Lecture 17

Wednesday March 11

CC Construction: goto

```
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 = [α → β • x δ, a] then
      moved := moved ∪ { [α → β x • δ, a] }
    end
  return closure(moved)
```

CC Construction: goto

R4 : [P → •(Pair),)]
R5 : [exercise]

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

✓

$$cc_0 = \left\{ \begin{array}{lll} \overset{1}{[Goal \to \bullet List, \texttt{eof}]} & \overset{2}{[List \to \bullet List\ Pair, \texttt{eof}]} & \overset{3}{[List \to \bullet List\ Pair, (]} \\ \overset{4}{[List \to \bullet Pair, \texttt{eof}]} & \overset{5}{[List \to \bullet Pair, (]} & \overset{6}{[Pair \to \bullet (\ Pair\ ), \texttt{eof}]} \\ \overset{7}{[Pair \to \bullet (\ Pair\ ), (]} & \overset{8}{[Pair \to \bullet (\ ), \texttt{eof}]} & \overset{9}{[Pair \to \bullet (\ ), (]} \end{array} \right\}$$

FIRST( ) eof) = {) }

src

Calculate *goto*(cc_0, ( ). ["next state" from cc_0 taking ( ]

δ a

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

```
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 = [α → β • x δ, a] then
      moved := moved ∪ { [α → β x • δ, a] }
    end
  return closure(moved)
```

CC Construction: Algorithm

```
ALGORITHM: BuildCC
  INPUT: a grammar G = (V, Σ, R, S), goal production S → S'
  OUTPUT:
  (1) a set CC = {cc0, cc1, ..., ccn} where cci ⊆ G's LR(1) items
  (2) a transition function
PROCEDURE:
  cc0 := closure({[S' → •S, eof]})
  CC := {cc0}
  processed := {cc0}
  lastCC := ∅
  while (lastCC ≠ CC) :
    lastCC := CC
    for cci s.t. cci ∈ CC ∧ cci ∉ processed:
      processed := processed ∪ {cci}
      for x s.t. [··· → ··· • x ...] ∈ cci
        temp := goto(cci, x)
        if temp ∉ CC then
          CC := CC ∪ {temp}
        end
        δ := δ ∪ (cci, x, temp)
```

initial state (cc_0) to start with.

$cc_i \xrightarrow{x} temp$

transition

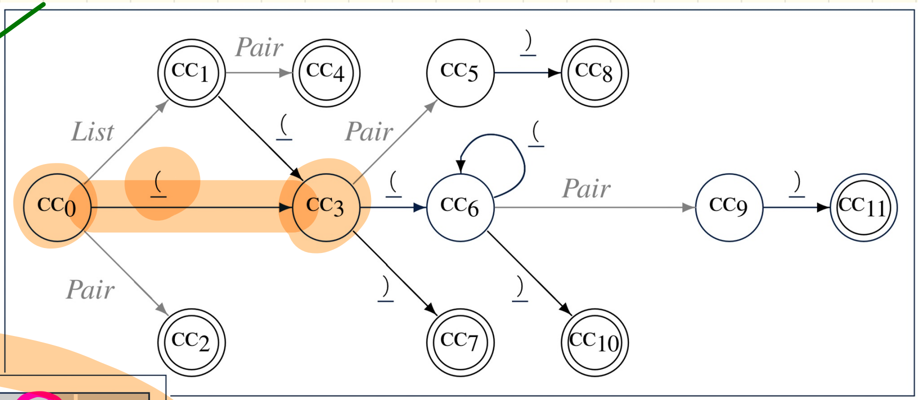
resulting set

CC Construction: Algorithm

1	*Goal*	$\rightarrow$	*List*
2	*List*	$\rightarrow$	*List Pair*
3		$\mid$	*Pair*
4	*Pair*	$\rightarrow$	$\underline{(}$ *Pair* $\underline{)}$
5		$\mid$	$\underline{(}\ \underline{)}$

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

CC Construction: Algorithm



closure (init item)

Iteration	Item	Goal	List	Pair	(	)	eof
0	cc_0	∅	cc_1	cc_2	cc_3	∅	∅
1	cc_1	∅	∅	cc_4	cc_3	∅	∅
	cc_2	∅	∅	∅	∅	∅	∅
	cc_3	∅	∅	cc_5	cc_6	cc_7	∅
2	cc_4	∅	∅	∅	∅	∅	∅
	cc_5	∅	∅	∅	∅	cc_8	∅
	cc_6	∅	∅	cc_9	cc_6	cc_{10}	∅
	cc_7	∅	∅	∅	∅	∅	∅
3	cc_8	∅	∅	∅	∅	∅	∅
	cc_9	∅	∅	∅	∅	cc_{11}	∅
	cc_{10}	∅	∅	∅	∅	∅	∅
4	cc_{11}	∅	∅	∅	∅	∅	∅

goto(cc0 , ( )

CC Construction: Algorithm

$$CC_0 = \left\{\begin{array}{ccc} [Goal \rightarrow \bullet\, List,\ \texttt{eof}] & [List \rightarrow \bullet\, List\ Pair,\ \texttt{eof}] & [List \rightarrow \bullet\, List\ Pair,\ \underline{(}] \\ [List \rightarrow \bullet\, Pair,\ \texttt{eof}] & [List \rightarrow \bullet\, Pair,\ \underline{(}] & [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)},\ \texttt{eof}] \\ [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)},\underline{(}] & [Pair \rightarrow \bullet\, \underline{(}\ \underline{)},\ \texttt{eof}] & [Pair \rightarrow \bullet\, \underline{(}\ \underline{)},\underline{(}] \end{array}\right\}$$

$$CC_1 = \left\{\begin{array}{ccc} [Goal \rightarrow List\, \bullet,\ \texttt{eof}] & [List \rightarrow List\, \bullet\, Pair,\ \texttt{eof}] & [List \rightarrow List\, \bullet\, Pair,\ \underline{(}] \\ [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)},\ \texttt{eof}] & [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)},\ \underline{(}] & [Pair \rightarrow \bullet\, \underline{(}\ \underline{)},\ \texttt{eof}] \\ & [Pair \rightarrow \bullet\, \underline{(}\ \underline{)},\ \underline{(}] & \end{array}\right\}$$

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

$$CC_3 = \left\{\begin{array}{ccc} [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)},\ \underline{)}] & [Pair \rightarrow \underline{(}\, \bullet\, Pair\ \underline{)},\ \texttt{eof}] & [Pair \rightarrow \underline{(}\, \bullet\, Pair\ \underline{)},\ \underline{(}] \\ [Pair \rightarrow \bullet\, \underline{(}\ \underline{)},\ \underline{)}] & [Pair \rightarrow \underline{(}\, \bullet\, \underline{)},\ \texttt{eof}] & [Pair \rightarrow \underline{(}\, \bullet\, \underline{)},\ \underline{(}] \end{array}\right\}$$

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

$$CC_5 = \left\{[Pair \rightarrow \underline{(}\ Pair\, \bullet\ \underline{)},\ \texttt{eof}]\quad [Pair \rightarrow \underline{(}\ Pair\, \bullet\ \underline{)},\ \underline{(}]\right\}$$

$$CC_6 = \left\{\begin{array}{cc} [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)},\ \underline{)}] & [Pair \rightarrow \underline{(}\, \bullet\, Pair\ \underline{)},\ \underline{)}] \\ [Pair \rightarrow \bullet\, \underline{(}\ \underline{)},\ \underline{)}] & [Pair \rightarrow \underline{(}\, \bullet\, \underline{)},\ \underline{)}] \end{array}\right\}$$

$$CC_7 = \left\{[Pair \rightarrow \underline{(}\ \underline{)}\, \bullet,\ \texttt{eof}]\quad [Pair \rightarrow \underline{(}\ \underline{)}\, \bullet,\ \underline{(}]\right\}$$

CC_8

$$CC_8 = \left\{[Pair \rightarrow \underline{(}\ Pair\ \underline{)}\, \bullet,\ \texttt{eof}]\quad [Pair \rightarrow \underline{(}\ Pair\ \underline{)}\, \bullet,\ \underline{(}]\right\}$$

$$CC_9 = \left\{[Pair \rightarrow \underline{(}\ Pair\, \bullet\ \underline{)},\ \underline{)}]\right\}$$

$$CC_{10} = \left\{[Pair \rightarrow \underline{(}\ \underline{)}\, \bullet,\ \underline{)}]\right\}$$

$$CC_{11} = \left\{[Pair \rightarrow \underline{(}\ Pair\ \underline{)}\, \bullet,\ \underline{)}]\right\}$$

Table Construction: Algorithm

```
ALGORITHM: BuildActionGotoTables
  INPUT:
    (1) a grammar G = (V, Σ, R, S)
    (2) goal production S → S'
    (3) a canonical collection CC = {cc0, cc1, ..., ccn}
    (4) a transition function δ : CC × Σ → CC
  OUTPUT: Action Table & Goto Table
PROCEDURE:
  for cci ∈ CC :
    for item ∈ cci :
      if item = [A → β•xγ, a]\pause∧δ(cci, x) = ccj then
        Action[i, x] := shift j
      elseif item = [A → β•, a] then
        Action[i, a] := reduce A → β
      elseif item = [S → S'•, eof] then
        Action[i, eof] := accept
      end
    for v ∈ V :
      if δ(cci, v) = ccj then
        Goto[i, v] = j
      end
```

may be ε

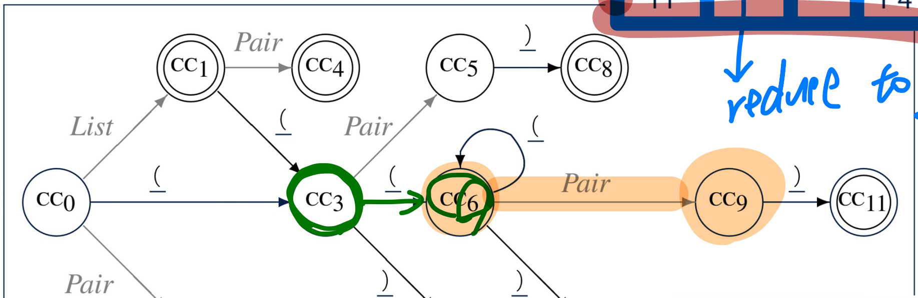
start variable

	Action Table			Goto Table	
State	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	r 5	r 5			
8	r 4	r 4			
9			s 11		
10			r 5		
11			r 4		

CC_3

reduce to R4

$P \to (\text{Pair})$



$cc_8 = \{[Pair \to (\ Pair\ )\ \bullet, \text{eof}]\quad [Pair \to (\ Pair\ )\ \bullet, (]\}$

TDP

↓ g

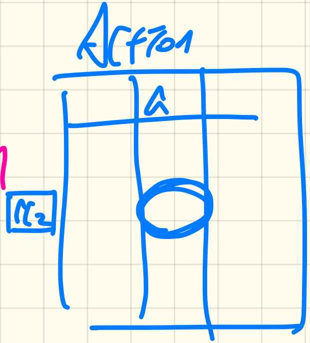
no left rec.

LR(1)

↓ backtrack free

BUP

↓ skeleton grammar



LR(1) items

CC ($\approx$ possible states of NFA)

↓

$\underline{CC_0}$ - - -

transition

↓ Actions, Goto → deterministic

Action

13

$$CC_{13} = \left\{ \begin{array}{l} [Stmt \rightarrow \texttt{if expr then } Stmt \bullet, \{\texttt{eof}, \texttt{else}\}], \\ [Stmt \rightarrow \texttt{if expr then } Stmt \bullet \texttt{ else } Stmt, \{\texttt{eof}, \texttt{else}\}] \end{array} \right\}$$

① [Stmt → if expr then Stmt •, eof]

r.

word

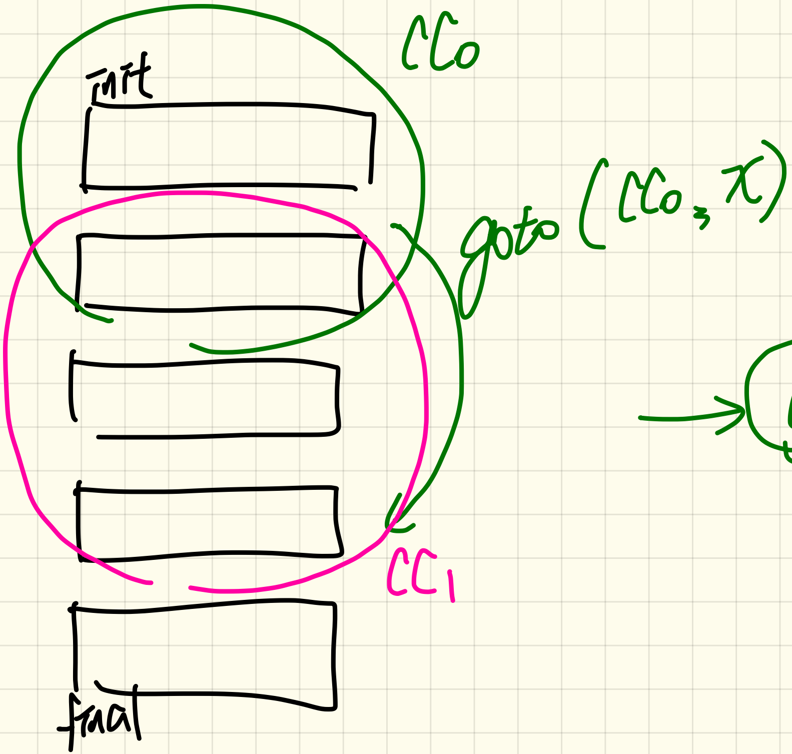
else

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



$\underline{CC}$

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

$$\rightarrow CC_0 \xrightarrow{x} CC_1$$