

Examples on LR(0) Parser (SLR Parser)

VII Semester Language Processors Unit 2-Lecture notes

M.B.Chandak

Course coordinator

Example-1: Generate LR(0) Parser: Canonical collections of LR(0) items

$S \rightarrow AA$

$A \rightarrow aA \mid b$

Solution:

Step: 1 – Grammar Augmentation

$S' \rightarrow .S$... Rule 0

$S \rightarrow .AA$... Rule 1

$A \rightarrow .aA$... Rule 2

$A \rightarrow .b$... Rule 3

Step: 2 – Closure operation = I0

$S' \rightarrow .S$

$S \rightarrow .AA$

$A \rightarrow .aA$

$A \rightarrow .b$

Goto (I0, S) = I1

$S' \rightarrow S.$ //**

Goto(I0, A) = I2

$S \rightarrow A.A$

$A \rightarrow .aA$

$A \rightarrow .b$

Goto(I0, a) = I3

$A \rightarrow a.A$

$A \rightarrow .aA$

$A \rightarrow .b$

Goto(I0, b) = I4

$A \rightarrow b.$ //**

Goto(I2, A) = I5

$S \rightarrow AA.$

Goto (I2, a) = I3

Goto (I2, b) = I4

Goto (I3, A) = I6

$A \rightarrow aA.$

Goto (I3, a) = I3

Goto (I3, b) = I4

Rules for construction of parsing table from Canonical collections of LR(0) items

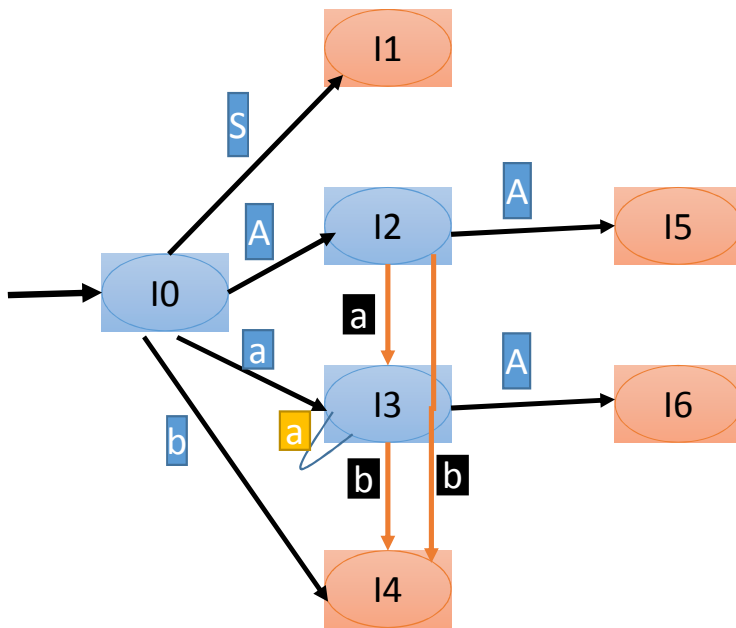
- **Action part: For Terminal Symbols**

- If $A \rightarrow \alpha.a\beta$ is state I_x in Items and $\text{goto}(I_x, a) = I_y$ then set action $[I_x, a] = S_y$ (represented as shift to state I_y)
- If $A \rightarrow \alpha.$ is in I_x , then set **action** $[I_x, f]$ to reduce $A \rightarrow \alpha$ for all symbols "f" where "f" is in $\text{Follow}(A)$ (Use rule number)
- If $S' \rightarrow S.$ is in I_x then set $\text{action}[I_x, \$] = \text{accept}$.

- **Go To Part: For Non Terminal Symbols**

- If $\text{goto}(I_x, A) = I_y$, then $\text{goto}(I_x, A)$ in table = Y
- It is numeric value of state Y .
- All other entries are considered as error.
- Initial state is $S' \rightarrow .S$

DFA and Parsing Table



	a	b	\$	S	A
	ACTION			GOTO	
I0	S3	S4		1	2
I1			Accept		
I2	S3	S4			5
I3	S3	S4			6
I4	r3	r3	r3		
I5			r1		
I6	r2	r2	r2		

Follow(S) = \$
 Follow(A) = {a,b,\$}

Example:2

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

Solution:

Step 1: Grammar Augmentation

$S' \rightarrow .S$... rule 0

$S \rightarrow .Aa$... rule 1

$S \rightarrow .bAc$... rule 2

$S \rightarrow .Bc$... rule 3

$S \rightarrow .bBa$... rule 4

$A \rightarrow .d$... rule 5

$B \rightarrow .d$... rule 6

Step 2: Closure operation

$S' \rightarrow .S$

$S \rightarrow .Aa$

$S \rightarrow .bAc$

$S \rightarrow .Bc$

$S \rightarrow .bBa$

$A \rightarrow .d$

$B \rightarrow .d$

Goto (i0, S) = i1

$S' \rightarrow S.$

Goto (i0,A) = i2

$S \rightarrow A.a$

Goto(i0,b) = i3

$S \rightarrow b.Ac$

$S \rightarrow b.Ba$

$A \rightarrow .d$

$B \rightarrow .d$

Goto(i0,B)=i4

$S \rightarrow B.c$

Goto(i0,d)=i5

$A \rightarrow d.$

$B \rightarrow d.$

Goto(i2,a)=i6

$S \rightarrow Aa.$

Goto(i3,A)=i7

$S \rightarrow bA.c$

Goto(i3,B)=i8

$S \rightarrow bB.a$

Goto(i3,d)=i5 //loop

Goto(i4,c)=i9

$S \rightarrow Bc.$

Goto(i7,c)=i10

$S \rightarrow bAc.$

Goto(i8,a)=i11

$S \rightarrow bBa.$

DFA and Parsing Table

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$

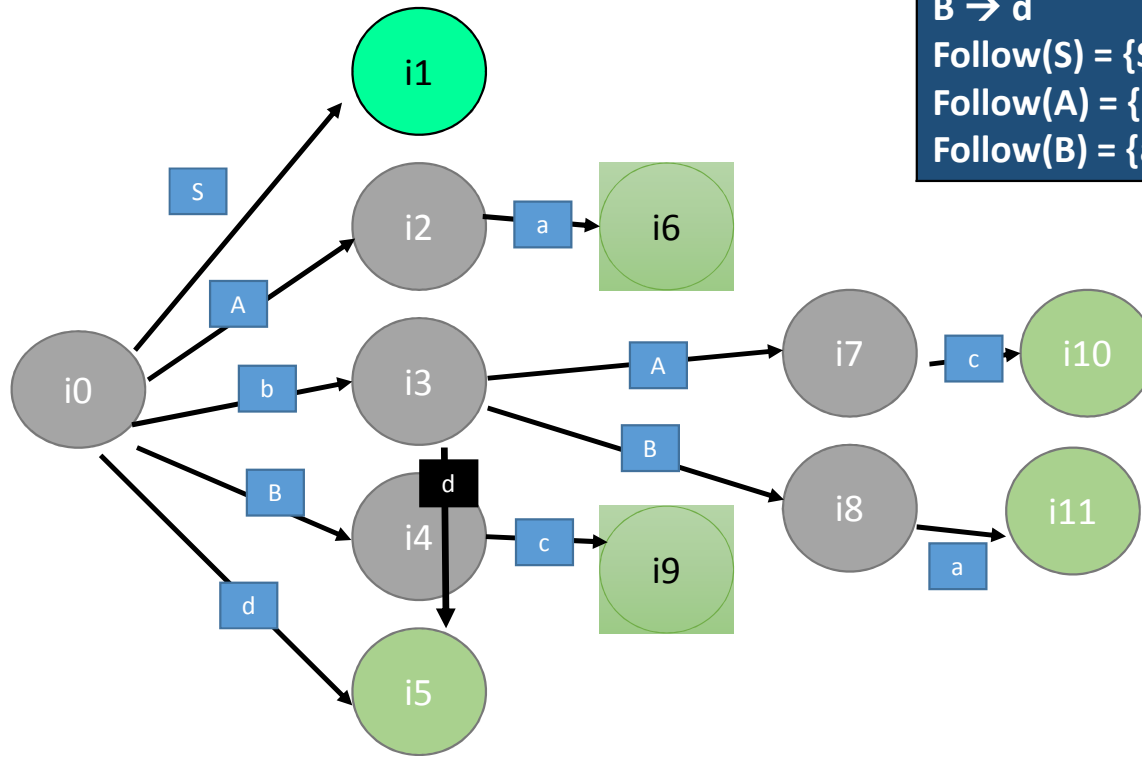
$A \rightarrow d$

$B \rightarrow d$

$\text{Follow}(S) = \{\$, \}$

$\text{Follow}(A) = \{a, c\}$

$\text{Follow}(B) = \{a, c\}$



Parsing table: Not LR(0): Reduce-Reduce conflict

	a	b	c	d	\$	S	A	B
I0		S3		S5		1	2	4
I1					Accept			
I2	S6							
I3				S5			7	8
I4			S9					
I5	R5,R6		R5,R6					
I6					R1			
I7			S10					
I8	S11							
I9					R3			
I10					R2			
I11					R4			

Example:3

$S \rightarrow L = R \mid R$

$L \rightarrow * R \mid id$

$R \rightarrow L$

Solution:

Step 1: Grammar Augmentation

$S' \rightarrow .S$... rule 0

$S \rightarrow .L = R$... rule 1

$S \rightarrow .R$... rule 2

$L \rightarrow .* R$... rule 3

$L \rightarrow .id$... rule 4

$R \rightarrow .L$... rule 5

Step 2: Closure operation ($S' \rightarrow .S$)

$S' \rightarrow .S$

$S \rightarrow .L = R$

$S \rightarrow .R$

$L \rightarrow .*R$

$L \rightarrow .id$

$R \rightarrow .L$

Goto (i0, S) = i1

$S' \rightarrow S.$

Goto (i0, L) = i2

$S \rightarrow L. = R$

$R \rightarrow L. // **$

Goto (i0, R) = i3

$S \rightarrow R. // **$

Goto (i0, *) = i4

$L \rightarrow *.R$

$R \rightarrow .L$

$L \rightarrow .*R$

$L \rightarrow .id$

Goto (i0, id) = i5

$L \rightarrow id.$

Goto (i2, =) = i6

$S \rightarrow L = .R$

$R \rightarrow .L$

$L \rightarrow .*R$

$L \rightarrow .id$

Goto (i4, R) = i7

$L \rightarrow *R.$

Goto (i4, L) = i8

$R \rightarrow L. // ****$

Goto (i4, *) = i4

Goto (i4, id) = i5

Goto (i6, R) = i9

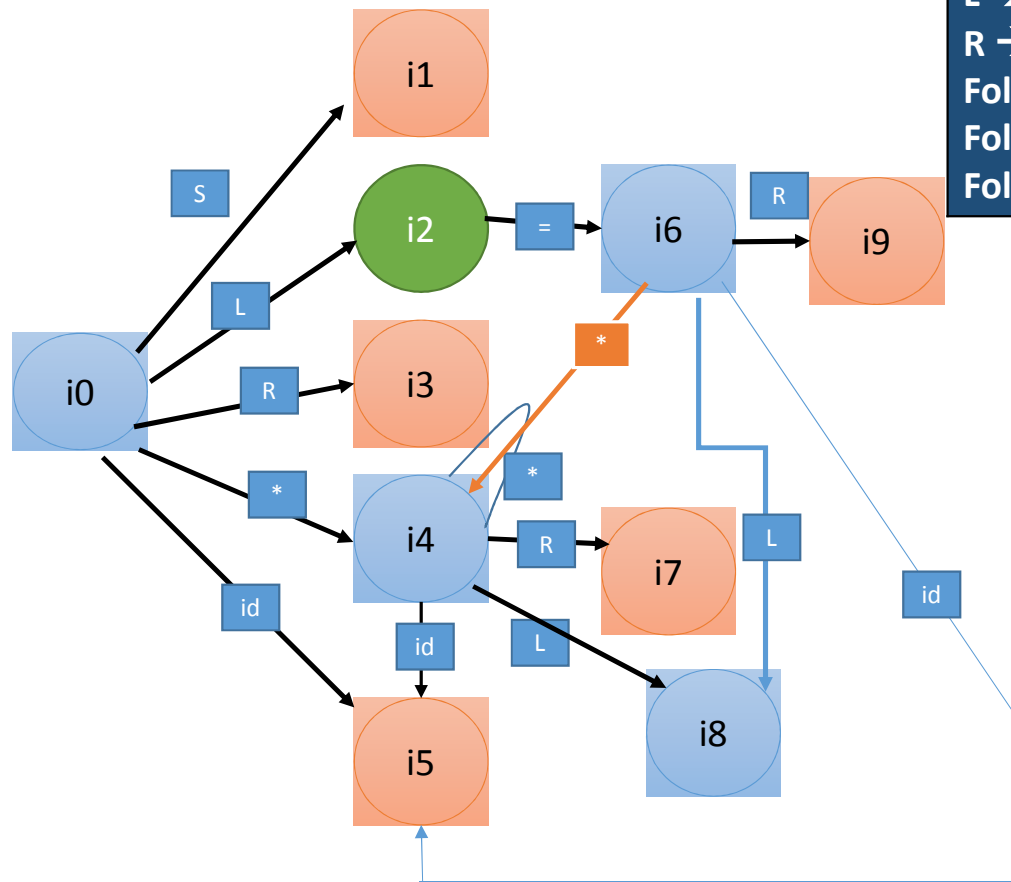
$S \rightarrow L = R.$

Goto (i6, L) = i8

Goto (i6, *) = i4

Goto (i6, id) = i5

DFA and Parsing Table



$S \rightarrow L = R \mid R$
 $L \rightarrow * R \mid id$
 $R \rightarrow L$
 $\text{Follow}(S) = \{\$, \}$
 $\text{Follow}(L) = \{=, \$\}$
 $\text{Follow}(R) = \{=, \$\}$

Parsing table: Not LR(0): Shift-Reduce conflict

	=	*	id	\$	S	L	R
I0		S4	S5		1	2	3
I1				Accept			
I2	S6, R5			R5			
I3				R2			
I4		S4	S5			8	7
I5	R4			R4			
I6		S4	S5			8	9
I7	R3			R3			
I8	R5			R5			
I9				R1			

Example 4: LR(0) or SLR Parsing

$E \rightarrow E+T \mid T$

$T \rightarrow T*F \mid F$

$F \rightarrow (E) \mid id$

Step 1: LR(0) Parser permits Left recursion and left factoring as it operates in Bottom up order.

Grammar Augmentation:

$E' \rightarrow .E$

$E \rightarrow .E+T$

$E \rightarrow .T$

$T \rightarrow .T*F$

$T \rightarrow .F$

$F \rightarrow .(E)$

$F \rightarrow .id$

Step 2:

Closure of $E' \rightarrow .E = i0$

$E' \rightarrow .E$

$E \rightarrow .E+T$

$E \rightarrow .T$

$T \rightarrow .T*F$

$T \rightarrow .F$

$F \rightarrow .(E)$

$F \rightarrow .id$

Step 3: $GOTO(i0, E) = i1$

$E' \rightarrow E.$

$E \rightarrow E.+T$ //as dot crosses E

$GOTO(i0, T) = i2$

$E \rightarrow T.$

$T \rightarrow T.*F$

$GOTO(i0, F) = i3$

$T \rightarrow F.$ //Rule completed

$GOTO\{i0, (\} = i4$

$F \rightarrow (.E)$ // dot is prefixed to E

$E \rightarrow .E+T$

$E \rightarrow .T$ //dot is prefixed to T

$T \rightarrow .T*F$

$T \rightarrow .F$

$F \rightarrow .(E)$

$F \rightarrow .id$

$GOTO(i0, id) = i5$

$F \rightarrow id.$ //rule completed

//Processing of step i0 is completed.

Example 4:

Continue with i1

GOTO(i1, +) = i6

$E \rightarrow E+.T$

$T \rightarrow .T*F$

$T \rightarrow .F$

$F \rightarrow .(E)$

$F \rightarrow .id$

GOTO(i2, *) = i7

$T \rightarrow T*.F$

$F \rightarrow .(E)$

$F \rightarrow .id$

Goto(i4, E) = i8

$F \rightarrow (E.)$

$E \rightarrow E.+T$

Goto(i4, T)=i2

Goto(i4,F)=i3

Goto(i4, ()=i4

Goto(i4,id) =i5

Goto(i6,T) = i9

$E \rightarrow E+T.$

$T \rightarrow T.*F$

Goto(i6,F)=i3

Goto(i6,()=i4

Goto(i6, id) = i5

Goto(i7,F)=i10

$T \rightarrow T*F.$

Goto(i7,()=i4

Goto(i7,id) =i5

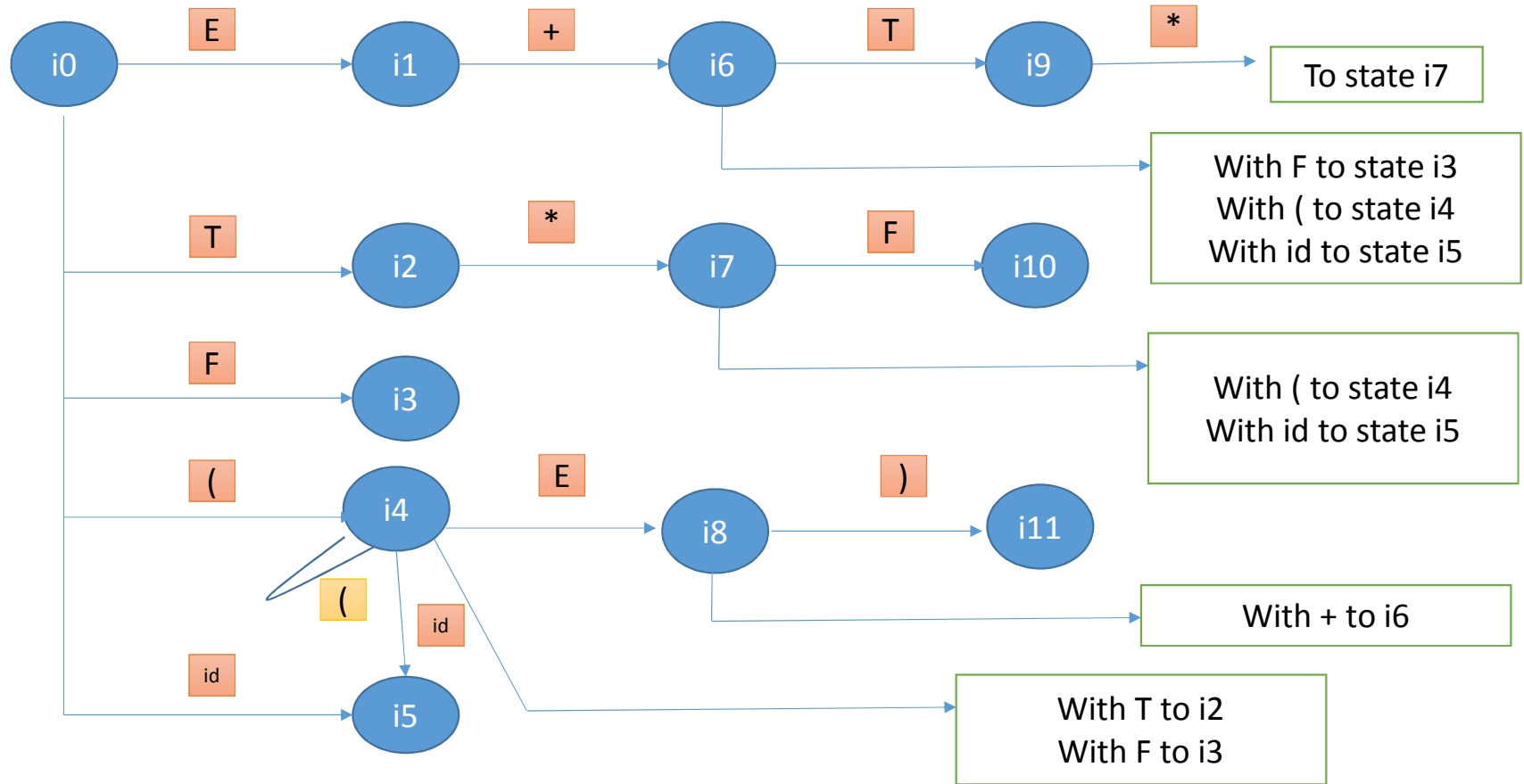
Goto(i8,))=i11

$F \rightarrow (E).$

Goto(i8,+)=i6

Goto(i9,*)=i7

DFA for the states



Parsing Table: No multiple entries: LR(0) Grammar

	id	+	*	()	\$	E	T	F
I0	s5						1	2	3
I1		s6				Accept			
I2		r2	s7		r2	r2			
I3		r4	r4		r4	r4			
I4	s5			s4			8	2	3
I5		r6	r6		r6	r6			
I6	s5			S4				9	3
I7	s5			S4					10
I8		s6			S11				
I9		r1	s7		r1	r1			
I10		r3	r3		r3	r3			
I11		r5	r5		r5	r5			

Example:5 → Production with only € transition

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \epsilon$

$B \rightarrow \epsilon$

Solution:

Step 1: Grammar Augmentation

$S' \rightarrow .S$.. Rule 0

$S \rightarrow .AaAb$.. Rule 1

$S \rightarrow .BbBa$... Rule 2

$A \rightarrow .$ (rule 3)

$B \rightarrow .$ (rule 4)

Step 2: Closure of $S' \rightarrow .S = i0$

$\{S' \rightarrow .S$

$S \rightarrow .AaAb$

$S \rightarrow .BbBa$

$A \rightarrow .$

$B \rightarrow .$ }

Step 3: Goto operations

Goto(i0,S)=i1

$S' \rightarrow S.$

Goto(i0, A)=i2

$S \rightarrow A.aAb$

Goto(i0,B) = i3

$S \rightarrow B.bBa$

Goto(i2, a) = i4

$S \rightarrow Aa.Ab$

$A \rightarrow .$

Goto(i3,b) = i5

$S \rightarrow Bb.Ba$

$B \rightarrow .$

Goto(i4,A) = i6

$S \rightarrow AaA.b$

Goto(i5,B) = i7

$S \rightarrow BbB.a$

Goto(i6,b) = i8

$S \rightarrow AaAb.$

Goto(i7, b) = i9

$S \rightarrow BbBa.$

FOLLOW(S) = {\$}

FOLLOW(A) = {a,b}

FOLLOW(B) = {a,b}

Parsing Table: Reduce – Reduce Conflict

	a	b	\$	S	A	B
I0	R3, R4	R3, R4		1		
I1			Accept			
I2	S4					
I3		S5				
I4	R3	R3			6	
I5	R4	R4				7
I6		S8				
I7	S9					
I8			R1			
I9			R2			

Example: 6: Total 9 states

$S \rightarrow xAy \mid xBy \mid xAz$

$A \rightarrow aS \mid q$

$B \rightarrow q$

Example 7:

$S \rightarrow aSbS \mid bSaS \mid \epsilon$

Example 8

$S \rightarrow aAB \mid bB$

$A \rightarrow Aa \mid$

$B \rightarrow Bb \mid$

Example-1:String Parsing using LR(0) parsing table

$S \rightarrow AA$

$A \rightarrow aA \mid b$

Solution:

Step: 1 – Grammar Augmentation

$S' \rightarrow .S$... Rule 0

$S \rightarrow .AA$... Rule 1

$A \rightarrow .aA$... Rule 2

$A \rightarrow .b$... Rule 3

Step: 2 – Closure operation = I0

$S' \rightarrow .S$

$S \rightarrow .AA$

$A \rightarrow .aA$

$A \rightarrow .b$

Goto (I0, S) = I1

$S' \rightarrow S.$ //**

Goto(I0, A) = I2

$S \rightarrow A.A$

$A \rightarrow .aA$

$A \rightarrow .b$

Goto(I0, a) = I3

$A \rightarrow a.A$

$A \rightarrow .aA$

$A \rightarrow .b$

Goto(I0,b) = I4

$A \rightarrow b.$ //**

Goto(I2, A) = I5

$S \rightarrow AA.$

Goto (I2, a) = I3

Goto (I2, b) = I4

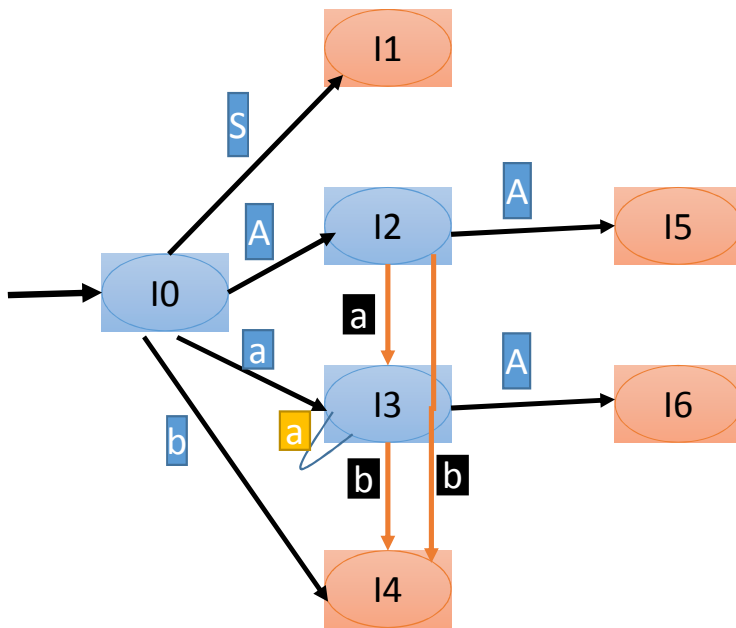
Goto (I3, A) = I6

$A \rightarrow aA.$

Goto (I3, a) = I3

Goto (I3, b) = I4

DFA and Parsing Table



	a	b	\$	S	A
	ACTION			GOTO	
I0	S3	S4		1	2
I1			Accept		
I2	S3	S4			5
I3	S3	S4			6
I4	r3	r3	r3		
I5			r1		
I6	r2	r2	r2		

Follow(S) = \$
 Follow(A) = {a,b,\$}

String: aabb

	a	b	\$	S	A
	ACTION			GOTO	
I0	S3	S4		1	2
I1			Accept		
I2	S3	S4			5
I3	S3	S4			6
I4	r3	r3	r3		
I5			r1		
I6	r2	r2	r2		

Stack	Input	Action
0	aabb\$	I0 → a, S3
0a3	aabb\$	I3 → a, S3
0a3a3	aabb\$	I3 → b, S4
0a3a3b4	aabb\$	I4 → b, r3
0a3a3A6	aabb\$	I6 → b, r2
0a3A6	aabb\$	I6 → b, r2
0A2	aabb\$	I2 → b, s4
0A2b4	aabb\$	I4 → \$, r3
0A2A5	aabb\$	I5 → \$, r1
0s1	aabb\$	accept

Shift "a" and goto state 3

Reduction means: reduce the previous symbol set to RHS and not reducing the actual symbol at the pointer.

Pop number symbols = Length of RHS * 2

Below "A" is 3, so 3 → A = i6