bullet$ “(“ “(“
   - $X \rightarrow \bullet$ “(“ “)“
   - $X \rightarrow \bullet$ “(“ $
   - $X \rightarrow \bullet Y$ “(“
   - $X \rightarrow \bullet Y$ “)“
   - $X \rightarrow \bullet Y$ $
   - Place only the items of the Closure which lookahead symbol belongs to First($?) = {$}
   - $X \rightarrow \bullet$ “(“ $
   - $X \rightarrow \bullet Y$ $

- $S \rightarrow X$ (1)
- $X \rightarrow$ “(“ | Y (2, 3)
- $Y \rightarrow$ “(“ Y “)” | ε (4, 5)

LR(1)

s0

$S \rightarrow \bullet X \$$	?	
$X \rightarrow \bullet$ “(“	$\$$	
$X \rightarrow \bullet Y$	$\$$	

2. Build DFA
   - For each new item I in the state determine closure(I)
     - closure($X \rightarrow \bullet$ “(“ $\$$) = { }
     - closure($X \rightarrow \bullet Y$ $\$$) =
     - $Y \rightarrow \bullet$ “(“ Y “)” “)“
     - $Y \rightarrow \bullet$ “(“ Y “)” “(“
     - $Y \rightarrow \bullet$ “(“ Y “)” $\$$
     - $Y \rightarrow \bullet$ “)“
     - $Y \rightarrow \bullet$ “(“
     - $Y \rightarrow \bullet$ $\$$

- $S \rightarrow X$ (1)
- $X \rightarrow$ “(“ | Y (2, 3)
- $Y \rightarrow$ “(“ Y “)” | ε (4, 5)

LR(1)

s0

$S \to \bullet X \$$	?
$X \to \bullet$ “(“	$\$$
$X \to \bullet Y$	$\$$
$Y \to \bullet$ “(“ Y “)”	$\$$
$Y \to \bullet$	$\$$

2. Build DFA
   - For each new item I in the state determine closure(I)
   - Place only the items of the Closure which lookahead symbol belongs to First($) = {$}
     - Y → • “(“ Y “)” $
     - Y → • $

- S → X (1)
- X → “(“ | Y (2, 3)
- Y → “(“ Y “)” | ε (4, 5)

LR(1)

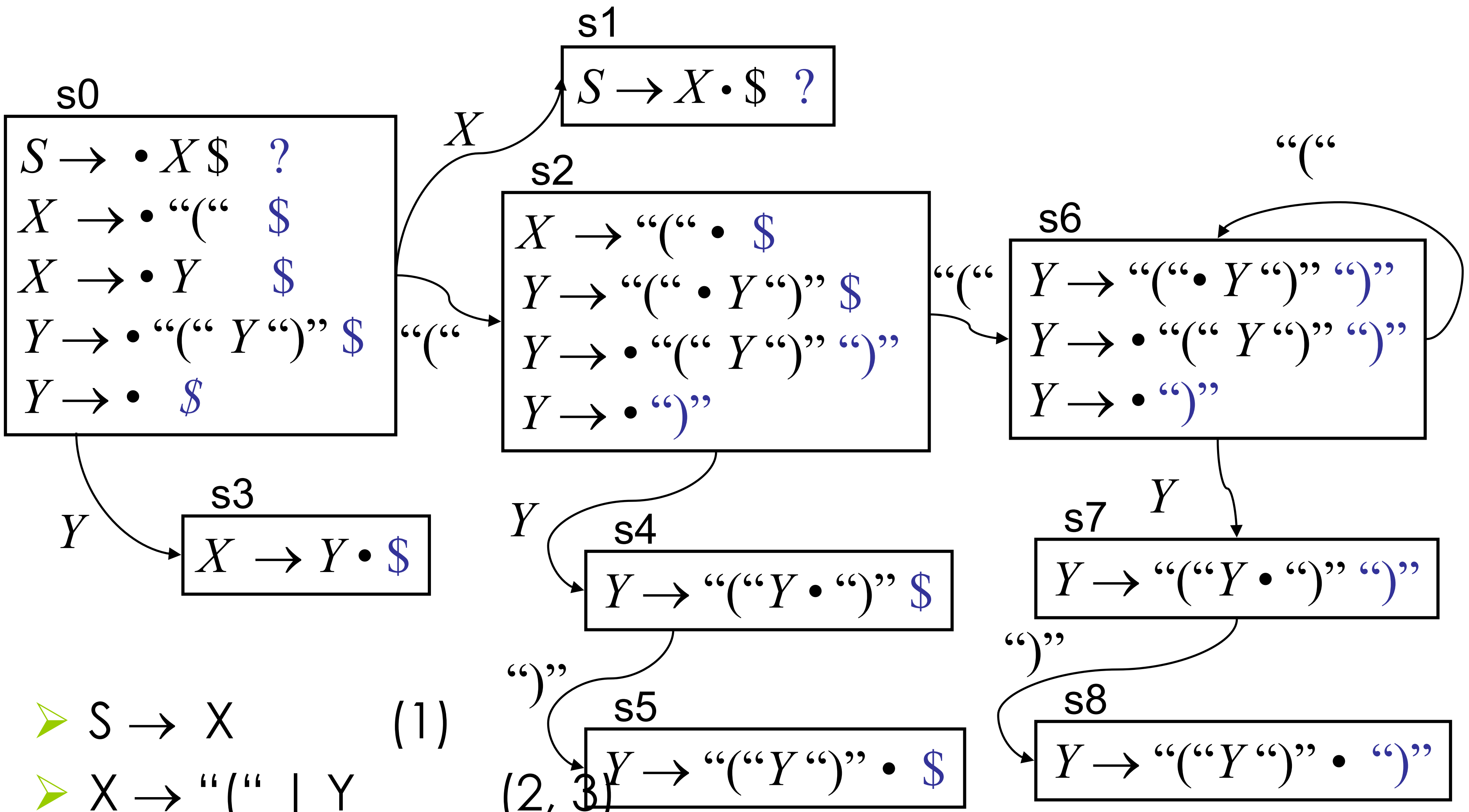
s0

$S \rightarrow \bullet X \$$	?
$X \rightarrow \bullet$ “(“	\$
$X \rightarrow \bullet Y$	\$
$Y \rightarrow \bullet$ “(“ Y “)”	\$
$Y \rightarrow \bullet$	\$

2. Buid DFA
   - Determine transitions
   - Repeat the steps to build the DFA done to the LR(0)

- $S \rightarrow X$ (1)
- $X \rightarrow$ “(“ | Y (2, 3)
- $Y \rightarrow$ “(“ Y “)” | ε (4, 5)

LR(1)



- $S \to X$ (1)
- $X \to$ "(" | Y (2, 3)
- $Y \to$ "(" Y ")" | ε (4, 5)

LR(1)

1. Identify LR(1) items
2. Build DFA
3. Build parser table
   - *Reductions are placed in the columns corresponding to the lookahead symbols in the item of each reduction*
     - *Example: The item of the s3 state:* $X \rightarrow Y \bullet \$$ implies reduction (3) in row s3, column $
4. Verify if there are conflicts

LR(1)

- Parser table

	“(“	“)”	$	X	Y
S0	Shift s2			Goto s1	goto s3
S1			Accept		
S2	Shift s6	Reduce (5)	Reduce (2)		Goto s4
S3			Reduce (3)		
S4		Shift s5			
S5			Reduce (4)		
S6	Shift s6	Reduce (5)			Goto s7
S7		Shift s8			
S8		Reduce (4)			

The grammar is LR(1) as there aren’t conflicts!

LR

- What about LR(2), LR(3), LR(k)?