BITS Pilani
Pilani Campus

LR Parsing

Dr. Shashank Gupta
Assistant Professor
Department of Computer Science and Information Systems

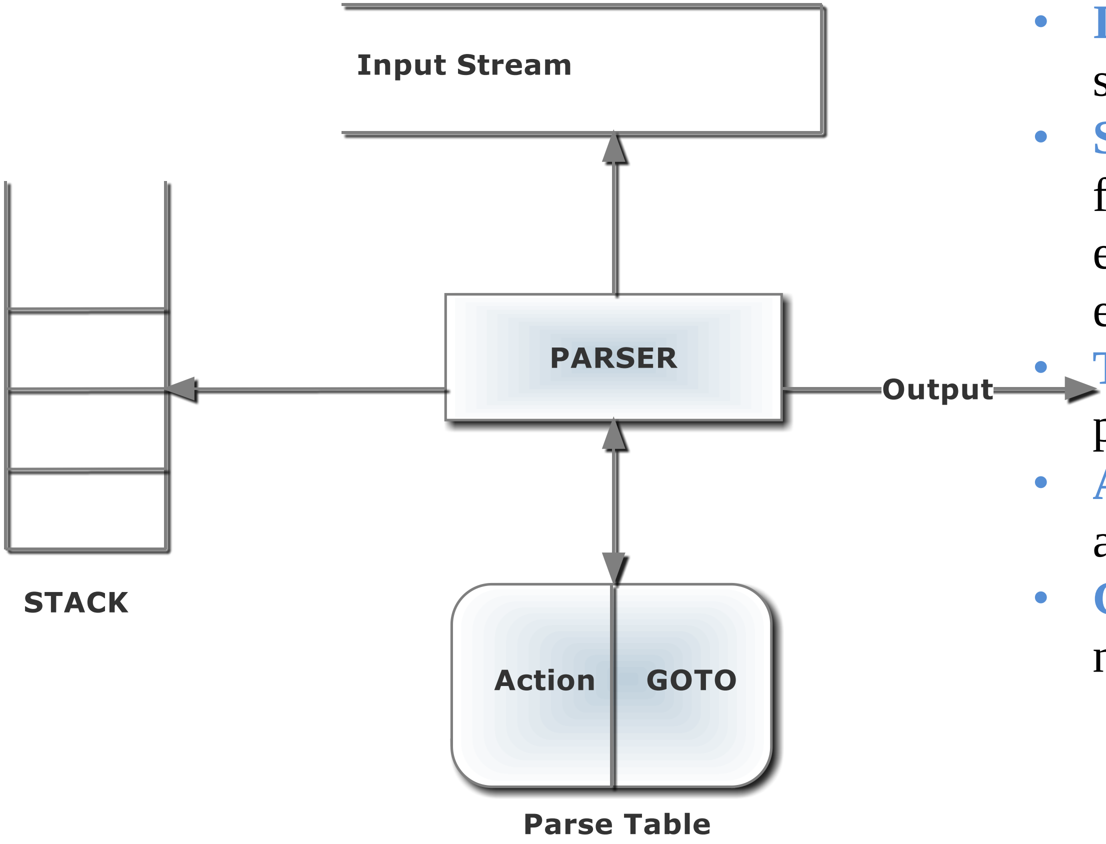
Issues in Bottom-up Parser

Whether to shift or reduce?

Which production to use for reduction?

LR(0) Parsing

LR(0) Parsing



- **Input Stream** contains the input string
- **Stack** contains a string of the form $S_0X_1 \; S_1X_2\ldots\ldots X_nS_n$ where each X_i is a grammar symbol and each S_i is a state.
- **Table** contains action and goto parts.
- **Action** table is indexed by state and terminal symbols.
- **Goto** table is indexed by state and non terminal symbols.

Augmentation of Grammar

- G is a grammar with start symbol S.
- The augmented grammar G` for G has a new start symbol S` and an additional production

  S`→ S

- When the parser reduces by this new rule, it will stop immediately with accept state.

Construction of LR (0) Items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the RHS.
- Thus, production A→XYZ gives four LR(0) items

  A → .XYZ

  A → X.YZ

  A → XY.Z

  A → XYZ.

Each item indicates how much of a production has been seen at a point in the process of parsing.

Closure Operation

- Let I be a set of items for a grammar G
- Closure(I) is a set constructed as follows:
  - Every item in I is in closure (I)
  - If $A \rightarrow \alpha.B\beta$ is in closure(I) and $B \rightarrow \gamma$ is a production then $B \rightarrow .\gamma$ is in closure(I)

Closure Operation

$$E' \rightarrow E$$

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid id$$

If I is { E` → .E } then closure(I) is

$$E^{'} \rightarrow .E$$

$$E \rightarrow .E + T$$

$$E \rightarrow .T$$

$$T \rightarrow .T * F$$

$$T \rightarrow .F$$

$$F \rightarrow .(E)$$

$$F \rightarrow .id$$

Goto Operation

Goto (I,X) , (where I is a set of items and X is a grammar symbol), is closure of set of item

$A \rightarrow \alpha X.\beta$ such that $A \rightarrow \alpha.X\beta$ is in I.

If I is a set of items for some valid prefix α then goto (I,X) is set of valid items for prefix αX

Goto Operation

- If I is {E → E. + T } then goto (I,+) is

$$E \rightarrow E + .T$$

$$T \rightarrow .T * F$$

$$T \rightarrow .F$$

$$F \rightarrow .(E)$$

$$F \rightarrow .id$$

$$E' \rightarrow E$$

$$E \rightarrow E + T | T$$

$$T \rightarrow T * F | F$$

$$F \rightarrow (E) | id$$

Steps in LR(0) Parsing

1. Design Augmented Grammar

2. Construct Goto Graph using Closure and Goto Operations on LR (0) Items.

3. Design Parsing Table

4. Execute Parsing on a String.

LR(0) Parser Example

Construct a LR (0) parsing table for the following grammar

$$S \rightarrow A\,A$$

$$A \rightarrow a\,A | b$$

In addition, parse the following i/p: aabb using LR (0) parsing table.

Augment the Grammar

$$S \rightarrow A\,A$$

$$A \rightarrow a\,A \mid b$$

$$0: S' \rightarrow S$$

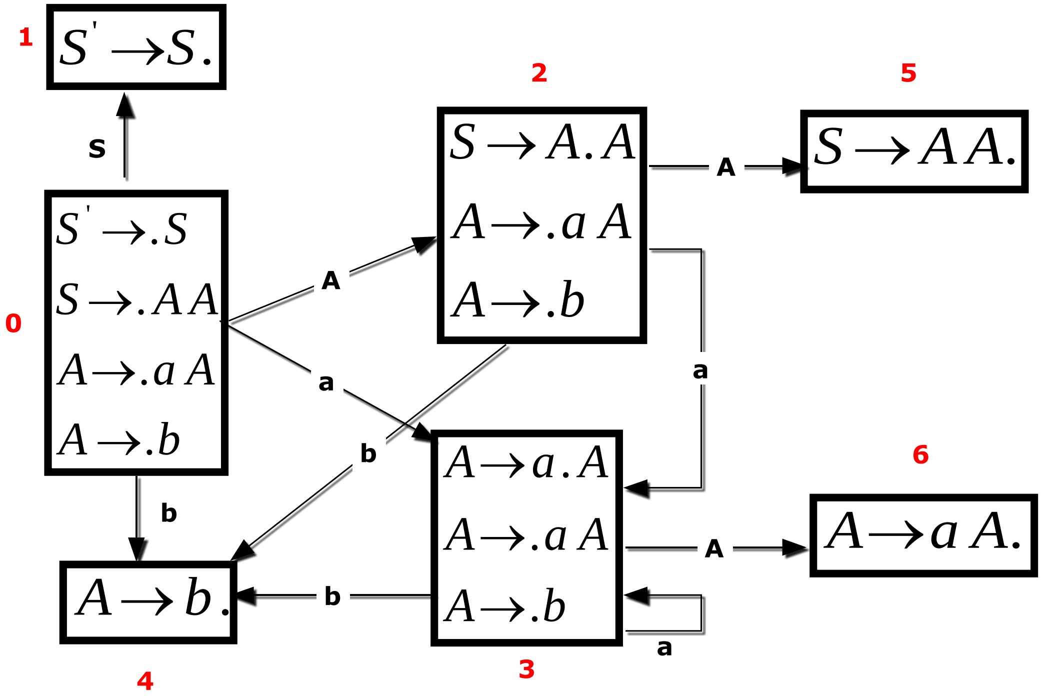
$$1: S \rightarrow A\,A$$

$$2: A \rightarrow a\,A$$

$$3: A \rightarrow b$$

Goto Graph

GOTO GRAPH



$$0: S' \to S$$

$$1: S \to AA$$

$$2: A \to aA$$

$$3: A \to b$$