SLR Parsing
The Table M (Action-Goto) for SLR

Let $G=<S,V,\Pi,s>$ and I be the set of LR(0) Collection of G.

Table M (also, called, Action-Goto) consists of two Tables:

- **Action**: is **$|I|$-rows** by **$|V \cup \{\$\}|$-columns**
- **Goto**: is **$|I|$-rows** by **$|S|$-columns**

For each state, of **I**:

- **Action**: The operation to apply (shift or reduce)
- **Goto**: The state to go after reduction

The definition of M requires the computation of:

- LR(0) Collection
- Follow(s), for each $s \in S$

SLR Parsing
The Table Action for SLR(1)

Component, Action, of table M for SLR(1)

ACTION(i,a)=s/j

if goto(Ii,a)=Ij and $a \in \Sigma$

ACTION(i,a)= r/p

if $A ::= \alpha . \in Ii$ and $p \equiv A ::= \alpha$ and $a \in follow(A)$

ACTION(i,$)= <accept,->

if $S' ::= S . \in Ii$

Where: s/j = shortening for Shift(Ij);
r/p = shortening for Reduce(p)

SLR Parsing
The Table Goto for SLR(1)

Let G=<S,V,Π,s>

GOTO(i,A)=j
if goto(Ii,A)=Ij and A ∈S

Example
Table Action-Goto SLR(1)

	a	b	c	d	e	$
0	S/2					
1						ACC
2		S/4				
3		S/6		S/7		
4		R/2		R/2		
5					S/8	
6			S/9			
7					R/3	
8						R/0
9		R/1		R/1		

	S	A	B
0	1		
1			
2		3	
3			5
4			
5			
6			
7			
8			
9			

LR(0) Collection

I0:Closure({S'::=.S}) ={S'::=.S, S::= .aABe}
I1:Goto(I0,S) ={S'::=S.}
I2:Goto(I0,a) ={S::= a.Abe, A::=.Abc, A::=.b}
I3:Goto(I2,A) ={S::= aA.Be, A::=A.bc, B::= .d}
I4:Goto(I2,b) ={A::=b.}
I5:Goto(I3,B) ={S::= aAB.e}
I6:Goto(I3,b) ={A::=Ab.c}
I7:Goto(I3,d) ={B::= d.}
I8:Goto(I5,e) ={S::= aABe.}
I9:Goto(I6,c) ={A::=Abc.}

Source Grammar

S'::=S
0 S::= aABe
1|2 A::= Abc | b
3 B::= d

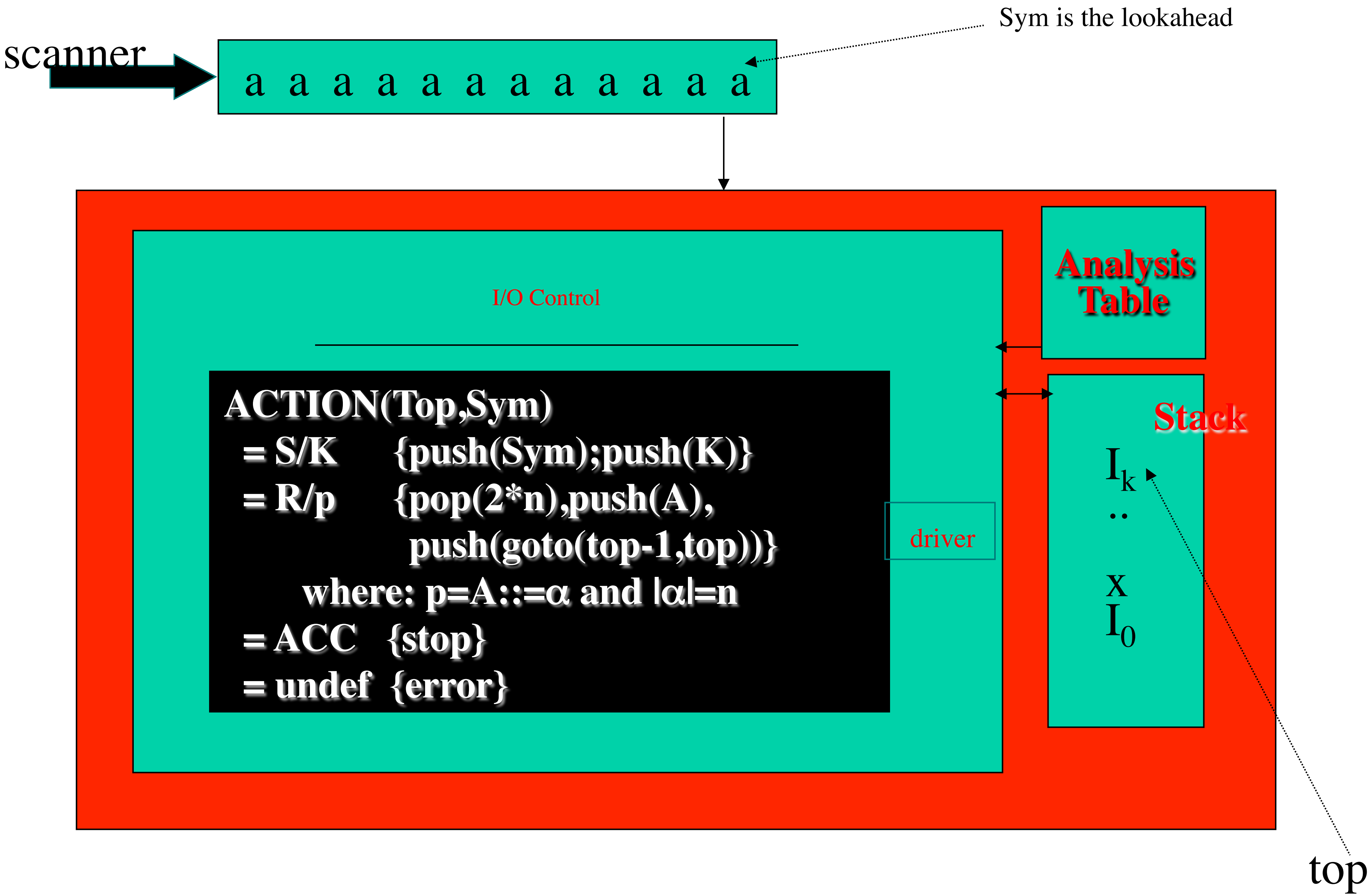
Augmented G'

S'::=S
0 S::= aABe
1|2 A::= Abc | b
3 B::= d

Table of Follow

follow(S)={$}
follow(A)={b,d}
follow(B)={e}

DRIVER: shift-reduce



SLR(1) Parsing
Properties of Applicability

Prop1: (No shift-reduce)
$(\forall\ I_k \in Coll(0)), (\forall\ A ::= \alpha.a\gamma,\ B ::= \beta. \in I_k)$
$a \notin follow(B)$

Prop2: (No reduce-reduce)
$(\forall\ I_k \in Coll(0)), (\forall\ A ::= \alpha.,\ B ::= \beta. \in I_k)$
$follow(A) \cap follow(B) = \{\}$

Theorem. Let G be a grammar.
- $G \in SLR(1)$ if and only if both Prop1 and Prop2 hold.
- G has shift-reduce, 1 symbol of lookahead, SLR parser if and only if both Prop1 and Prop2 hold.

SLR
Concluding Remarks

1. In what sense is "bottom-up" non predictive **?**

Bottom-up decides the derivation on the basis of the maximum VP to the handle and this VP entirely, includes the string to derive.

2. Let G be a SLR(1) grammar.
- Can its ACTION contain more than one action per entry **?**
- Can its GOTO contain more than one state for a same entry **?**

3. What means that G∈SLR(1) has Table:
ACTION such that: for each I, for each pair a≠b,
ACTION(I,a)=ACTION(I,b)
when ACTION(I,a) ≠⊥≠ ACTION(I,b) **?**

To answer 3, let us consider the **SLR(0)** grammar below:

E::= E+T | T
T::= num

SLR
Concluding Remarks - 2

4. Is G $\in$SLR(1) decidable for all Context Free grammars ?

5. Is **L** $\in$SLR(1) decidable for all Context free languages ?

Recall that: L $\in$SLR(1) $\equiv$ $\exists$ G $\in$LL(1): L(G) = **L**

6. Is the class of SLR(1) grammars including that of LL(1) grammars ?

To answer 6, let us consider the grammar below:

S::= Bc | b | A
A::= aBb
B::=ε

SLR
Concluding Remarks - 3

7. Is the class of SLR(1) languages, including that of LL(1) languages ?

To answer 7, let us consider the role of non-terminal B:

S::= Bc | b | A
A::= aBb
B::=ε

8. Consider the condition C:

$C \equiv A::=\alpha . \in I_k \in Coll(0)$ *iff* $A::=\alpha$ is the handhe for all $a \in follow(A)$.

C states a necessary and sufficient for reduction in SLR(1) Grammars.

- Is C again necessary and sufficient for reduction in botton-up (non SLR grammars) ?
- Is C, only a necessary, condition for reduction in botton-up (non SLR grammars) ?
- Is C, only a sufficient, condition for reduction in botton-up (non SLR grammars) ?

To answer to 8, let us consider the grammar above and the existence of a rightmost derivation for string b. Such a derivation (in reverse order) is b<=S, but SLR(1) fails in doing it. Why?

Bottom-Up Analysis:

First(β)∈Follow(A)

Let: $S \Rightarrow_r^* \gamma\alpha\beta$ for

γα ∈viable prefix - *is maximum vp*
First(β)∈follow(A)
A::=α. is Valid Item for γα

→

Can you conclude that
A::= α is the handle of γαβ ?

Use this grammar for a proof

S::= Bc | b | A
A::= aBb
B::=ε

First(β)∈follow(A) is not a sufficient condition

Proposition. (Inclusion is not a sufficient condition)
∀G, ∀γα ∈VP_G with Valid Item A->α.
First(β) ∈follow(A) **does not imply that** A::=α is the handle of γα*β*

Use this grammar for a proof: (proof)

Let γαβ≡b. Then, S => b
γα≡λ∈VP is maximum prefix; First(β)∈Follow(B); B::=ε is valid for γα BUT B::=ε is not the handle of b

A New Class of
More Powerful Items

Let $X::=\alpha.A\beta \in I_k$ and $A::=\gamma$ be a grammar (normal) production

Then

$A::=.\gamma \in I_k$ **only for the symbols in the set U**=First(β) (for $B \neq> \lambda$)

First(β *look*($X::=\alpha.A\beta$))

Pairing each LR(0) item with the set U of all the symbols that can follow the handle string associated to the item

$A::=.\gamma$,U $\in I_k$

Item LR(1)

U is called *look(A::=.γ)*

Canonical collection Coll(1)

A New Class of Items that Remember Prefix-Follow Combinations

Let G=<S,V,Π,s>

S'::=.S/$ ∈I_0 by definition

Clos(I) $=_{min}$ I ∪ Clos{B::= .γ/U | A::=α.Bβ/V∈Clos(I),B::=γ∈Π,
U ={first(βx)| x∈V}}

Let B a nonterminal
let u ∈U, then:
if u ∈V, A::=α.Bβ/V *propagates* on B::= .γ/U
if u ∉V, A::=α.Bβ/V *spontaneously generates* on B::= .γ/U

Goto(I,x)=Closure{A::= αx.β/U | A::=α.xβ/U∈I}

LR(1) Grammars

Prop1: (No shift-reduce)

$(\forall\ I_k \in Coll(1)), (\forall\ A ::= \alpha.a\gamma/V\ ,\ B ::= \beta./U\ \in I_k)$

$a \notin U$

Prop2: (No reduce-reduce)

$(\forall\ I_k \in Coll(1)), (\forall\ A ::= \alpha./V\ ,\ B ::= \beta./U \in I_k)$

$V \cap U = \{\}$

Theorem. Let G be a grammar.

- **$G \in LR(1)$ if and only if both Prop1 and Prop2 hold.**
- **G has shift-reduce, 1 symbol of lookahead, LR parser if and only if both Prop1 and Prop2 hold.**

LR Parsing
The Table Action for LR(1)

ACTION(i,a)=<shift,j>

if goto(Ii,a)=Ij and a∈Σ

ACTION(i,a)= <reduce,p>

if A::=α./U ∈Ii and

p≡ A::= α and a∈U

ACTION(i,$)= <accept,->

if S'::=S./$ ∈ Ii

GOTO(i,A)=j

if goto(Ii,A)=Ij and A ∈N

N=Nonterminal Set

Apply LR(1) to G∉SLR(1)

I0:Closure({S'::=.S}) ={S'::= .S,$
S::= .Au,$, S::= .av,$, S::= .Bv,$,
A::= .a,u, B::=.xA,v}
I1:Goto(I0,S) ={S'::= S.,$}
I2:Goto(I0,A)={S::= A.u,$}
I3:Goto(I0,a) ={S::= a.v,$
A::= a.,u} No conflict
I4:Goto(I0,B) ={S::= B.v,$}
I5:Goto(I0,x) ={S::= x.A,v
A::= .a,v}
I6:Goto(I2,u) ={S::= Au.,$}
I7:Goto(I3,v) ={S::= av.,$}
I8:Goto(I4,v) ={S::= Bv.,$}
I9:Goto(I5,A) ={S::= xA.,v}
I10:Goto(I5,a) ={A::= a.,v}

S::= A u | a v | B v
A::= a
B::= x A

Space Complexity

Table Size **G=<S,V,Π,s>**

LL(1): O(N*T) (with N =|S| , T=|V|)

SLR: O(N*P*T) (with P=production right hand size)

LR(1): O(N*P*T*T)

LALR(1): the same of SLR