CS143: Parsing V

David L. Dill

Stanford University

Parsing

- LR(1) Parsing
- LALR(1) Parsing
- Precedence and Associativity in LR Parse Tables
- Connections with Formal Language Theory

LR(1) Parsing

LR Conflicts

Based on items and lookaheads don't know what to do

$$\boxed{\begin{array}{l} A \to \alpha \bullet a \beta \\ B \to \gamma \bullet a \delta \end{array}}$$

not a conflict

LR(1) parsing

- More powerful than SLR(1)
- Tables are very large
- Basis for LALR(1)

$$SLR(1) < LALR(1) < LR(1)$$

smaller tables

LR Conflicts

Based on items and lookaheads don't know what to do

$$\boxed{\begin{array}{l} A \to \alpha \cdot a\, \beta \\ B \to \gamma \cdot a\, \delta \end{array}} \xrightarrow{a} \boxed{\begin{array}{l} A \to \alpha\, a \cdot \beta \\ B \to \gamma\, a \cdot \delta \end{array}}$$

not a conflict

Don't have to decide what to do.

Just advance both items

LR Conflicts

Based on items and lookaheads don't know what to do

$$\begin{array}{|l|}\hline A \to \bullet B\alpha \\ C \to \beta\bullet \\ \hline\end{array} \xrightarrow{\quad B \quad} \begin{array}{|l|}\hline A \to B\bullet\alpha \\ \\ \hline\end{array}$$

not a conflict

only do this after a reduction.

LR Conflicts

No conflict unless

- at least one reduce item, and
- a shift item or another reduce item
- overlapping lookaheads

LR(0) conflicts - no lookaheads

$$\boxed{\begin{array}{l} A \to \cdot\, a \\ B \to \alpha\, \cdot \end{array}}$$

shift-reduce

$$\boxed{\begin{array}{l} A \to \alpha\, \cdot \\ B \to \beta\, \cdot \end{array}}$$

reduce-reduce

LR(1) non conflicts

$$\boxed{\begin{array}{l} A \to \bullet a, w \\ B \to \alpha \bullet, b \end{array}}$$

shift-reduce

$$\boxed{\begin{array}{l} A \to \alpha \bullet, a \\ B \to \beta \bullet, b \end{array}}$$

reduce-reduce

Disjoint lookaheads say what to do.

XLR(1) ~~non~~ conflicts

$$\boxed{\begin{array}{l} A \to \bullet a,\ \omega \\ B \to \alpha \bullet,\ a \end{array}} \qquad \boxed{\begin{array}{l} A \to \alpha \bullet,\ a \\ B \to \beta \bullet,\ a \end{array}}$$

shift-reduce reduce-reduce

Lookaheads **don't** distinguish actions.

ACTION table

terminals

States	a
6	s3, r5

ACTION table

terminals

States

	a
6	$\boxed{s3, r5}$

XLR(1) conflict = >1 entry in ACTION table.

$S' \to S$

$S \to A\,a$

$S \to B\,b$

$S \to b\,A\,b$

$A \to a$

$B \to a$

$$
\boxed{\begin{array}{l}
S' \to \bullet S \\
S \to \bullet A a \\
S \to \bullet B b \\
S \to \bullet b A b \\
A \to \bullet a \\
B \to \bullet a
\end{array}}
\xrightarrow{a}
\boxed{\begin{array}{l}
A \to a \bullet \\
B \to a \bullet
\end{array}}
$$

← LR(0)

← reduce-reduce conflict

$S' \to S$
$S \to A\,a$
$S \to B\,b$
$S \to b\,A\,b$
$A \to a$
$B \to a$

$S' \to \cdot S$
$S \to \cdot A a$
$S \to \cdot B b$
$S \to \cdot b A b$
$A \to \cdot a$
$B \to \cdot a$

$\xrightarrow{a}$

$A \to a \cdot \quad \{a, b\}$
$B \to a \cdot \quad \{b\}$

SLR(1) conflict

FOLLOW SETS

$A \quad \{a, b\}$
$B \quad \{b\}$

still have reduce/reduce conflict on b.

LR(1) items

Idea: Keep track of lookaheads in the items.

LR(1) item: $[A \to \alpha \bullet \beta, a]$

lookahead for this item

Same production can have left context-dependent lookaheads

LR(1) items

1. First item $[S' \to \cdot S, \$]$
2. LR(1) closure
   When $[A \to \alpha \cdot B \beta, a]$
   add $[B \to \cdot \gamma, b]$
   for all $b \in FNE(\beta a)$
3. Goto, etc. are the same as in LR(0)

$S' \to S$

$S \to Aa$

$S \to Bb$

$S \to bAb$

$A \to a$

$B \to a$

$S' \to \cdot S, \$$

$S' \to S$

$S \to Aa$

$S \to Bb$

$S \to bAb$

$A \to a$

$B \to a$

$S' \to \cdot S, \$$

$S \to \cdot Aa, \$$

$S \to \cdot Bb, \$$

$S \to \cdot bAb, \$$

$S' \to S$

$S \to A\,a$

$S \to B\,b$

$S \to b\,A\,b$

$A \to a$

$B \to a$

$S' \to \bullet S, \$$

$S \to \bullet A\,a, \$$

$S \to \bullet B\,b, \$$

$S \to \bullet b\,A\,b, \$$

$A \to \bullet a, a$

$$
\begin{aligned}
S' &\to S \\
S &\to A\,a \\
S &\to B\,b \\
S &\to b\,A\,b \\
A &\to a \\
B &\to a
\end{aligned}
\qquad
\begin{aligned}
S' &\to \bullet S,\ \$ \\
S &\to \bullet A\,a,\ \$ \\
S &\to \bullet B\,b,\ \$ \\
S &\to \bullet b\,A\,b,\ \$ \\
A &\to \bullet a,\ a \\
B &\to \bullet a,\ b
\end{aligned}
$$

$S' \to S$

$S \to A a$

$S \to B b$

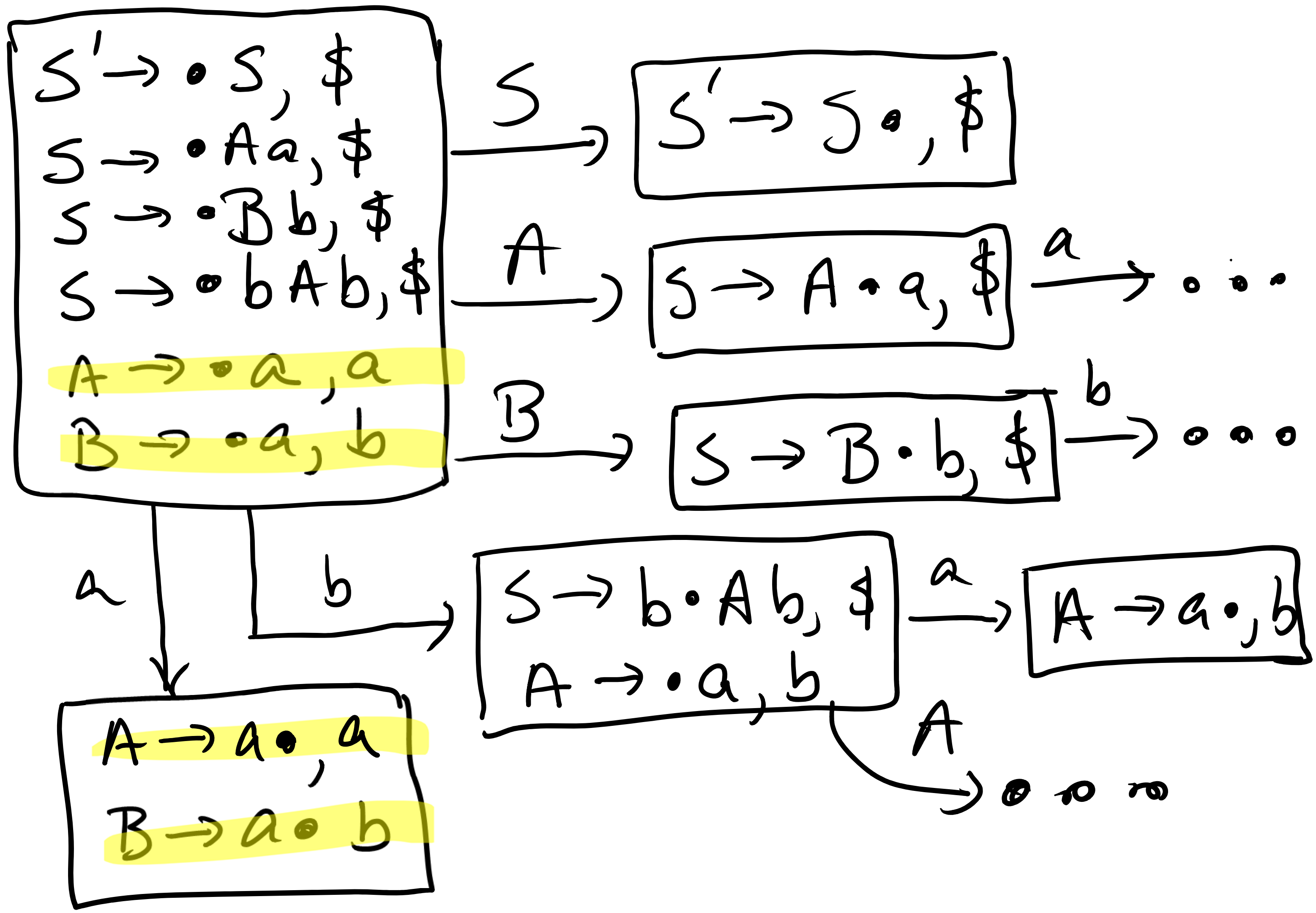
$S \to b A b$

$A \to a$

$B \to a$

$$\boxed{\begin{array}{l} S' \to \bullet S, \$ \\ S \to \bullet A a, \$ \\ S \to \bullet B b, \$ \\ S \to \bullet b A b, \$ \\ A \to \bullet a, a \\ B \to \bullet a, b \end{array}} \xrightarrow{S} \boxed{S' \to S \bullet, \$}$$

$S' \to S$
$S \to A a$
$S \to B b$
$S \to b A b$
$A \to a$
$B \to a$



LALR(1) Parsing

LALR(1) parsing

- LR(1) - lots of states (maybe 10X)
- LALR(1) - use LR(0) states but LR(1) lookaheads.

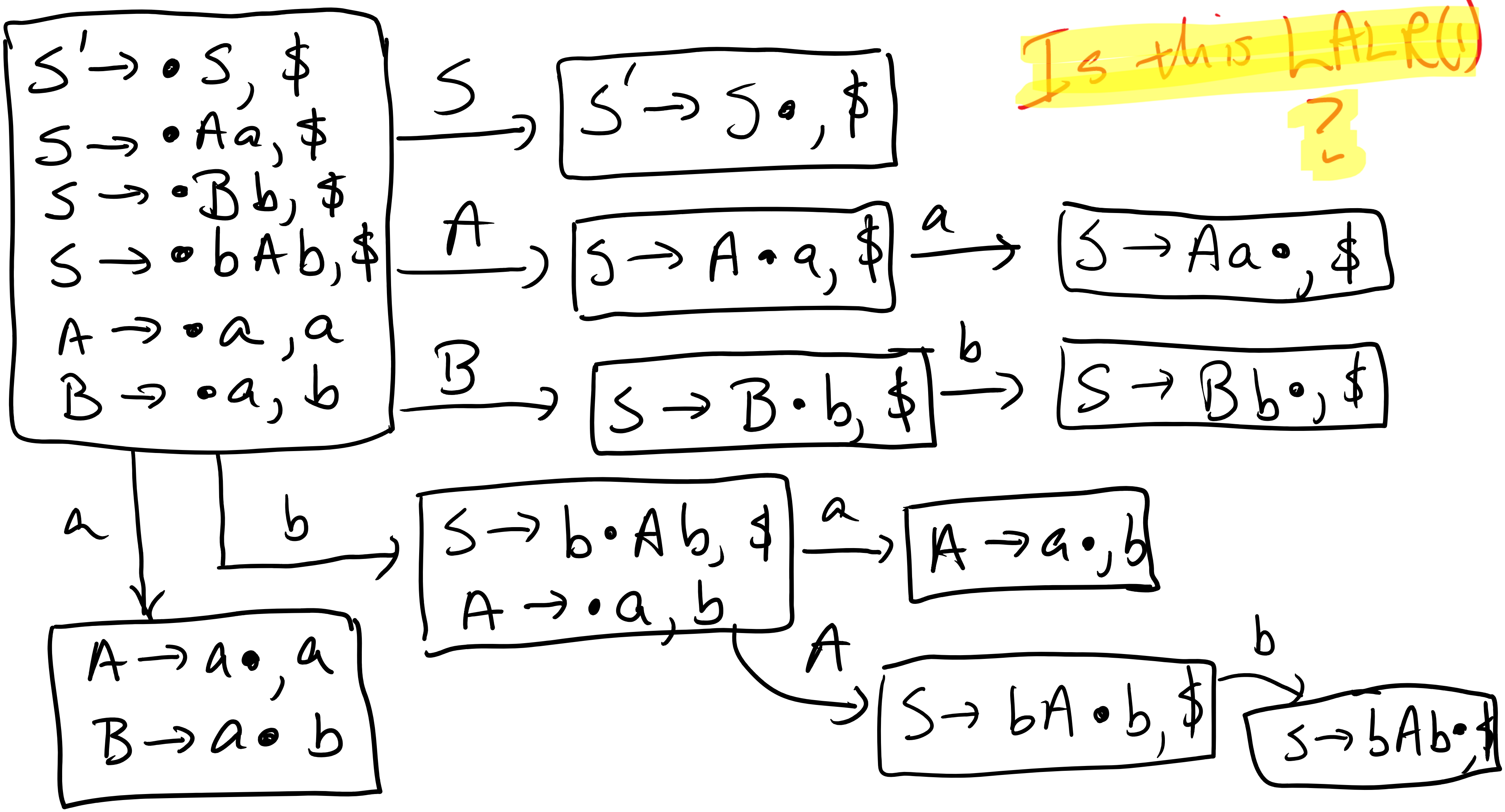
LALR(1)

Construct LR(1) machine

Merge states that have same LR(0) items.

$$\boxed{\begin{array}{l} A \to \alpha \cdot \beta,\ a \\ B \to \gamma \cdot \delta,\ b \end{array}} \qquad \boxed{\begin{array}{l} A \to \alpha \cdot \beta,\ c \\ B \to \gamma \cdot \delta,\ d \end{array}}$$

$$\boxed{\begin{array}{l} A \to \alpha \cdot \beta,\ ac \\ B \to \gamma \cdot \delta,\ bb \end{array}}$$



Is this LALR(1)?

$S' \to S$

$S \to A\,a$

$S \to B\,b$

$S \to b\,A\,b$

$S \to b\,B\,a$

$A \to a$

$B \to a$

LR(1) but not LALR(1)

$S' \to S$
$S \to A a$
$S \to B b$
$S \to b A b$
$S \to b B a$
$A \to a$
$B \to a$

LR(1) but not LALR(1)

$S' \to \cdot S, \$$

$S' \to S$
$S \to A a$
$S \to B b$
$S \to b A b$
$S \to b B a$
$A \to a$
$B \to a$

LR(1) but not LALR(1)

$S' \to \bullet S, \$$
$S \to \bullet A a, \$$
$S \to \bullet B b, \$$
$S \to \bullet b A b, \$$
$S \to \bullet b B a, \$$

$S' \to S$
$S \to Aa$
$S \to Bb$
$S \to bAb$
$S \to bBa$
$A \to a$
$B \to a$

LR(1) but not LALR(1)

$S' \to \bullet S, \$$
$S \to \bullet Aa, \$$
$S \to \bullet Bb, \$$
$S \to \bullet bAb, \$$
$S \to \bullet bBa, \$$
$A \to \bullet a, a$

$S' \to S$
$S \to A\,a$
$S \to B\,b$
$S \to b\,A\,b$
$S \to b\,B\,a$
$A \to a$
$B \to a$

LR(1) but not LALR(1)

$S' \to \bullet S, \$$
$S \to \bullet A\,a, \$$
$S \to \bullet B\,b, \$$
$S \to \bullet b\,A\,b, \$$
$S \to \bullet b\,B\,a, \$$
$A \to \bullet a, a$
$B \to \bullet a, b$

LR(1) but not LALR(1)

$S' \to S$
$S \to A\,a$
$S \to B\,b$
$S \to b\,A\,b$
$S \to b\,B\,a$
$A \to a$
$B \to a$

$$\boxed{\begin{array}{l} S' \to \bullet S,\ \$ \\ S \to \bullet A\,a,\ \$ \\ S \to \bullet B\,b,\ \$ \\ S \to \bullet b\,A\,b,\ \$ \\ S \to \bullet b\,B\,a,\ \$ \\ A \to \bullet a,\ a \\ B \to \bullet a,\ b \end{array}} \xrightarrow{a} \boxed{\begin{array}{l} A \to a \bullet,\ a \\ B \to a \bullet,\ b \end{array}}$$

LR(1)

$S' \to S$
$S \to A a$
$S \to B b$
$S \to b A b$
$S \to b B a$
$A \to a$
$B \to a$

LR(1) but not LALR(1)

$$\boxed{\begin{array}{l} S' \to \cdot S, \$ \\ S \to \cdot A a, \$ \\ S \to \cdot B b, \$ \\ S \to \cdot b A b, \$ \\ S \to \cdot b B a, \$ \\ A \to \cdot a, a \\ B \to \cdot a, b \end{array}} \xrightarrow{b} \begin{array}{l} S \to b \cdot A b, \$ \\ S \to b \cdot B a, \$ \end{array}$$

$$\xrightarrow{a} \boxed{\begin{array}{l} A \to a \cdot, a \\ B \to a \cdot, b \end{array}}$$

LR(1)

LR(1) but not LALR(1)

$S' \to S$
$S \to A a$
$S \to B b$
$S \to b A b$
$S \to b B a$
$A \to a$
$B \to a$

$S' \to \cdot S, \$$
$S \to \cdot A a, \$$
$S \to \cdot B b, \$$
$S \to \cdot b A b, \$$
$S \to \cdot b B a, \$$
$A \to \cdot a, a$
$B \to \cdot a, b$

b

$S \to b \cdot A b, \$$
$S \to b \cdot B a, \$$
$A \to \cdot a, b$

a

$A \to a \cdot, a$
$B \to a \cdot, b$

LR(1)

LR(1) but not LALR(1)

$S' \to S$
$S \to A\,a$
$S \to B\,b$
$S \to b\,A\,b$
$S \to b\,B\,a$
$A \to a$
$B \to a$

$S' \to \cdot S,\ \$$
$S \to \cdot A\,a,\ \$$
$S \to \cdot B\,b,\ \$$
$S \to \cdot b\,A\,b,\ \$$
$S \to \cdot b\,B\,a,\ \$$
$A \to \cdot a,\ a$
$B \to \cdot a,\ b$

$\xrightarrow{b}$

$S \to b \cdot A\,b,\ \$$
$S \to b \cdot B\,a,\ \$$
$A \to \cdot a,\ b$
$B \to \cdot a,\ a$

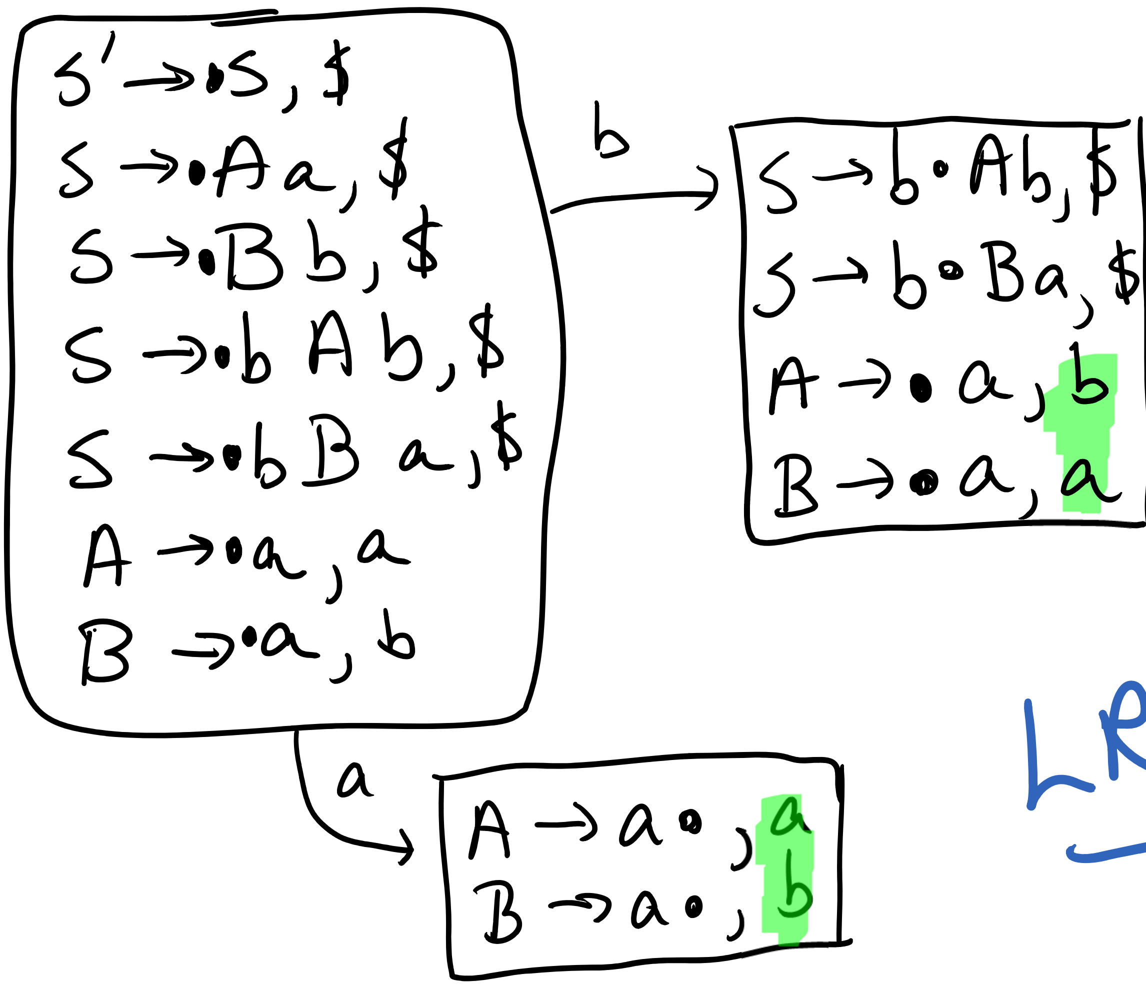
$\xrightarrow{a}$

$A \to a \cdot,\ a$
$B \to a \cdot,\ b$

LR(1)

LR(1) but not LALR(1)

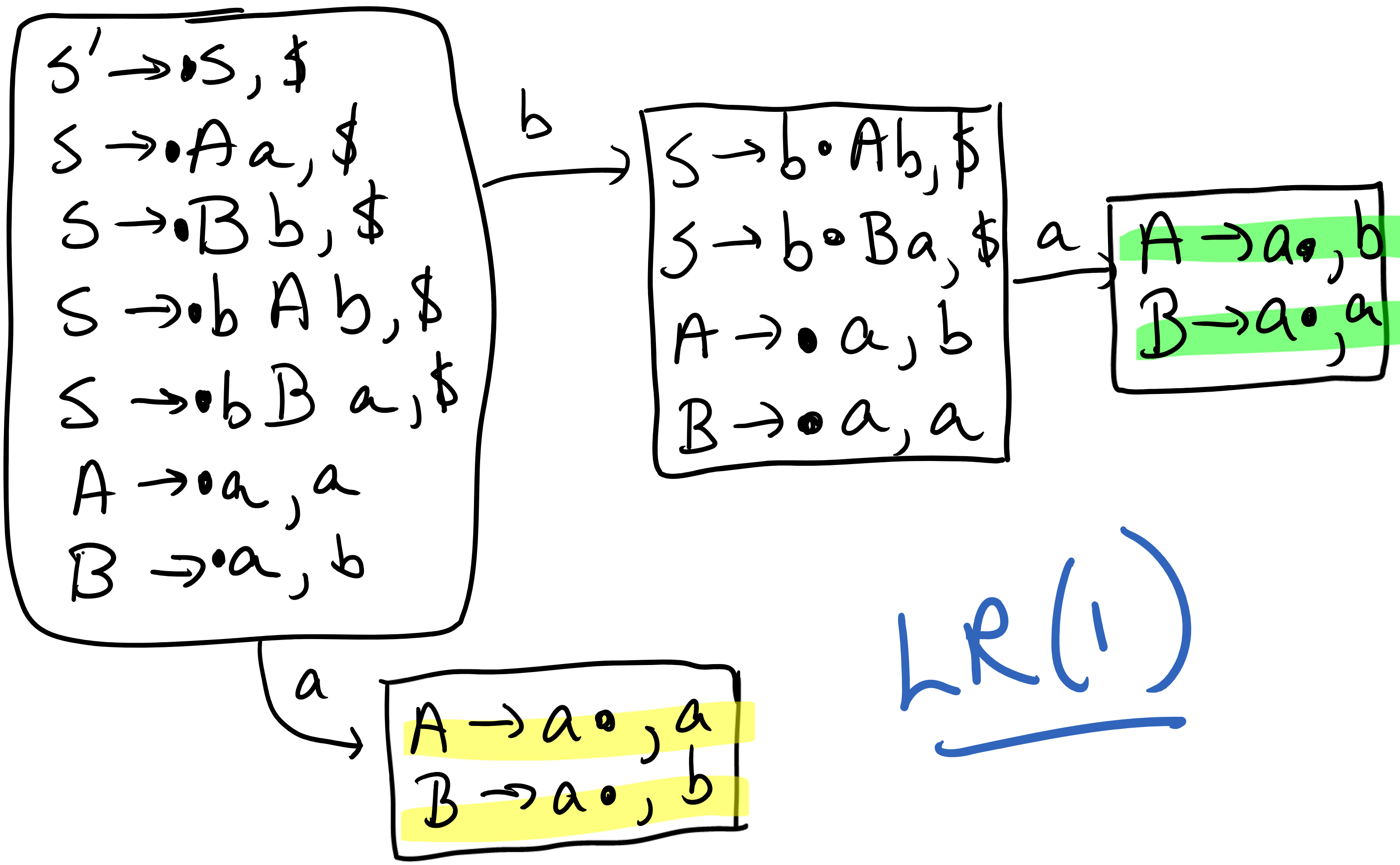
$S' \to S$
$S \to A\,a$
$S \to B\,b$
$S \to b\,A\,b$
$S \to b\,B\,a$
$A \to a$
$B \to a$



LR(1)

$S' \to S$
$S \to A a$
$S \to B b$
$S \to b A b$
$S \to b B a$
$A \to a$
$B \to a$

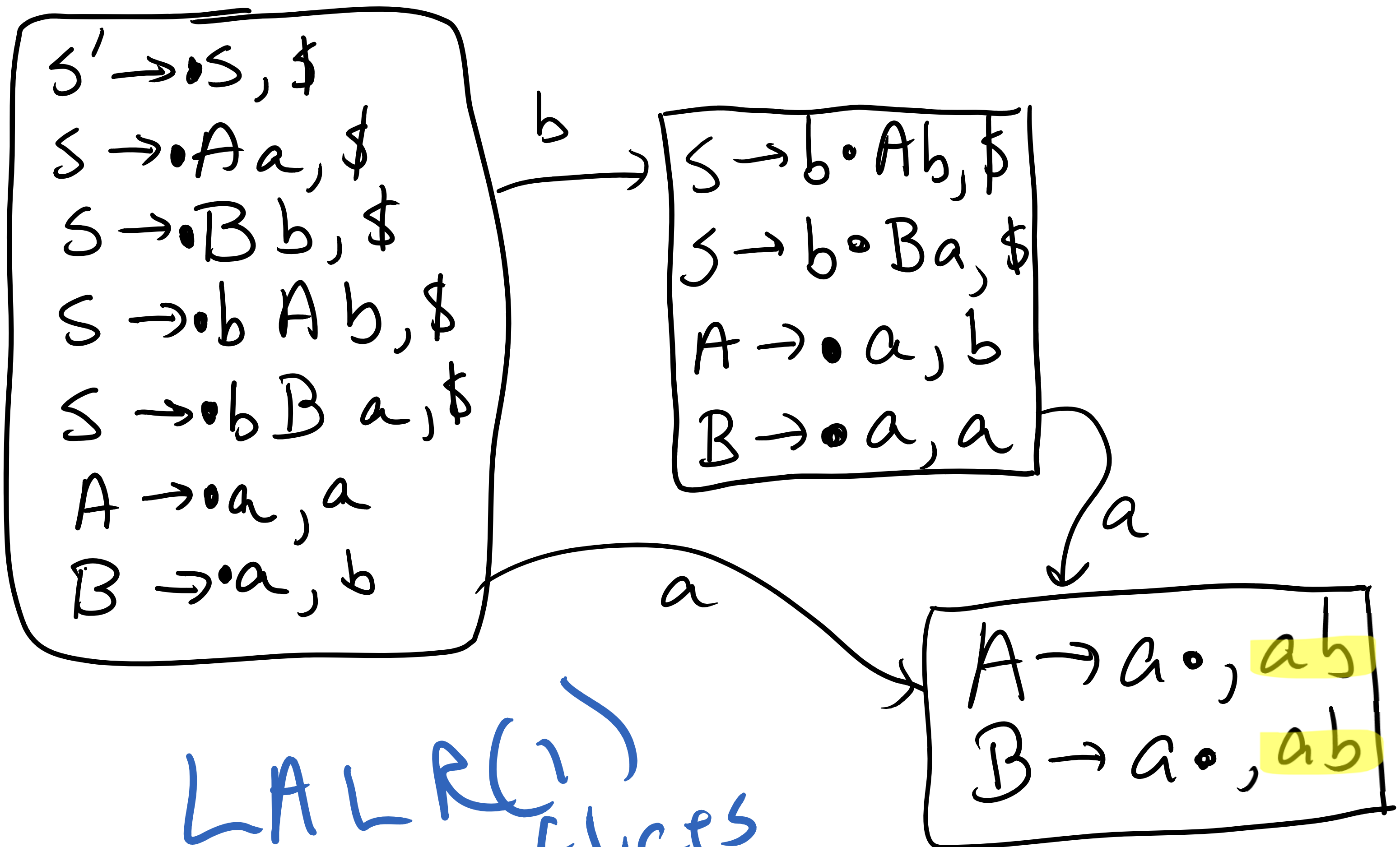
LR(1) but not LALR(1)



LR(1)

LR(1) but not LALR(1)

$S' \to S$
$S \to A\,a$
$S \to B\,b$
$S \to b\,A\,b$
$S \to b\,B\,a$
$A \to a$
$B \to a$



LALR(1) conflicts

LR parsing methods

SLR(1) — LR(0) item sets
Lookaheads from FOLLOW sets

LR(1) — LR(1) itemsets
Lookaheads computed directly

LALR(1) - LR(0) itemsets
Lookaheads from LR(1).

Announcements

- Midterm (here, during class) 1 week from today (Thursday).
- Covers all of WA1, PA1, WA2
- Solutions to WA2 will be distributed shortly after the due date.
- Exam policies:
  - Open book/open notes
  - Computers may be used to view downloaded notes.
  - No “computation”, no network access
  - No communication with others except questions to instructor and TAs (of course).

Precedence and Association in LR Parsers

Rewriting an ambiguous CFG to get the correct precedence, associativity

- Takes work
- CFG becomes complex
- Unit productions ($E \to T$, $T \to F$) slow down parsing

Idea - remove conflicting table entries.

Ambiguous expression grammar

$\text{FOLLOW}(E) = \{+, *,)\}$

$E \to E + E \bullet$

$E \to E \bullet + E$

$E \to E \bullet * E$

shift-reduce conflict on +

$n + n + n$

+ left or right associative?

reduce - left assoc

ACTION

'+'

n	$r_i, \cancel{s_j}$

Idea - remove conflicting table entries.

Ambiguous expression grammar

$\text{Follow}(E) = \{+, *,)\}$

$E \to E + E \bullet$

$E \to E \bullet + E$

$E \to E \bullet * E$

shift reduce conflict on *

n + n * n

shift or reduce?

precedence of + vs *

ACTION

	'*'	
n	~~ri~~, sk	

YACC, BYACC, BISON, CUP, etc.

Precedence declarations

%left '+'
%left '*'

YACC, BYACC, BISON, CUP, etc.

Precedence declarations

%left '+'
%left '*'

	next +	*
reduce $E \to E + E$		
reduce $E \to E * E$		

Connections With Formal Language Theory

Properties of Grammars vs Properties of Languages

"G is LL(1)" — no conflicts in LL(1) parse table.

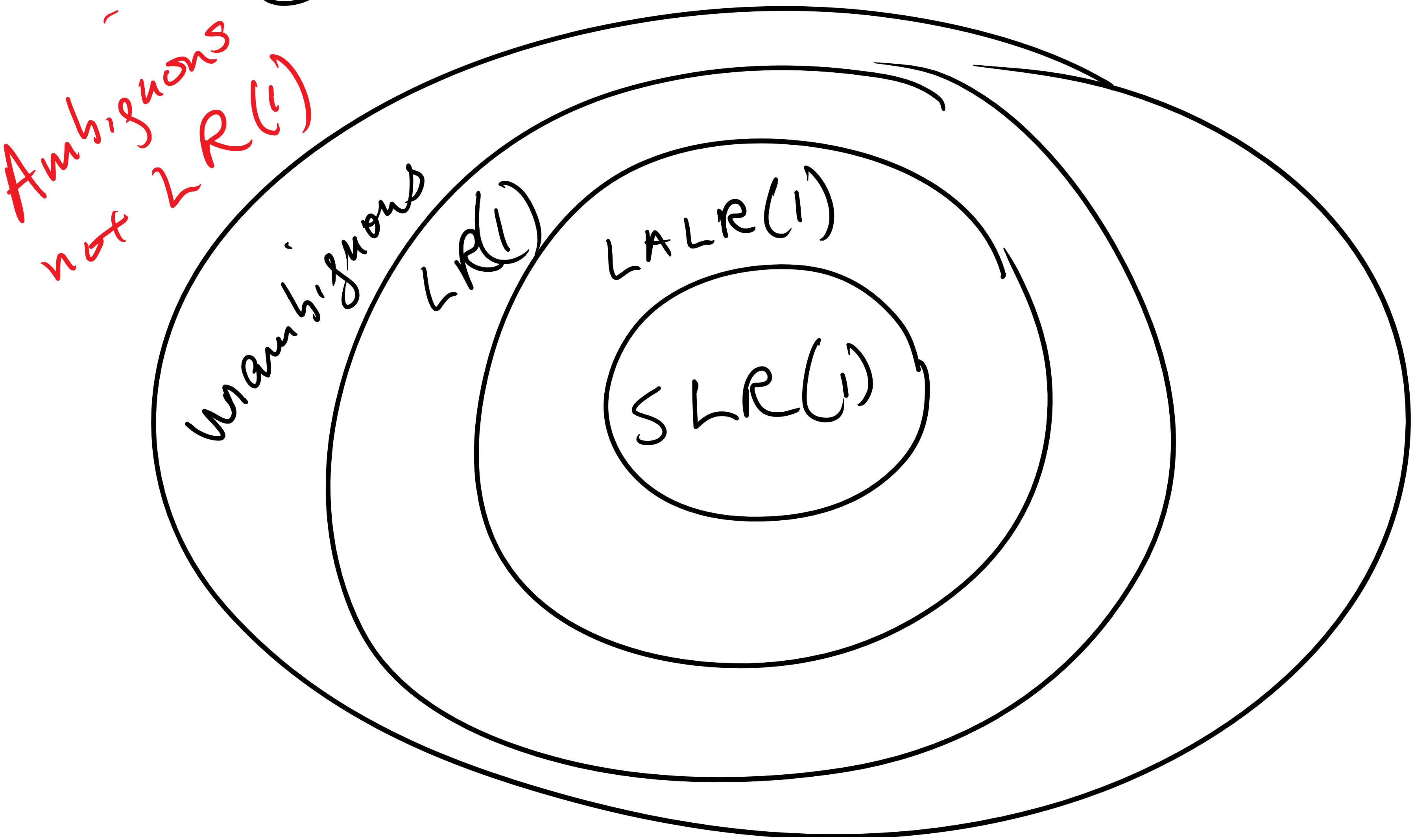
"$\mathcal{L}$ is LL(1)" — there <u>exists</u> an LL(1) grammar for $\mathcal{L}$.

$S \to S$
$S \to a$

Is grammar LL(1)?

Is language LL(1)?

Classes of Context-Free Grammars



$$
\begin{aligned}
S' &\to S \\
S &\to Aa \\
S &\to Bb \\
S &\to ac \\
A &\to a \\
B &\to a
\end{aligned}
$$

SLR(1)

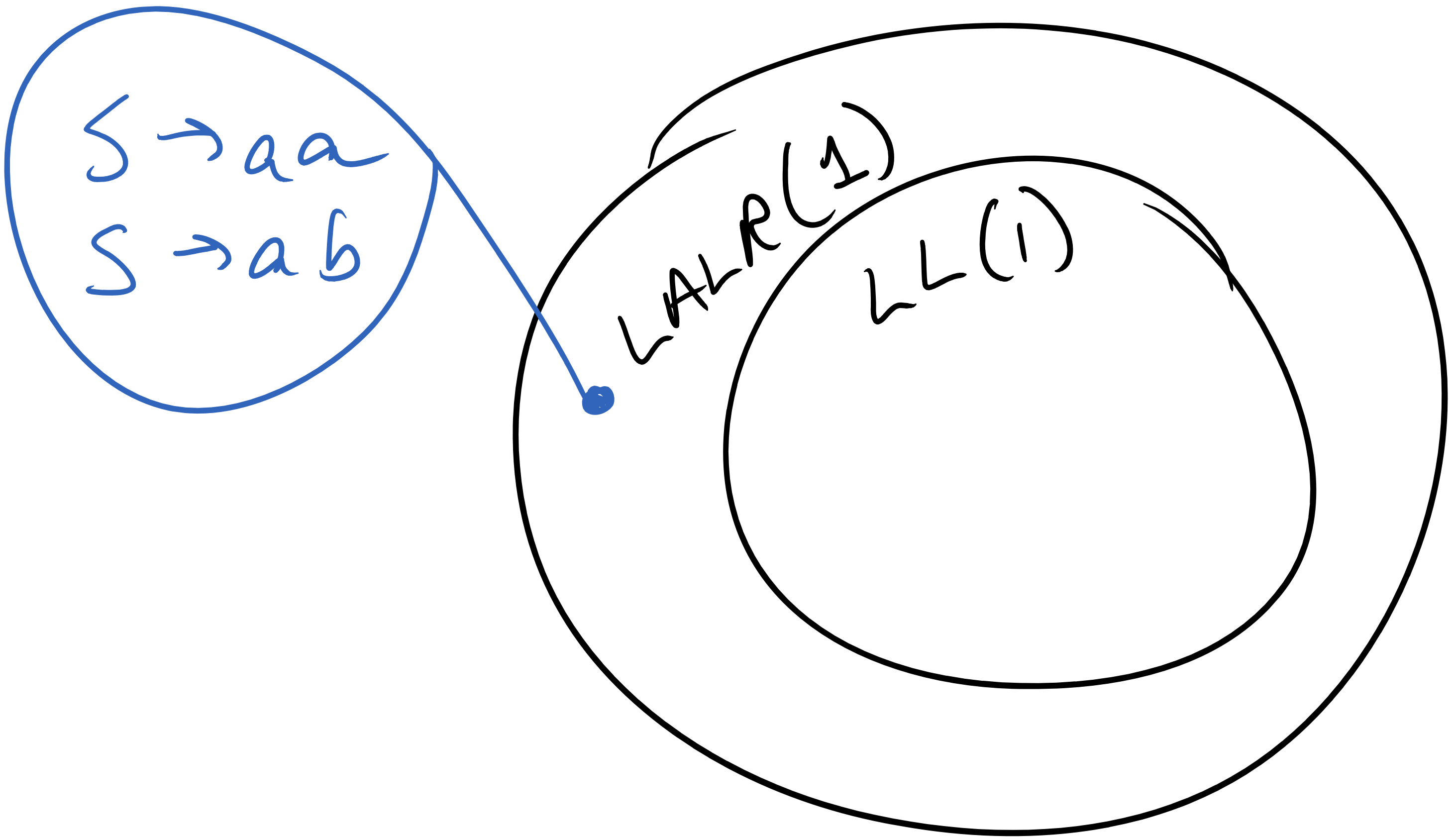
$$
\begin{aligned}
S' &\to S \\
S &\to Aa \\
S &\to Bb \\
S &\to bAb \\
A &\to a \\
B &\to a
\end{aligned}
$$

LALR(1)
(not SLR(1))

$$
\begin{aligned}
S' &\to S \\
S &\to Aa \\
S &\to Bb \\
S &\to bAb \\
S &\to bBa \\
A &\to a \\
B &\to a
\end{aligned}
$$

LR(1)
(not LALR(1))

Classes of Grammars



Intuition for why LR parsing is more powerful than LL parsing.

$A \to \alpha$
$A \to \beta$

LL parsing looks at first symbol of RHS of production to decide.

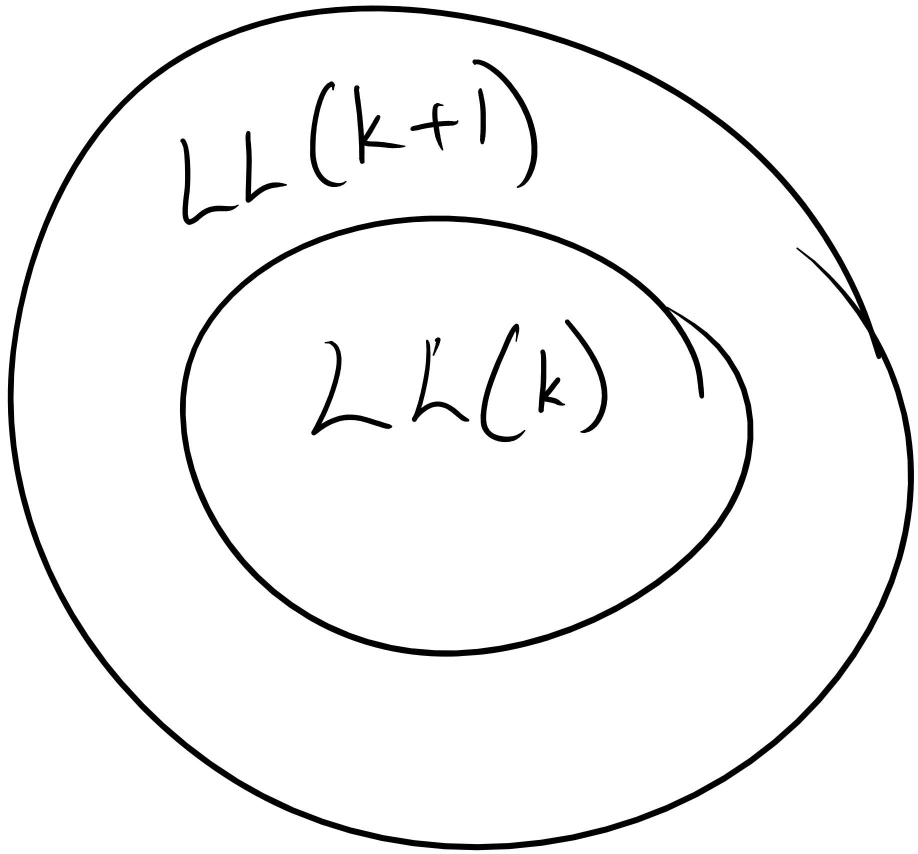
$A \to \alpha \bullet a$
$A \to \beta \bullet b$

LR parsing uses <u>all</u> of RHS <u>plus</u> following symbol(s) to decide.

Classes of Grammars

$S \to aa$
$S \to ab$

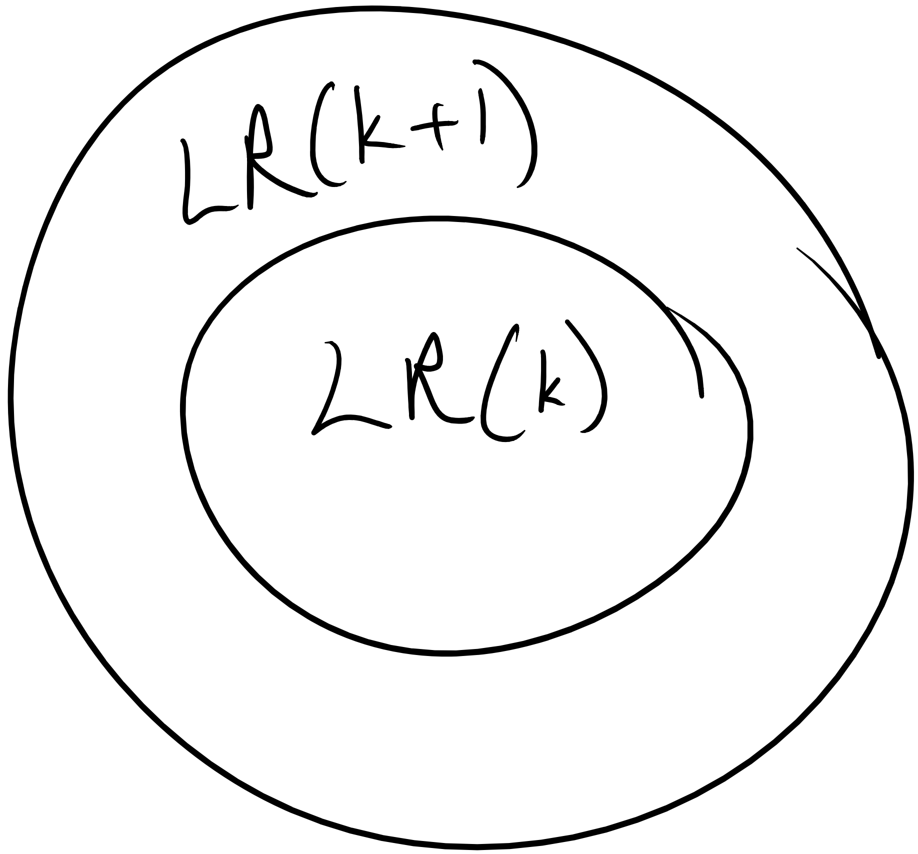
LL(2) but not LL(1)



Classes of Grammars

$S \to Aaa$
$S \to Bab$
$A \to a$
$B \to a$

LR(2)
but not
LR(1)



Undecidable Problems

Equivalence of CFGs

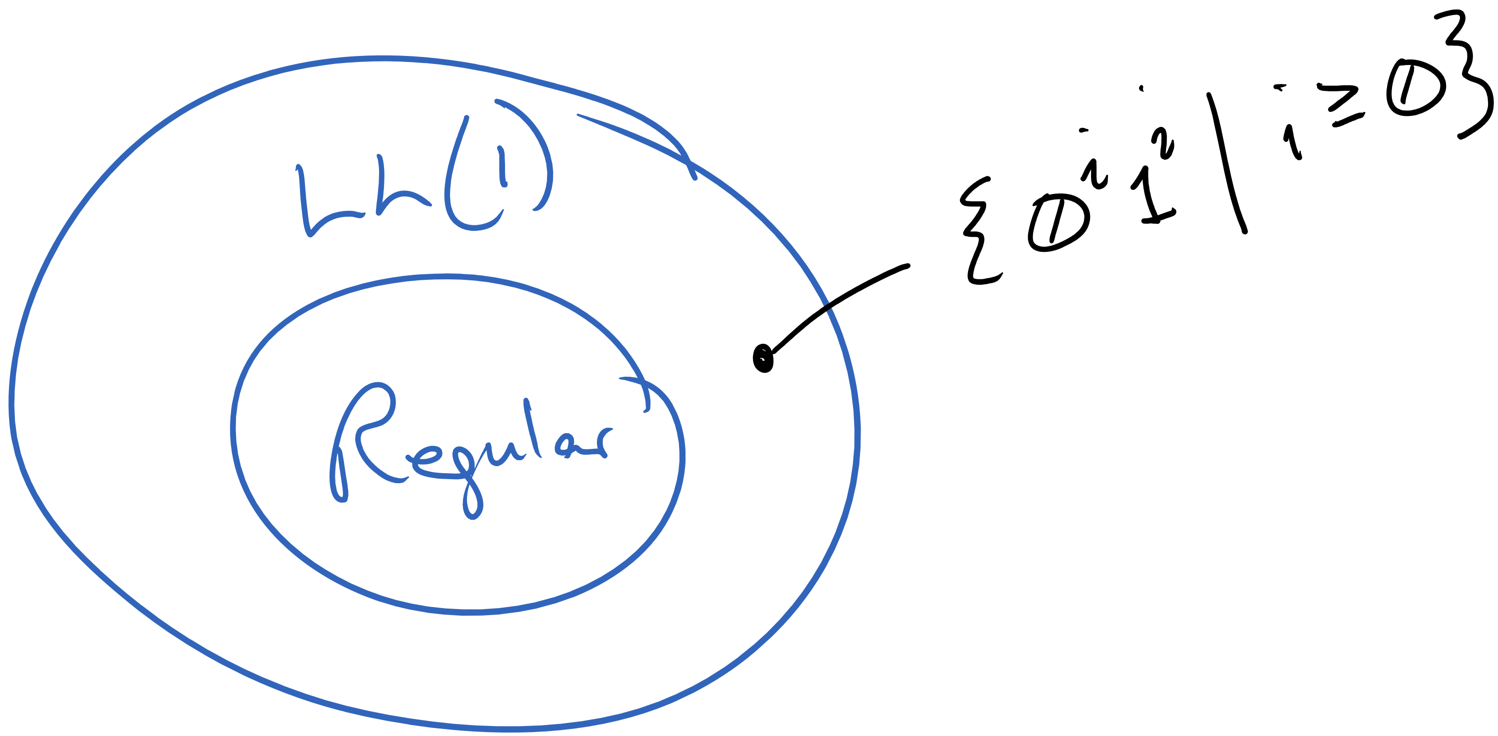
Same language?

Ambiguity of a CFG (!)

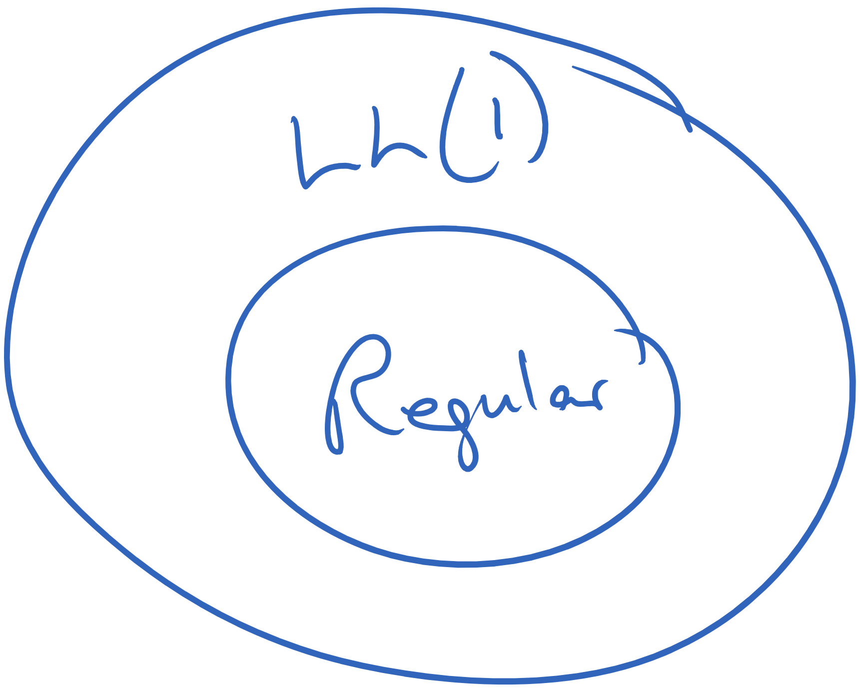
But LL(1), LR(1) etc <u>are</u> decidable

If LR(1) then not ambiguous.

Languages



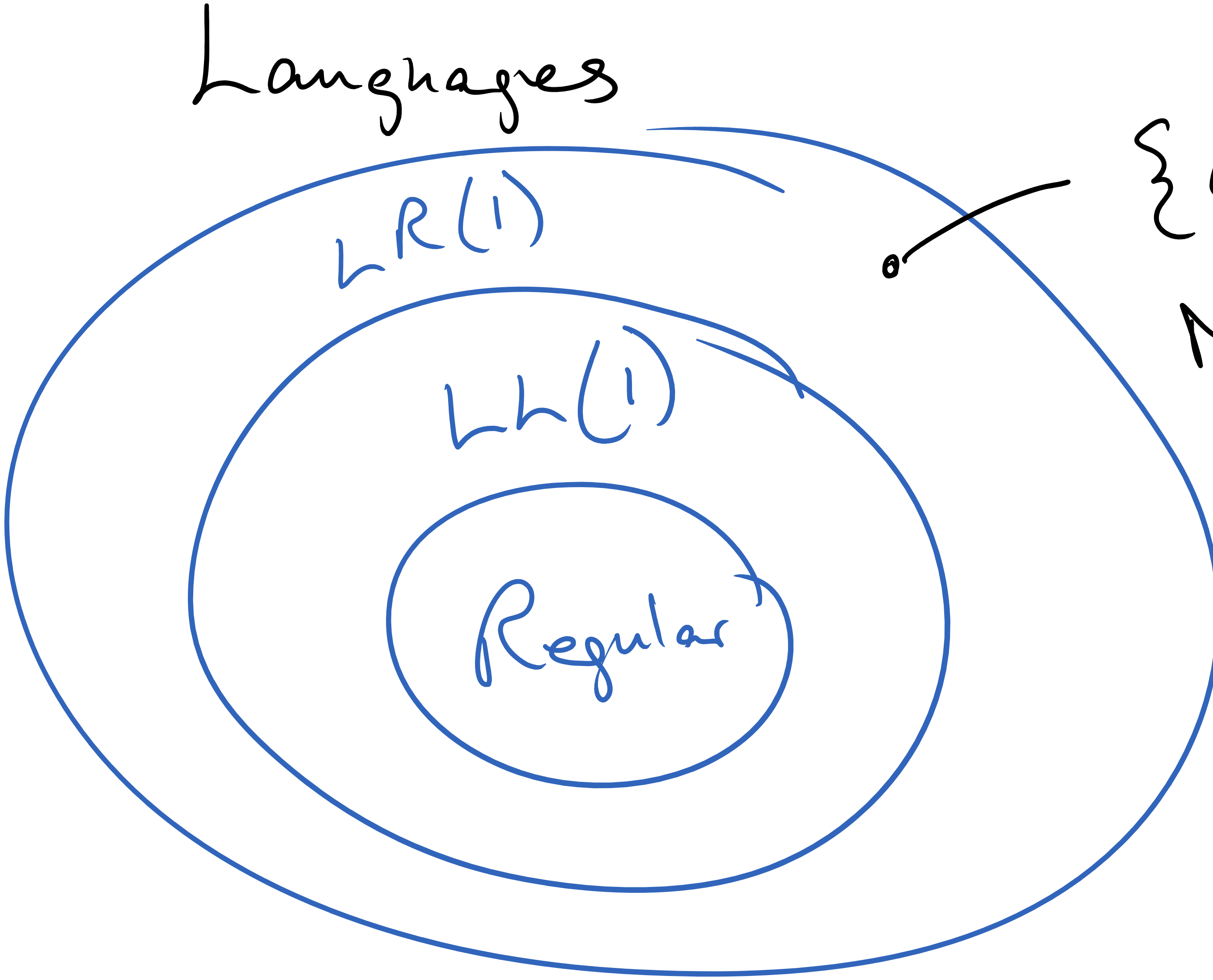
Languages



Every regular language can be written as a "right linear CFG"

$A \to w B$

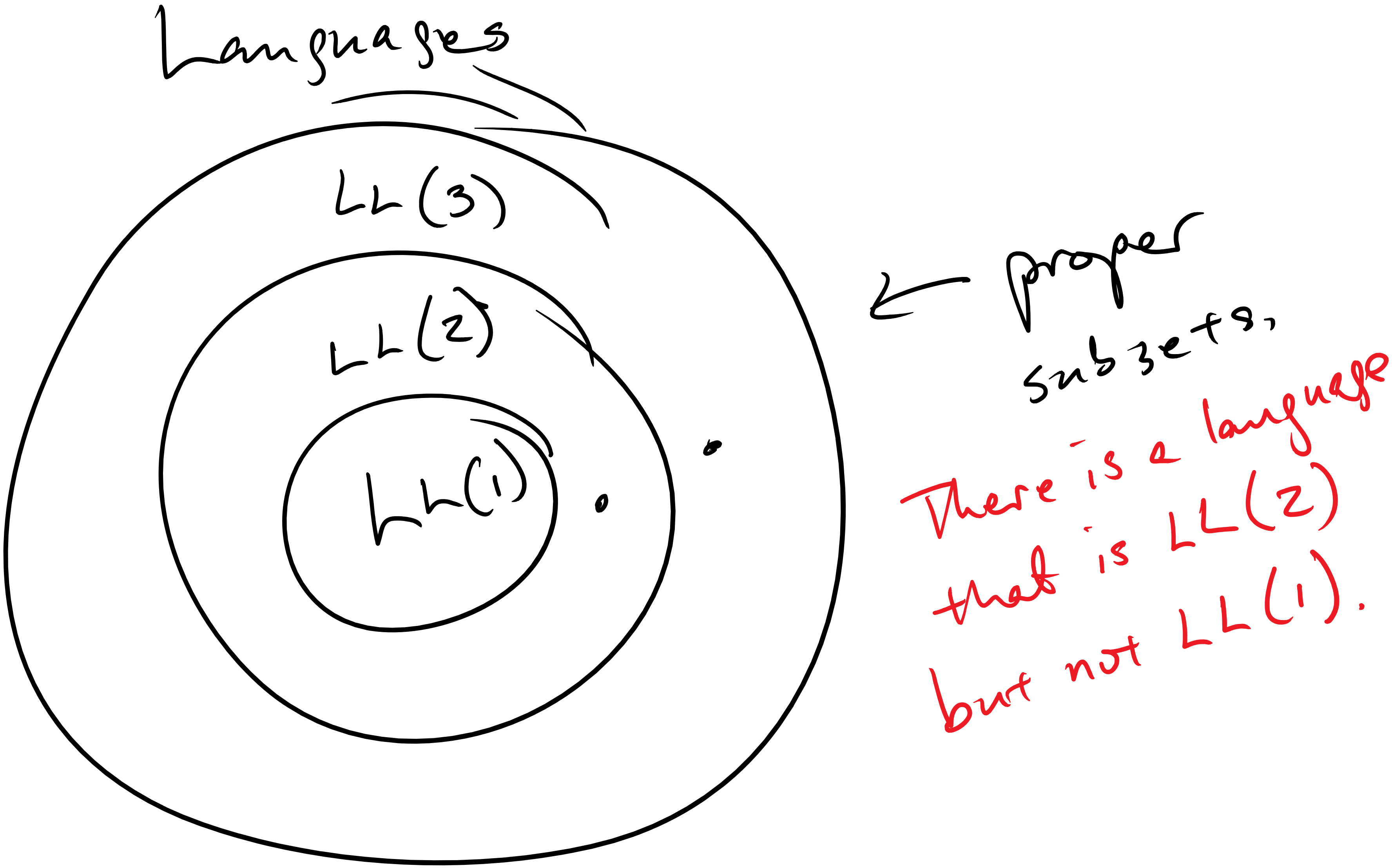
at most one nonterminal, always at right end.



$\{0^i 1^j \mid i \leq j\}$

No LL(1) grammar!

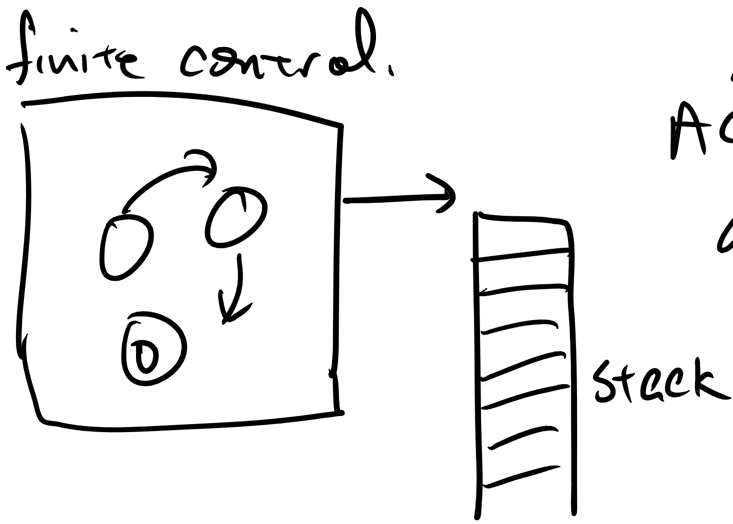
if then else has the same problem!


Languages
LL(3)
LL(2)
LL(1)
proper subsets,
There is a language that is LL(2) but not LL(1).

Languages

$LR(k) = LR(1)$ for all $k \geq 1$.

Deterministic Context-Free Languages.



Accepted by a deterministic push-down automaton

$\{ww^R \mid w \in \{a,b\}^*\}$ — not LR(1)

reverse

Context-free, but not deterministic

No LR(k) grammar.

$\left(\{wcw \mid w \in \{a,b\}^*\} \text{ — is LR(1)} \right)$

Equivalence of deterministic CFLs is decidable

Equivalence of two LR(1) CFG's is decidable

But probably not practical.