Space Complexity

	Table Size	$G=\langle S,V,\Pi,s\rangle$
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	

Example

S::=Sb | caSd | c

Example: S::=Sb | caSd | c

I0:{S'::=.S S::=.Sb S::=.caSd S::=.c}
I1=Goto(I0,S)={S'::=S. S::=S.b}
I2=Goto(I0,c)={S::=c.aSd S::=c.}
I3=Goto(I1,b)={S::=Sb.}
I4=Goto(I2,a)={S::=ca.Sd S::=.Sb S::=.caSd S::=.c} kernel
I5=Goto(I4,S)={S::=caS.d S::=S.b}
Goto(I4,c)=I2
I6=Goto(I5,d)={S::=caSd.}

Example - cnt.

S::=Sb | caSd | c

I0:{S'::=.S/$ S::=.Sb/$/b S::=.caSd/$/b S::=.c/$/b}

I1=Goto(I0,S)={S'::=S./$, S::=S.b/$/b}

I2=Goto(I0,c)={S::=c.aSd/$/b, S::=c./$/b}

I3=Goto(I1,b)={S::=Sb./$/b}

I4=Goto(I2,a)={S::=ca.Sd/$/b, S::=.Sb/d/b. S::=.caSd/d/b, S::=.c/d/b}

I5=Goto(I4,S)={S::=caS.d/$/b, S::=S.b/d/b}

I6=Goto(I4,c)={S::=c.aSd/d/b, S::=c./d/b}

I7=Goto(I5,d)={S::=caSd./$/b}=Goto(I10,b)

I8=Goto(I5,b)={S::=Sb./d/b}=

I9=Goto(I6,a)={S::=ca.Sd/d/b, S::=.Sb/d/b, S::=.caSd/d/b, S::=.c/d/b}

I10=Goto(I9,S)={S::=caS.d/d/b, S::=S.b/d/b}

Goto(I9,c)=I6

I11=Goto(I10,d)={S::=caSd./d/b}

From *LR(1)* to more compact ***LALR(1)***

Each SLR(1) state

$\{t_1\ t_2\ \dots\ t_k\}$

may be duplicated into n LR(1) states

$\{t_1/U_{11}\ \ t_2/U_{21}\ \dots\ \ t_k/U_{k1}\}$

......

$\{t_1/U_{1n}\ \ t_2/U_{2n} \dots\ t_k/U_{kn}\}$

LALR(1)

Merge all together, states that are sharing the same kernel

$\{t_1/U_{11}\cup\dots\cup U_{1n}\quad t_2/U_{21}\cup\dots\cup U_{2n}\ \dots\ t_k/U_{k1}\cup\dots\cup U_{kn}\}$

Canonical Collection of LALR states: $Coll_{LALR}$

Let $\Leftrightarrow$ be the equivalence relation on Coll(1) defined below:

$$I_j \Leftrightarrow I_k \;\; \textit{iff} \;\; I_j{\downarrow}0 = I_k{\downarrow}0$$

where $I{\downarrow}0 = \{A::=\alpha.\beta \mid A::=\alpha.\beta/U \in I\}$

$Coll_{LALR}$ = Coll(1)/$\Leftrightarrow$ = {[I] | I $\in$Coll(1)}

(read: partition Coll(1) modulo $\Leftrightarrow$)

where: [I] is a representant, i.e.: [I]=[J] *iff* J$\Leftrightarrow$I

$(\forall\, I_j, I_k \in Coll(1)), (\forall\, a \in N \cup T),$

$I_j = Goto(I_k, a) \;\; \textit{iff} \;\; [I_j] = Goto([I_k], a)$

Function Goto of LALR parser

SLR(1), LR(1), LALR(1): Conflict Comparation

shift/reduce

$A ::= \alpha . / U$, $B ::= \beta . a\delta / V \in Ij \in Coll_{LALR(1)}$ and $a \in U$

$U = U_1 \cup ... \cup U_n$

The conflict persists in LR(1) for some Ij_i ($1 \leq i \leq n$)

reduce/reduce

The conflict could be absent in LR(1)

LALR(1) Grammars

Prop1: (No shift-reduce)
$(\forall\ I_k \in Coll_{LALR}), (\forall\ A ::= \alpha.a\gamma/V\ ,\ B ::= \beta./U\ \in I_k)$
$a \notin U$

Prop2: (No reduce-reduce)
$(\forall\ I_k \in Coll_{LALR}), (\forall\ A ::= \alpha./V\ ,\ B ::= \beta./U \in I_k)$
$V \cap U = \{\}$

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

Bottom-up:
Concluding Remarks

1. Why SLR(1) ⊂ LALR(1) ⊂ LR(1) is strict inclusion ?

2. Is for each language **L** decidable the existence of some **G** such that:
L(**G**) = **L** and **G** ∈LALR(1) [resp. **G** ∈LR(1)] ?

3. Why, looking for a 1-lookahead, bottom-up, parser for some G, people try proving G ∈SLR(1), for first ?

Bottom-up:
Concluding Remarks

4. If G$\notin$LALR(1) for some G, due to the violation of Prop.2, nevertheless G $\in$LR can hold. Why?

5. Why SLR(1) $\subset$ LALR(1) ?
 - Prove that: if G $\in$SLR(1) then G $\in$LALR(1)
 - Prove that inclusion is really a strict one ?

6. Let G be an ambiguous grammar:
 Which of the two properties is violated, for sure: Prop.1 or Prop2 ?
 Why ?

Context Free Languages and Grammars Concluding Remarks - 1

Theorem. Pumping Lemma for $L \in CFL$.

- $\forall$ L, $\exists$ k, such that:
  - if $x \in L$, $|x| > k$, then: $\exists$ p,u,w,v,q such that
    - $x = p.u.w.v.q$
    - $0 < |puwv| \leq k$, and $|uv| > 0$
    - $pu^iwv^iq \in L$ *(for all natural i)*

Exercise. The language $L = \{a^n\ b^m\ c^p \mid n=m=p \geq 0\} \notin CFL$

Context Free Languages and Grammars Concluding Remarks - 2

Proposition. CFL and CFG.

- $\exists L \in CFL$, such that:
  - $\forall G \in CFG$:
    - if $L(G) = L$ then G is an ambiguous grammar

Example. The language $L = \{a^n b^m c^p \mid n=m \text{ or } m=p \geq 0\}$ is intrinsically ambiguous in CFG

$n, m, p \geq 0$

Exercise. Give a CFG for $L = \{a^n b^m c^p \mid n=m \text{ or } m=p \geq 0\}$

$n, m, p \geq 0$

Prove that

$L = \{ a^n b^m c^m \mid m \geq 0 \} \notin CFL$

- Let k be the constant of the Pumping Lemma.
- Let $x = a^k b^k c^k$
  - $x \in L$
  - $x = p u v w q$
    - $|puvw| \leq k \Rightarrow p = a^{h_1}$ $u = a^{h_2}$ $v = a^{h_3}$ $w = a^{h_4}$ $q = a^{h_5} b^k c^k$
      $h_1 + h_2 + h_3 + h_4 + h_5 = k$, $h_5 \geq 0$
    - $|uw| > 0 \Rightarrow h_2 + h_4 > 0$
    - Let $i = 0$, $a^{h_1} a^0 a^{h_3} a^0 a^{h_5} b^k c^k \in L \Rightarrow h_1 + h_3 + h_5 = k \Rightarrow$
      $\Rightarrow h_1 + h_3 + h_5 = h_1 + h_2 + h_3 + h_4 + h_5 \Rightarrow h_2 + h_4 = 0$

Define a grammar for the language $L = \{a^n b^m c^k \mid n = m \vee m = k\}$.

$S ::= P \mid Q \mid \varepsilon$
$P ::= aAbC \mid AcC$
$A ::= aAb \mid \varepsilon$
$C ::= cC \mid \varepsilon$
$Q ::= A_o b B c \mid a A_o B$
$B ::= bBc \mid \varepsilon$
$A_o ::= a A_o \mid \varepsilon$

G is an ambiguous grammar, since it has more than one leftmost derivation for all the strings of the subset $\{a^n b^n c^n \mid n > 0\}$.