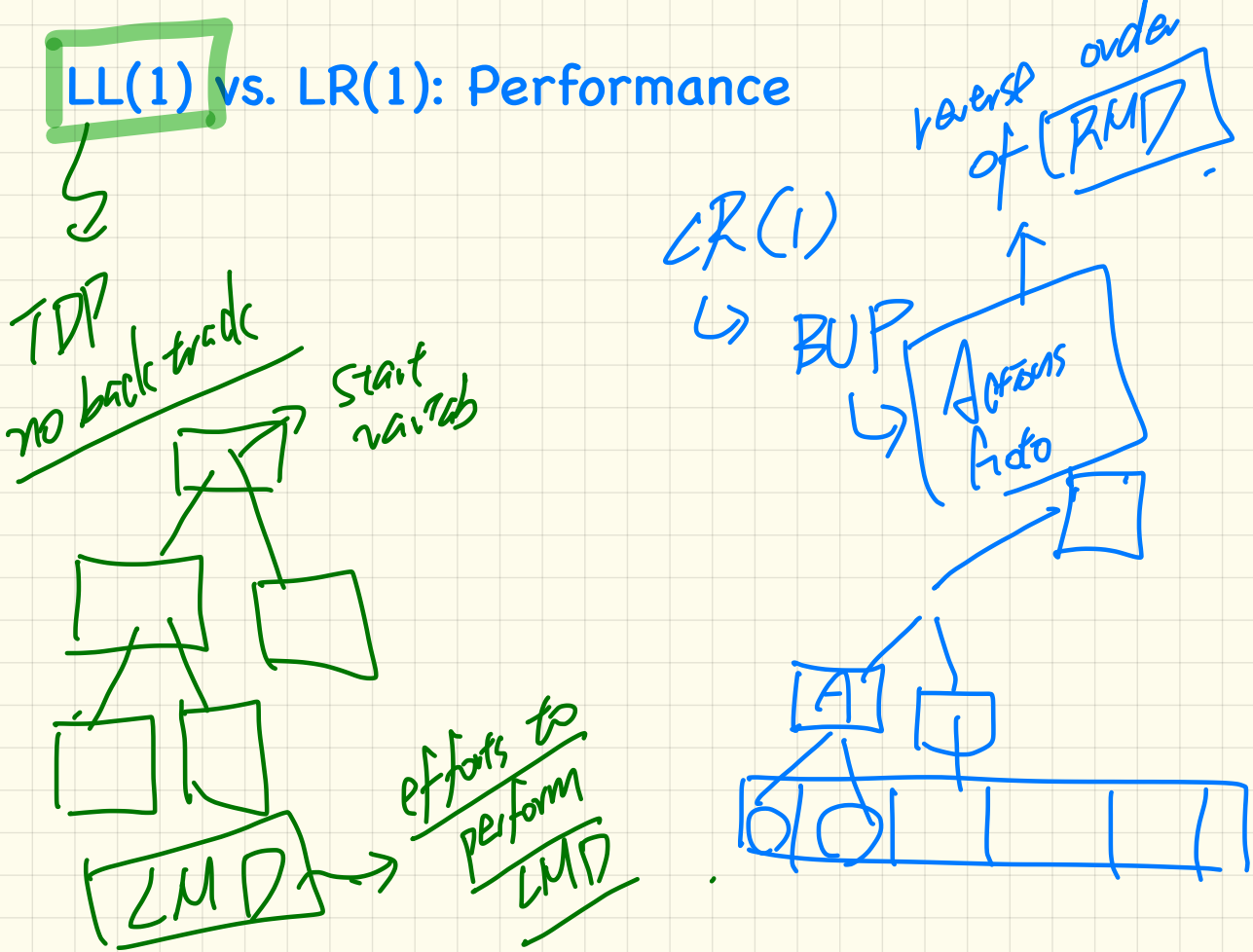
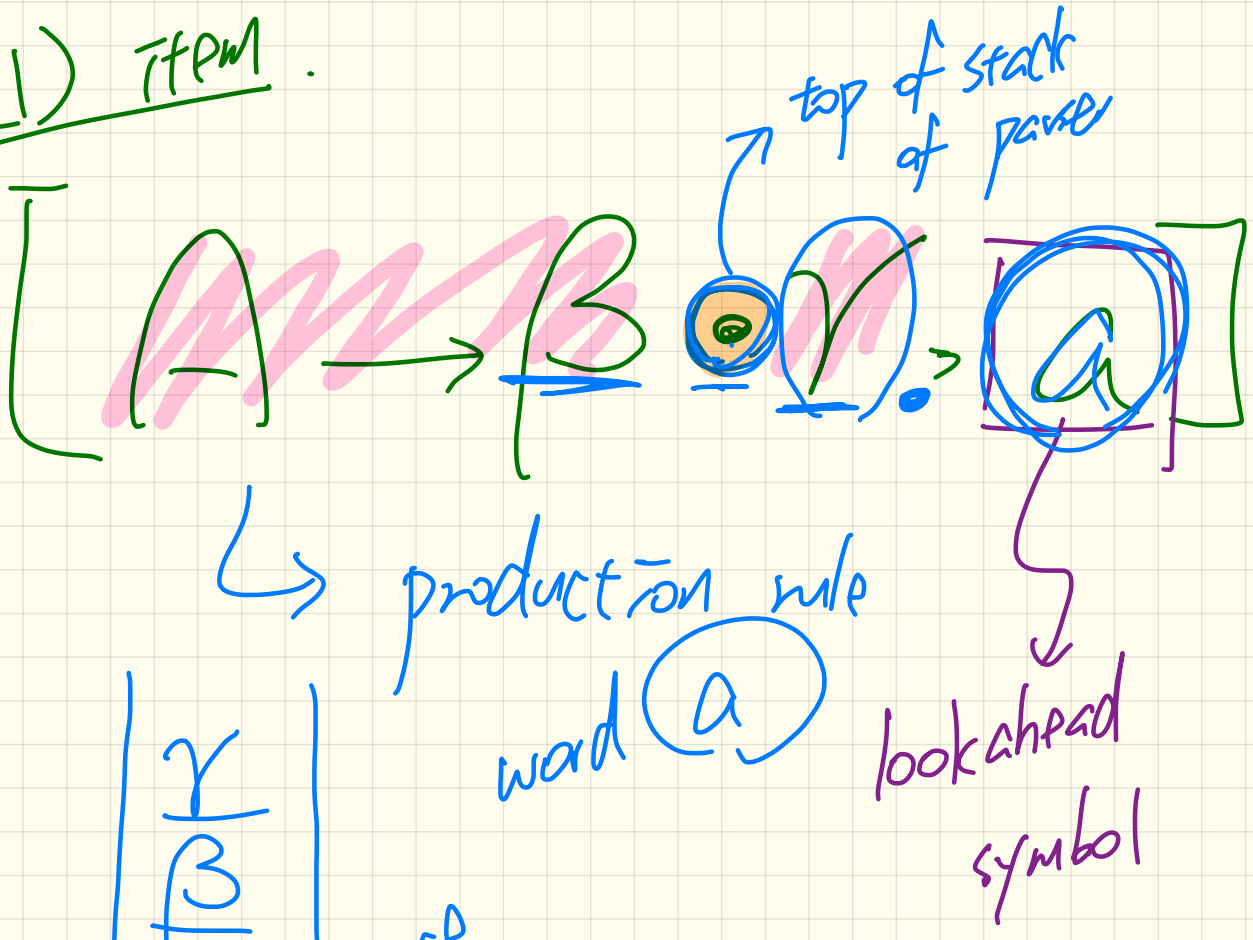
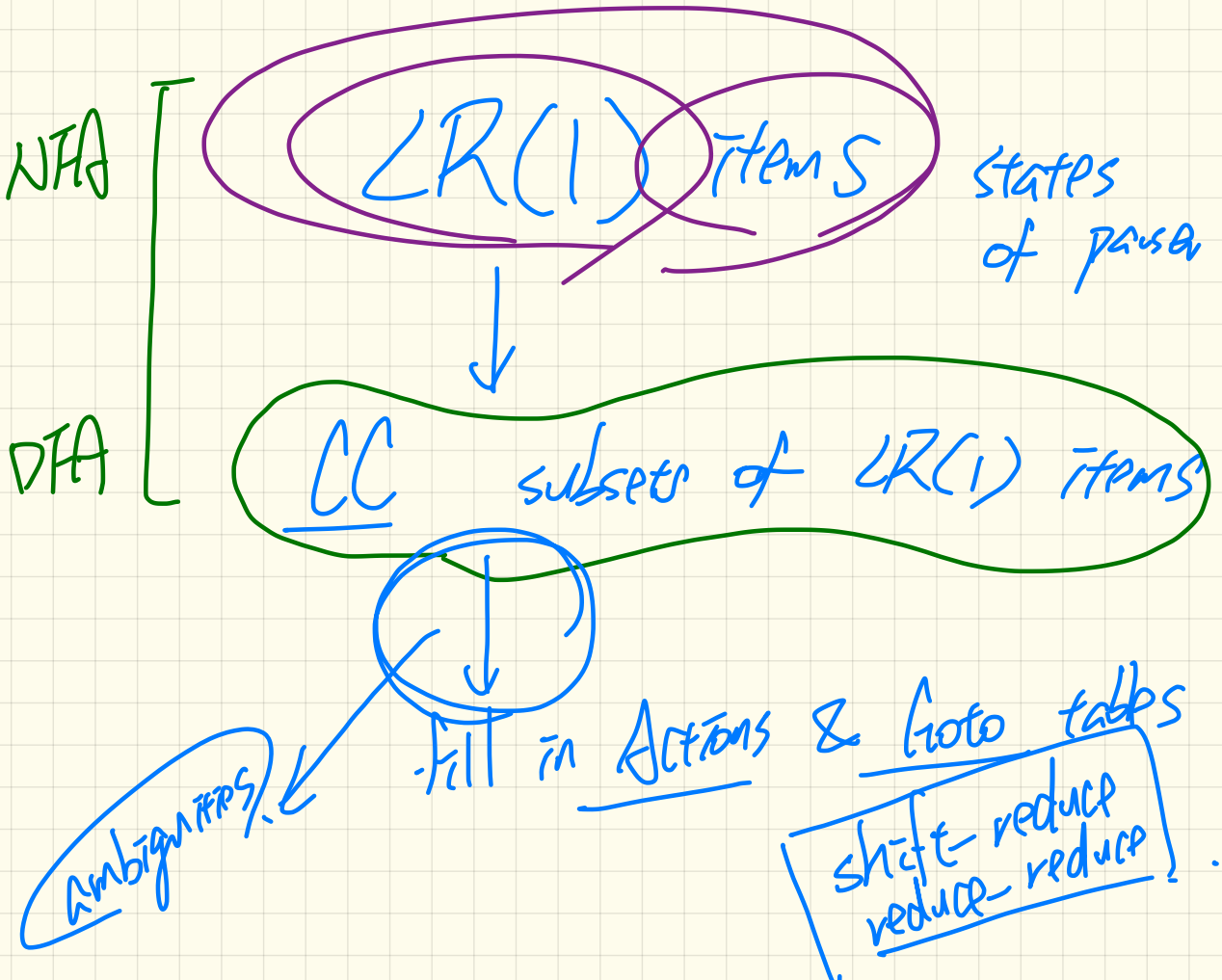
Lecture 16
Monday March 9

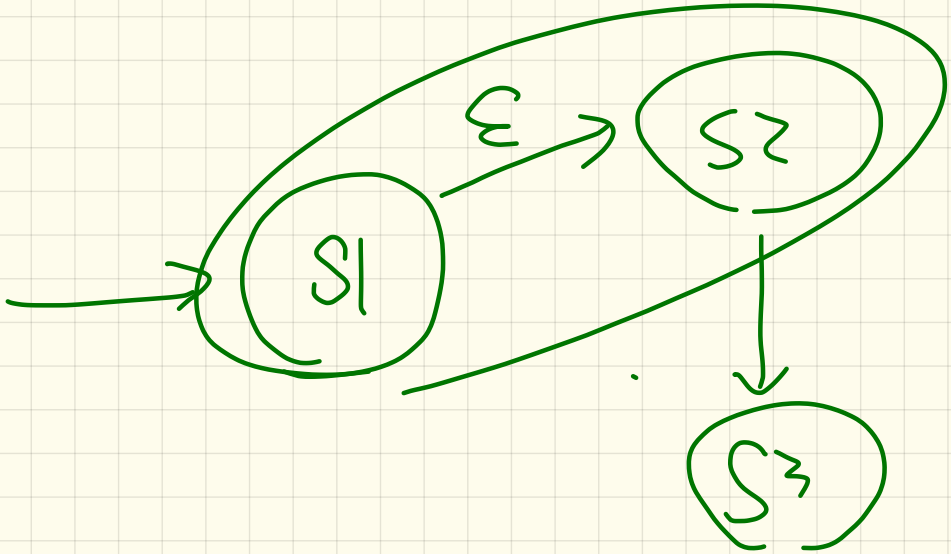
LL(1) vs. LR(1): Performance

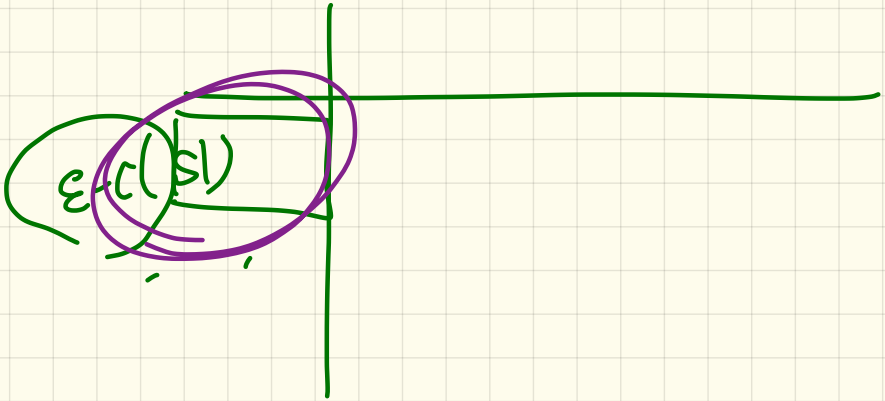


LR(1) item.


top of stack of parser
production rule
word a
lookahead symbol
γ
β
track


NFA
LR(1) items
states of parser
DFA
CC
subsets of LR(1) items
Fill in Actions & Goto tables
shift-reduce
reduce-reduce
ambiguities


ε
S2
S1
S3


ε-c(S1)

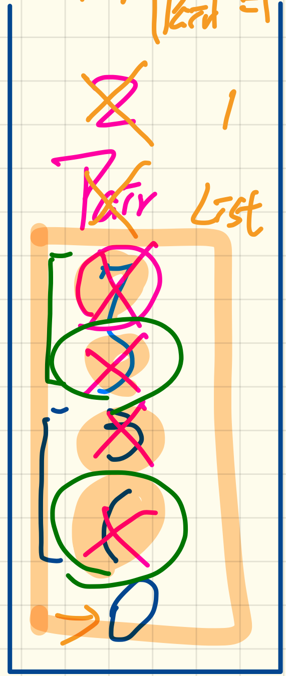
Bottom-Up Parsing: Discovering Rightmost Derivations (1)

```
ALGORITHM: BUParse
  INPUT: CFG G = (V, Σ, R, S), Action & Goto Tables
  OUTPUT: Report Parse Success or Syntax Error
PROCEDURE:
  initialize an empty stack trace
  trace.push(S) /* start state */
  word := NextWord()
  while(true)
    state := trace.top()
    act := Action[state, word]
    if act = ''accept'' then
      succeed()
    elseif act = ''reduce A → β'' then
      trace.pop() 2 × |β| times /* word + state */
      state := trace.top()
      trace.push(A)
      next := Goto[state, A]
      trace.push(next)
    elseif act = ''shift s_i'' then
      trace.push(word)
      trace.push(s_i)
      word := NextWord()
    else
      fail()
```

Parse: ( )

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

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	r 5	r 5			
8	r 4	r 4			
9			s 11		
10			r 5		
11			r 4		



LR(1) Items: Exercise (1.1)

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

Initial State: [*Goal* → •*List*, `eof`]

Desired Final State: [*Goal* → *List*•, `eof`]

Intermediate States: Subset Construction

Q. Derive all *LR(1) items* for the above grammar.

- **FOLLOW**(*List*) = {`eof`, `(`} **FOLLOW**(*Pair*) = {`eof`, `(`, `)`}

[• (Pair), eof]

[• (Pair), (]

[• (Pair),)]

LR(1) Items: Exercise (1.2)

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

FOLLOW(*List*) = {eof, (} **FOLLOW**(*Pair*) = {eof, (,)}

$[Goal \rightarrow \bullet\ List, \texttt{eof}]$

$[Goal \rightarrow List\ \bullet, \texttt{eof}]$

$[List \rightarrow \bullet\ List\ Pair, \texttt{eof}]$	$[List \rightarrow \bullet\ List\ Pair, (\ ]$
$[List \rightarrow List \bullet Pair, \texttt{eof}]$	$[List \rightarrow List \bullet Pair, (\ ]$
$[List \rightarrow List\ Pair \bullet, \texttt{eof}]$	$[List \rightarrow List\ Pair \bullet, (\ ]$
$[List \rightarrow \bullet\ Pair, \texttt{eof}\ ]$	$[List \rightarrow \bullet\ Pair, (\ ]$
$[List \rightarrow Pair \bullet, \texttt{eof}\ ]$	$[List \rightarrow Pair \bullet, (\ ]$

$[Pair \rightarrow \bullet\ (\ Pair\ ), \texttt{eof}\ ]$	$[Pair \rightarrow \bullet\ (\ Pair\ ),)]$	$[Pair \rightarrow \bullet\ (\ Pair\ ), (]$
$[Pair \rightarrow (\ \bullet\ Pair\ ), \texttt{eof}\ ]$	$[Pair \rightarrow (\ \bullet\ Pair\ ),)]$	$[Pair \rightarrow (\ \bullet\ Pair\ ), (]$
$[Pair \rightarrow (\ Pair \bullet\ ), \texttt{eof}\ ]$	$[Pair \rightarrow (\ Pair \bullet\ ),)]$	$[Pair \rightarrow (\ Pair \bullet\ ), (]$
$[Pair \rightarrow (\ Pair\ ) \bullet, \texttt{eof}\ ]$	$[Pair \rightarrow (\ Pair\ ) \bullet,)]$	$[Pair \rightarrow (\ Pair\ ) \bullet, (]$
$[Pair \rightarrow \bullet\ (\ ), \texttt{eof}]$	$[Pair \rightarrow \bullet\ (\ ), (]$	$[Pair \rightarrow \bullet\ (\ ),)]$
$[Pair \rightarrow (\ \bullet\ ), \texttt{eof}]$	$[Pair \rightarrow (\ \bullet\ ), (]$	$[Pair \rightarrow (\ \bullet\ ),)]$
$[Pair \rightarrow (\ )\ \bullet, \texttt{eof}]$	$[Pair \rightarrow (\ )\ \bullet, (]$	$[Pair \rightarrow (\ )\ \bullet,)]$

LR(1) Items: Exercise (2)

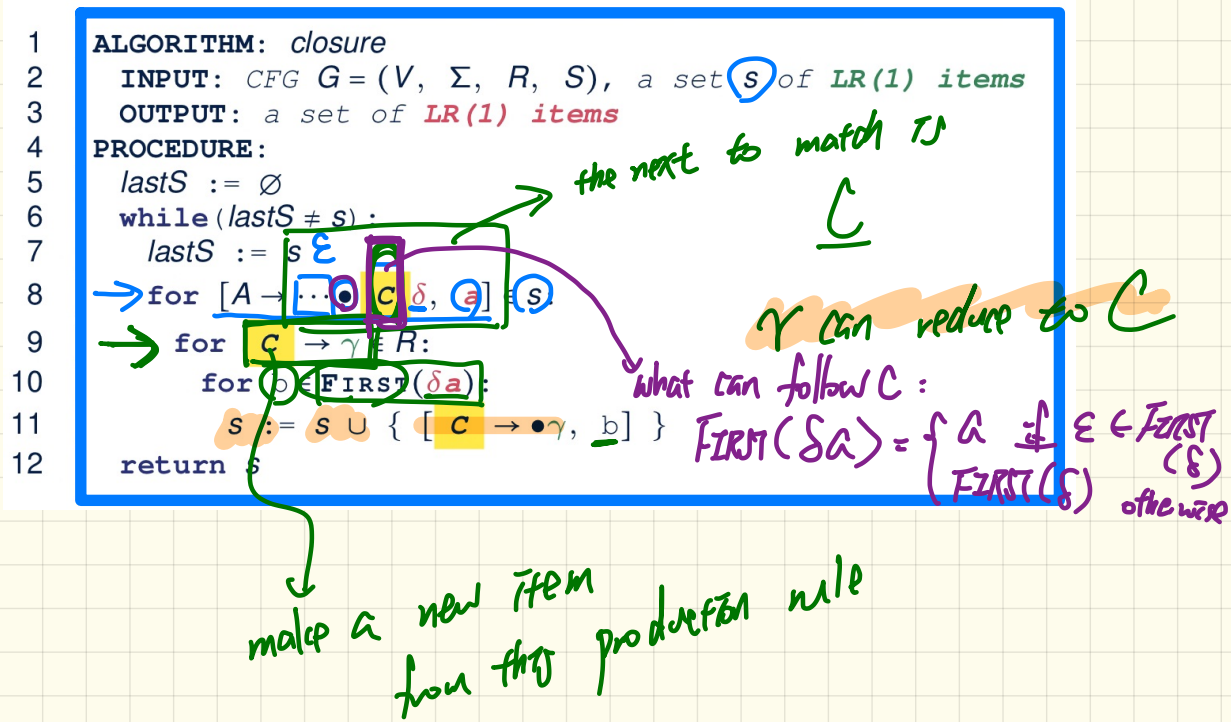
0	*Goal*	→	*Expr*	6	*Term′*	→	x *Factor Term′*
1	*Expr*	→	*Term Expr′*	7		|	÷ *Factor Term′*
2	*Expr′*	→	+ *Term Expr′*	8		|	ϵ
3		|	- *Term Expr′*	9	*Factor*	→	(*Expr*)
4		|	ϵ	10		|	num
5	*Term*	→	*Factor Term′*	11		|	name

FOLLOW Set

	Expr	Expr'	Term	Term'	Factor
FOLLOW	eof,)	eof,)	eof, +, -,)	eof, +, -,)	eof, +, -, x, ÷,)

| LR(1) items of R2 | = 4 x 2 = 8.

CC Construction: closure

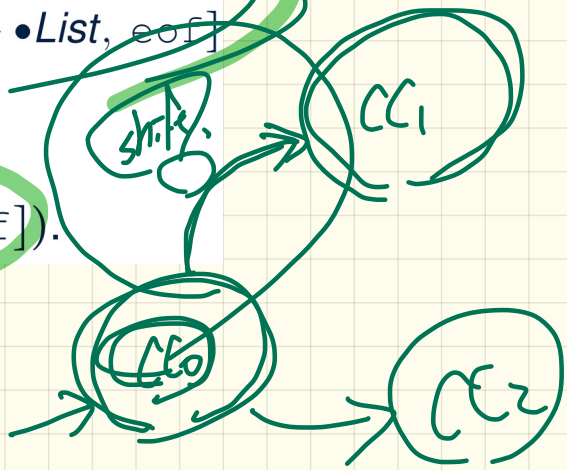


CC Construction: closure

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

Initial State: $[Goal \rightarrow \bullet List,\ \texttt{eof}]$

Calculate $cc_0 = closure([Goal \rightarrow \bullet List,\ \texttt{eof}])$.




```
1  ALGORITHM: closure
2    INPUT: CFG G = (V, Σ, R, S), a set s of LR(1) items
3    OUTPUT: a set of LR(1) items
4  PROCEDURE:
5    lastS := ∅
6    while (lastS ≠ s) :
7      lastS := s
8      for [A → ··· • C δ, a] ∈ s:
9        for C → γ ∈ R:
10         for b ∈ FIRST(δa):
11           s := s ∪ { [ C → •γ, b] }
12   return s
```

Goal ε List (ε) eof

[A → B • C δ, a]

① [Goal → • List, eof] initial

FIRST(δa) = FIRST(ε eof) = {eof}

Step 1

① [List → • List Pair, eof]

② [List → • Pair, eof]

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

```
ALGORITHM: closure
  INPUT: CFG G = (V, Σ, R, S), a set s of LR(1) items
  OUTPUT: a set of LR(1) items
PROCEDURE:
  lastS := ∅
  while (lastS ≠ s) :
    lastS := s
    for [A → ··· • C δ, a] ∈ s:
      for C → γ ∈ R:
        for b ∈ FIRST(δa):
          s := s ∪ { [C → • γ, b] }
  return s
```

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

Step 2

⓪ [Goal → • List , eof]

① [List → • List Pair , eof]

② [List → • Pair , eof]

[A → B • C δ , a]

ε

C = Pair

δ → ε

FIRST(δa) = FIRST(ε eof) = { eof }

③ [Pair → • (Pair) , eof]

④ [Pair → • ( ) , eof]

$[A \to B \cdot CS, a]$

① $[Goal \to \cdot List, eof]$

① $[List \to \cdot List\ Pair, eof]$

② $[List \to \cdot Pair, eof]$

③ $[Pair \to \cdot (Pair), eof]$

④ $[Pair \to \cdot (\ ), eof]$

FIRST(Sa) =

1	$Goal \to List$
2	$List \to List\ Pair$
3	$\mid Pair$
4	$Pair \to (\ Pair\ )$
5	$\mid (\ )$

Step 3

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

$[A \to B \cdot CS, a]$

① [Goal → • List, eof]

① [List → • List Pair, eof]

② [List → • Pair, eof]

③ [Pair → • (Pair), eof]

④ [Pair → • ( ), eof]

⑤ [List → • List Pair, (]

⑥ [List → • Pair, (]

FIRST(Sa) =

Step 4

$|CC_0| = 9$

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 := ∅
6     for item ∈ s:
7       if item = [α → β • X δ, a] then
8         moved := moved ∪ { [α → β X • δ, a] }
9       end
10    return closure(moved)
```

e.g. I_0

X

I_0 → goto(I_0)

state: - currently we are trying to reduce α (LHS)

- have already matched β
- next: X

CC Construction: goto

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

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

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

```
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 := ∅
6    for item ∈ s:
7      if item = [α → β • xδ, a] then
8        moved := moved ∪ { [α → βx • δ, a] }
9      end
10   return closure(moved)
```