

Compiler Design

Fatemeh Deldar

Isfahan University of Technology

1403-1404

Constructing LALR Parsing Tables

- **LALR (LookAhead LR)**
- This method is often used in practice, because:
 - The tables obtained by it are considerably smaller than the canonical LR tables
 - Most common syntactic constructs of programming languages can be expressed conveniently by an LALR grammar
- The SLR and LALR tables for a grammar always have the same number of states
- **Example: For a language like C:**
 - The SLR and LALR tables have typically *several hundred states*
 - The canonical LR table would typically have *several thousand states*

Constructing LALR Parsing Tables

- We look for sets of LR(1) items having the same core, and merge these sets with common cores into one set of items
- *The merging of states with common cores can never produce a **shift/reduce** conflict that was not present in one of the original states, **because shift actions depend only on the core, not the lookahead***
- But it is possible that a merger will produce a **reduce/reduce** conflict

1. Construct $C = \{I_0, I_1, \dots, I_n\}$, the collection of sets of LR(1) items.
2. For each core present among the set of LR(1) items, find all sets having that core, and replace these sets by their union.

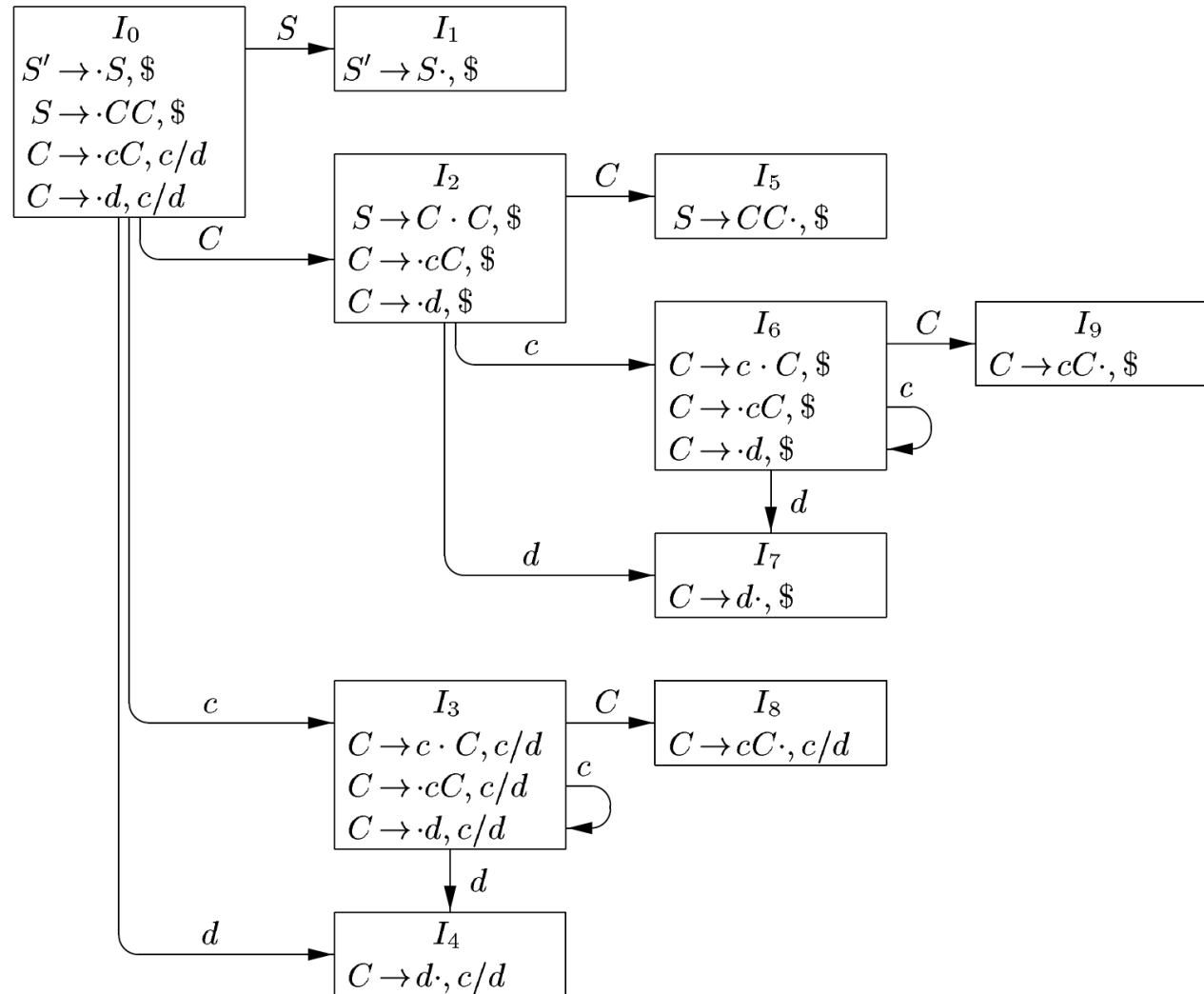
LALR(1) Grammar

- Example**

I_{36} : $C \rightarrow c \cdot C, c/d/\$$
 $C \rightarrow \cdot cC, c/d/\$$
 $C \rightarrow \cdot d, c/d/\$$

I_{47} : $C \rightarrow d \cdot, c/d/\$$

I_{89} : $C \rightarrow cC \cdot, c/d/\$$



LALR(1) Grammar

- **Example**

I_{36} : $C \rightarrow c \cdot C, c/d/\$$
 $C \rightarrow \cdot cC, c/d/\$$
 $C \rightarrow \cdot d, c/d/\$$

I_{47} : $C \rightarrow d \cdot, c/d/\$$

I_{89} : $C \rightarrow cC \cdot, c/d/\$$

| STATE | ACTION | | | GOTO | |
|-------|----------|----------|-----------|----------|----------|
| | <i>c</i> | <i>d</i> | <i>\$</i> | <i>S</i> | <i>C</i> |
| 0 | s3 | s4 | | 1 | 2 |
| 1 | | | acc | | |
| 2 | s6 | s7 | | | 5 |
| 3 | s3 | s4 | | | 8 |
| 4 | r3 | r3 | | | |
| 5 | | | r1 | | |
| 6 | s6 | s7 | | | 9 |
| 7 | | | r3 | | |
| 8 | r2 | r2 | | | |
| 9 | | | r2 | | |



| STATE | ACTION | | | GOTO | |
|-------|----------|----------|-----------|----------|----------|
| | <i>c</i> | <i>d</i> | <i>\$</i> | <i>S</i> | <i>C</i> |
| 0 | s36 | s47 | | 1 | 2 |
| 1 | | | acc | | |
| 2 | s36 | s47 | | | 5 |
| 36 | s36 | s47 | | | 89 |
| 47 | r3 | r3 | r3 | | |
| 5 | | | r1 | | |
| 89 | r2 | r2 | r2 | | |

LALR(1) Grammar

- **Example: Reduce/Reduce Conflict**
 - The grammar is not LALR(1)

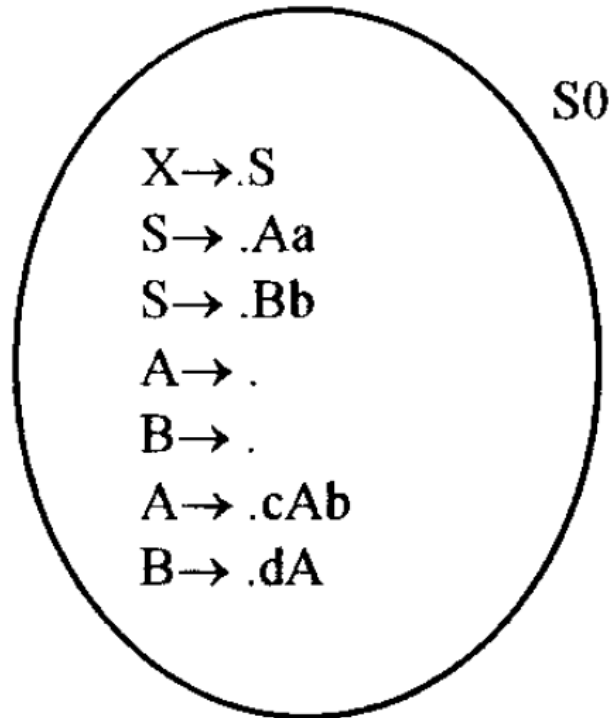
$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & a A d \mid b B d \mid a B e \mid b A e \\ A & \rightarrow & c \\ B & \rightarrow & c \end{array}$$
$$\begin{array}{l} \{[A \rightarrow c\cdot, d], [B \rightarrow c\cdot, e]\} \\ \{[A \rightarrow c\cdot, e], [B \rightarrow c\cdot, d]\} \end{array}$$

$$\begin{array}{l} A \rightarrow c\cdot, d/e \\ B \rightarrow c\cdot, d/e \end{array}$$

LALR(1) Grammar

- **Example:**
 - Reduce/Reduce conflict in SLR(1) automata
 - Not SLR(1)

$S \rightarrow Aa$
 $S \rightarrow Bb$
 $A \rightarrow \epsilon$
 $B \rightarrow \epsilon$
 $A \rightarrow cAb$
 $B \rightarrow dAa$



LALR(1) Grammar

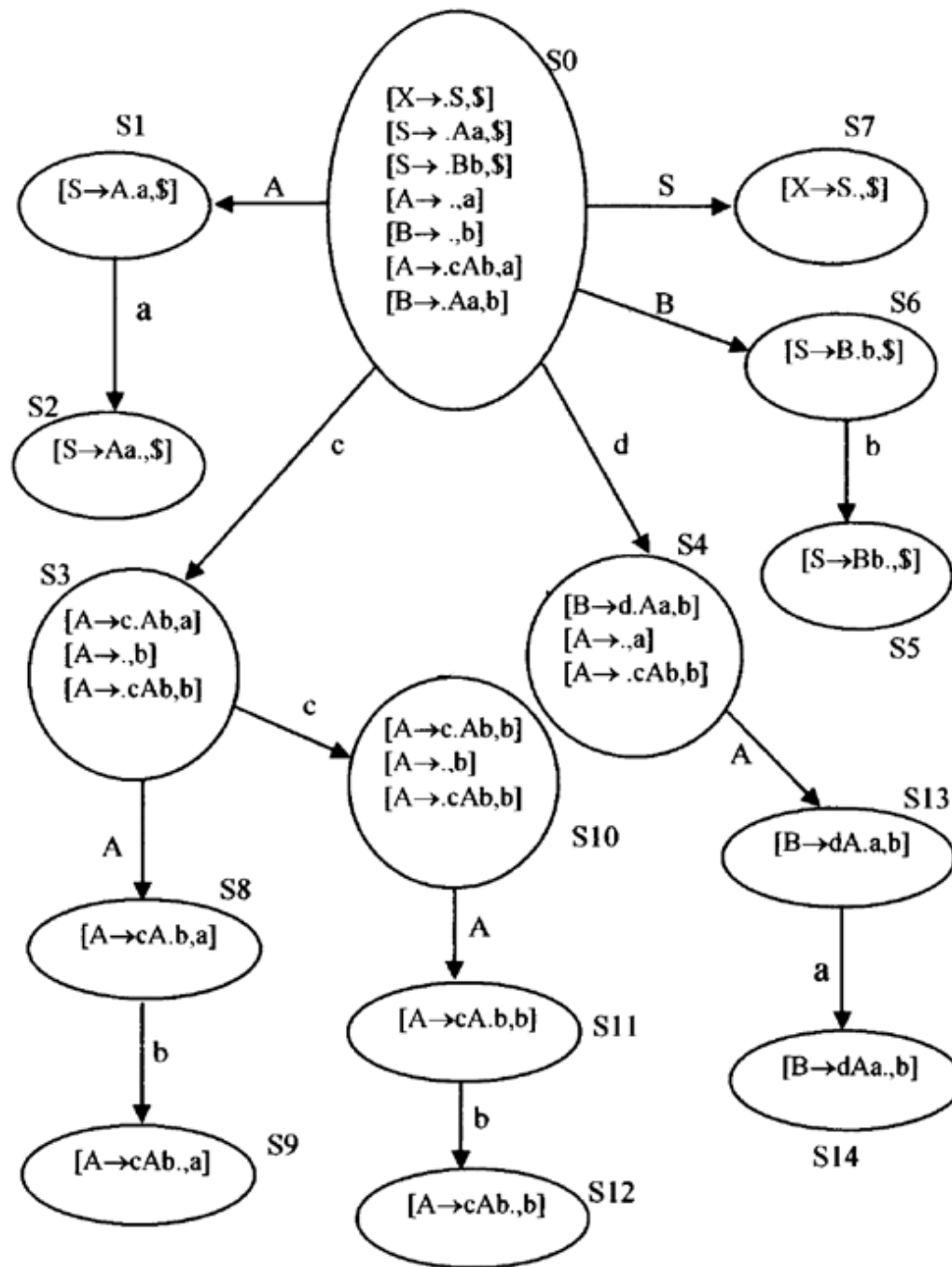
- Example:**

- No conflict in LR(1) automata
 - LR(1)

$S \rightarrow Aa$
 $S \rightarrow Bb$
 $A \rightarrow \epsilon$
 $B \rightarrow \epsilon$
 $A \rightarrow cAb$
 $B \rightarrow dAa$

ایرادات موجود در دیاگرام رسم شده:

- در وضعیت صفر آخرین آیتم $B \rightarrow \cdot dAa, b$ است.
- در وضعیت ۴ آخرین آیتم $A \rightarrow \cdot cAb, a$ است.
- وضعیت ۴ با C به وضعیت ۳ می‌رود.
- وضعیت ۱۰ با C به خودش برمی‌گردد.



LALR(1) Grammar

- **Example:**
 - No conflict in LALR(1) automata
 - LALR(1)

| | | |
|--|---|--|
| S8:[A→cA.b,a] | S11:[A→cA.b,b] | S8,11:[A→cA.b,ab] |
| S9:[A→cAb.,a] | S12:[A→cAb.,b] | S9,12:[A→cAb.,ab] |
| S3:[A→c.Ab,a] [A→.,b] [A→.cAb,b] | S10:[A→c.Ab,b] [A→.,b] [A→.cAb,b] | S3,10:[A→c.Ab,ab] [A→.,b] [A→.cAb,b] |