

LECTURE 17

WEDNESDAY MARCH 11

CC Construction: goto

```
1  ALGORITHM: goto
2  INPUT: a set  $S$  of LR(1) items, a symbol  $x$ 
3  OUTPUT: a set of LR(1) items
4  PROCEDURE:
5  moved :=  $\emptyset$ 
6  for item  $\in S$ :
7    if item =  $[\alpha \rightarrow \beta \bullet x \delta, a]$  then
8      moved := moved  $\cup \{ [\alpha \rightarrow \beta x \bullet \delta, a] \}$ 
9    end
10 return closure(moved)
```

CC Construction: goto

R4: $[P \rightarrow \cdot (Pair), _)]$
 R5: $[\text{exercise}]$

```

1 Goal → List
2 List → List Pair
3   | Pair
4 Pair → ( Pair )
5   | ( )
    
```

CC0 = {
 1 [Goal → • List, eof]
 2 [List → • List Pair, eof]
 3 [List → • Pair, eof]
 4 [Pair → • (Pair), eof]
 5 [Pair → • (Pair), eof]
 6 [Pair → • (Pair), eof]
 7 [Pair → • (Pair), eof]
 8 [Pair → • (Pair), eof]
 9 [Pair → • (Pair), eof]
 10 [Pair → • (Pair), eof]
 }

$\text{FIRST}(> \text{af}) = \{>\}$

Calculate goto(cc₀, ().

["next state" from cc₀ taking (]

ALGORITHM: goto

INPUT: a set S of LR(1) items, a symbol x

OUTPUT: a set of LR(1) items

PROCEDURE:

moved := ∅

for item ∈ S:

if item = $[\alpha \rightarrow \beta \cdot x \delta, a]$ then

moved := moved ∪ { $[\alpha \rightarrow \beta x \cdot \delta, a]$ }

end

return closure(moved)

6 [P → (P) , eof]
 7 [P → (P) , (]
 8 [P → (.) , eof]
 9 [P → (.) , (]

CC Construction: Algorithm

```
1  ALGORITHM: BuildCC
2  INPUT: a grammar  $G = (V, \Sigma, R, S)$ , goal production  $S \rightarrow S$ 
3  OUTPUT:
4  → (1) a set  $CC = \{cc_0, cc_1, \dots, cc_n\}$  where  $cc_i \subseteq G$ 's LR(1) items
5  → (2) a transition function
6  PROCEDURE:
7   $cc_0 := \text{closure}(\{[S' \rightarrow \bullet S \text{ eof}]\})$  → initial state ( $cc_0$ ) to start with.
8   $CC := \{cc_0\}$ 
9   $processed := \{cc_0\}$ 
10  $lastCC := \emptyset$ 
11 while  $lastCC \neq CC$ :
12    $lastCC := CC$ 
13   for  $cc_j$  s.t.  $cc_j \in CC \wedge cc_j \notin processed$ :
14      $processed := processed \cup \{cc_j\}$ 
15     for  $x$  s.t.  $[\dots \rightarrow \dots \bullet x \dots] \in cc_j$ :
16        $temp := \text{goto}(cc_j, x)$  → transition
17       if  $temp \notin CC$  then
18          $CC := CC \cup \{temp\}$ 
19       end
20    $\delta := \delta \cup (cc_j, x, temp)$ 
```

resulting set

transition

temp

CC

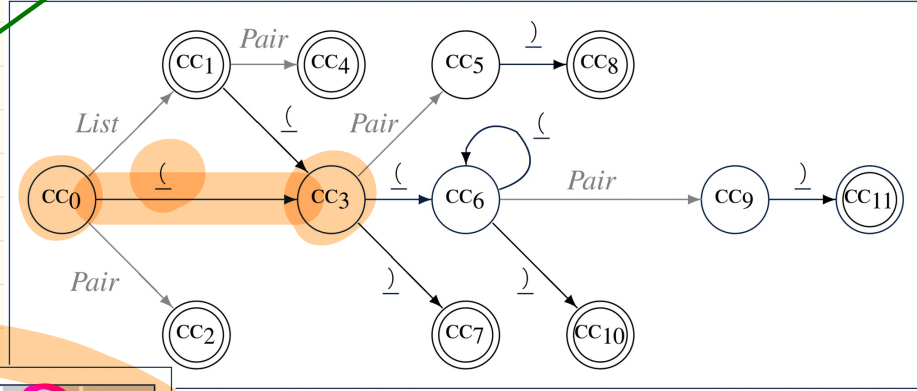
π

CC Construction: Algorithm

1	$Goal \rightarrow List$
2	$List \rightarrow List\ Pair$
3	$\quad \quad Pair$
4	$Pair \rightarrow (\quad Pair \quad)$
5	$\quad \quad (\quad)$

- Calculate $\mathcal{CC} = \{cc_0, cc_1, \dots, cc_{11}\}$
- Calculate the transition function $\delta : \mathcal{CC} \times \Sigma \rightarrow \mathcal{CC}$

CC Construction: Algorithm



close (just pop)

Iteration	Item	Goal	List	Pair	()	eof
0	CC0	\emptyset	CC1	CC2	CC3	\emptyset	\emptyset
1	CC1	\emptyset	\emptyset	CC4	CC3	\emptyset	\emptyset
	CC2	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
	CC3	\emptyset	\emptyset	CC5	CC6	CC7	\emptyset
2	CC4	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
	CC5	\emptyset	\emptyset	\emptyset	\emptyset	CC8	\emptyset
	CC6	\emptyset	\emptyset	CC9	CC6	CC10	\emptyset
	CC7	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
3	CC8	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
	CC9	\emptyset	\emptyset	\emptyset	\emptyset	CC11	\emptyset
	CC10	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
4	CC11	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

goto (CC0, (

CC Construction: Algorithm

$$CC_0 = \left\{ \begin{array}{lll} [Goal \rightarrow \bullet List, eof] & [List \rightarrow \bullet List Pair, eof] & [List \rightarrow \bullet List Pair, _] \\ [List \rightarrow \bullet Pair, eof] & [List \rightarrow \bullet Pair, _] & [Pair \rightarrow \bullet _ Pair _, eof] \\ [Pair \rightarrow \bullet _ Pair _, _] & [Pair \rightarrow \bullet _, eof] & [Pair \rightarrow \bullet _, _] \end{array} \right\}$$

$$CC_2 = \left\{ [List \rightarrow Pair \bullet, eof] \quad [List \rightarrow Pair \bullet, _] \right\}$$

$$CC_4 = \left\{ [List \rightarrow List Pair \bullet, eof] \quad [List \rightarrow List Pair \bullet, _] \right\}$$

$$CC_6 = \left\{ \begin{array}{ll} [Pair \rightarrow \bullet _ Pair _, _] & [Pair \rightarrow _ \bullet Pair _, _] \\ [Pair \rightarrow \bullet _ _, _] & [Pair \rightarrow _ \bullet _, _] \end{array} \right\}$$

CC₈

$$CC_8 = \left\{ [Pair \rightarrow _ Pair _, \bullet, eof] \quad [Pair \rightarrow _ Pair _, \bullet, _] \right\}$$

$$CC_{10} = \left\{ [Pair \rightarrow _ _ \bullet, _] \right\}$$

$$CC_1 = \left\{ \begin{array}{lll} [Goal \rightarrow List \bullet, eof] & [List \rightarrow List \bullet Pair, eof] & [List \rightarrow List \bullet Pair, _] \\ [Pair \rightarrow \bullet _ Pair _, eof] & [Pair \rightarrow \bullet _ Pair _, _] & [Pair \rightarrow \bullet _, eof] \\ & [Pair \rightarrow \bullet _, _] & \end{array} \right\}$$

$$CC_3 = \left\{ \begin{array}{lll} [Pair \rightarrow \bullet _ Pair _, _] & [Pair \rightarrow _ \bullet Pair _, eof] & [Pair \rightarrow _ \bullet Pair _, _] \\ [Pair \rightarrow \bullet _, _] & [Pair \rightarrow _ \bullet _, eof] & [Pair \rightarrow _ \bullet _, _] \end{array} \right\}$$

$$CC_5 = \left\{ [Pair \rightarrow _ Pair \bullet _, eof] \quad [Pair \rightarrow _ Pair \bullet _, _] \right\}$$

$$CC_7 = \left\{ [Pair \rightarrow _ _ \bullet, eof] \quad [Pair \rightarrow _ _ \bullet, _] \right\}$$

$$CC_9 = \left\{ [Pair \rightarrow _ Pair \bullet _, _] \right\}$$

$$CC_{11} = \left\{ [Pair \rightarrow _ Pair _, \bullet, _] \right\}$$

Table Construction: Algorithm

ALGORITHM: BuildActionGotoTables

INPUT:

- (1) a grammar $G = (V, \Sigma, R, S)$
- (2) goal production $S \rightarrow S'$
- (3) a canonical collection $CC = \{cc_0, cc_1, \dots, cc_n\}$
- (4) a transition function $\delta: CC \times \Sigma \rightarrow CC$

OUTPUT: Action Table & Goto Table

PROCEDURE:

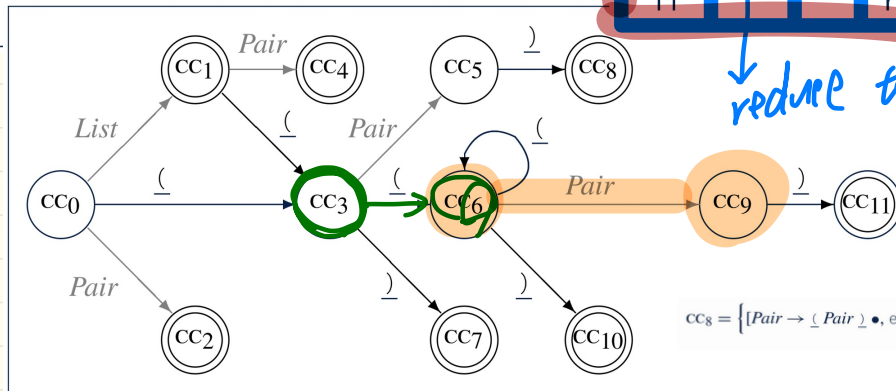
```

for  $cc_i \in CC$ :
  for item  $\in cc_i$ :
    if item =  $[A \rightarrow \beta \bullet xy, a]$  &  $\text{pause} \delta(cc_i, x) = cc_j$  then
      Action[i, x] := shift(j)
    elseif item =  $[A \rightarrow \beta \bullet a]$  then
      Action[i, a] := reduce A  $\rightarrow$   $\beta$ 
    elseif item =  $[S \rightarrow S' \bullet, \text{eof}]$  then
      Action[i, eof] := accept
    end
  end
  for  $v \in V$ :
    if  $\delta(cc_i, v) = cc_j$  then
      Goto[i, v] = j
    end
  end
end
    
```

Handwritten notes:
 - "may be ϵ " with an arrow pointing to the xy in the first item.
 - "start variable" with an arrow pointing to the S in the third item.
 - "CC₃" with an arrow pointing to the cc_3 state in the diagram below.

State	Action Table			Goto Table	
	eof	()	List	Pair
0		s 3		1	2
1	acc	s 3			4
2	r 3	r 3			
3		s 6	s 7		5
4	r 2	r 2			
5			s 8		
6		s 6	s 10		9
7					
8		r 5	r 5		
9		r 4	r 4		
10			s 11		
11			r 5		
12			r 4		

Handwritten notes on the table:
 - Green circles around the (and) symbols in the header and in the first row.
 - Red circles around the "acc" and "s 3" in row 1.
 - Blue circles around the "r 5" and "r 4" in row 8.
 - Orange circle around the "s 6" in row 6.
 - Blue arrow pointing from the "r 4" in row 9 to the "reduce to R4" note.
 - Blue arrow pointing from the "r 4" in row 12 to the "P → (Pair)" note.



$CC_8 = \{ [Pair \rightarrow _ Pair _ \bullet, \text{eof}] \quad [Pair \rightarrow _ Pair _ \bullet, _] \}$

TDP

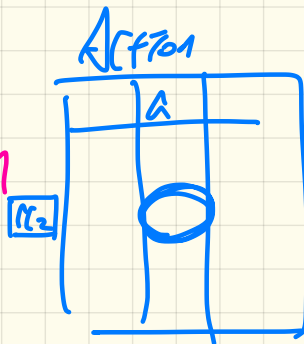
↓ g
no left rec.

LR(1)

↓ back track
free

BRP

↓ skeleton
grammar



LR(1) items

CC (possible states of NFA)

↓
CC₀ - - -
transition

↓ Actions
how

→ deterministic

Act 13

13		

$$CC_{13} = \left\{ \begin{aligned} &[Stmt \rightarrow \text{if expr then } Stmt \bullet, \{eof, else\}], \\ &[Stmt \rightarrow \text{if expr then } Stmt \bullet \text{ else } Stmt, \{eof, else\}] \end{aligned} \right\}$$

word
else

① [Stmt → if expr then Stmt • , eof]

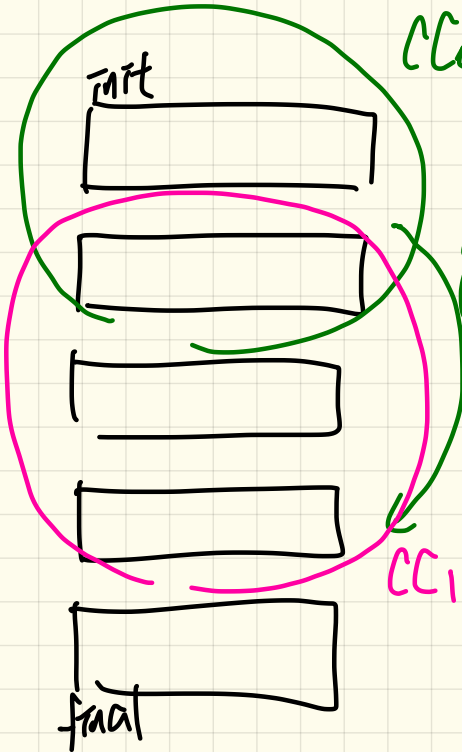
② [Stmt → if expr then Stmt • , else]

③ [Stmt → if expr then Stmt • else Stmt , eof]

④ [Stmt → if expr then Stmt • else Stmt , else]

$LR(1)$ items

CC_0



$goto(CC_0, x)$

CC_1

CC

each member is
a subset of
 $LR(1)$ items.

