Syntax Analysis: Bottom-Up Parsing

Lecture 5

Objectives

By the end of this lecture you should be able to:

1. Identify bottom-up parsing.
2. Construct reductions of a given string and grammar.
3. Construct a shift-reduce PDA.
4. Identify types of shift-reduce conflicts.

Outline

Outline

What is Bottom-Up Parsing?

Definition

Bottom-up parsing consists in the construction of a parse tree starting with the leaves up towards the root.

A particular style of bottom-up parsing may be equivalently defined thus:

Definition

Bottom-up parsing consists in finding a right-most derivation of a given string in a given CFG.

What is Bottom-Up Parsing?

Definition

Bottom-up parsing consists in the construction of a parse tree starting with the leaves up towards the root.

A particular style of bottom-up parsing may be equivalently defined thus:

Definition

Bottom-up parsing consists in finding a right-most derivation of a given string in a given CFG.

Example: G_2

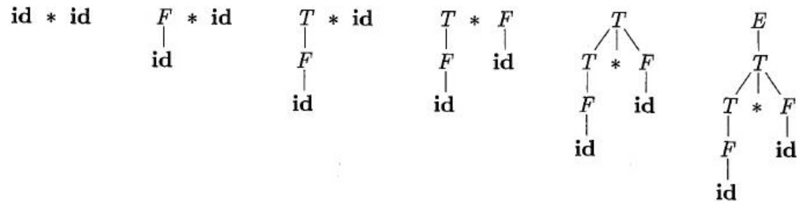
Example

$$
\begin{array}{lcl}
E & \longrightarrow & E + T \mid T \\
T & \longrightarrow & T * F \mid F \\
F & \longrightarrow & (\ E\ ) \mid \textbf{id} \mid \textbf{number}
\end{array}
$$

Input: $\textbf{id} * \textbf{id}$.

Example: Bottom-Up Parsing

Example



© Aho et al. (2007)

Reductions

- A reduction is the reverse of a derivation.

Definition

A sentential form α reduces to a sentential form β, denoted $\alpha \overset{*}{\mapsto} \beta$ if $\beta \overset{*}{\Rightarrow} \alpha$.

Example

$$\mathbf{id}*\mathbf{id} \mapsto F*\mathbf{id} \mapsto T*\mathbf{id} \mapsto T*F \mapsto T \mapsto E$$

- At every step of a reduction, we replace a sentential form by a variable.

Reductions

- A reduction is the reverse of a derivation.

Definition

A sentential form α reduces to a sentential form β, denoted $\alpha \overset{*}{\mapsto} \beta$ if $\beta \overset{*}{\Rightarrow} \alpha$.

Example

$$\mathbf{id}*\mathbf{id} \mapsto F*\mathbf{id} \mapsto T*\mathbf{id} \mapsto T*F \mapsto T \mapsto E$$

- At every step of a reduction, we replace a sentential form by a variable.

Bottom-Up Parsing, Again

Definition

Given a CFG $G = \langle V, \Sigma, R, S \rangle$ and $w \in \Sigma^*$, bottom-up parsing, with a left-to-right scan of the input, is the problem of constructing a reduction from w to S such that if

$$\alpha\gamma\beta \mapsto \alpha A \beta$$

is a step in the reduction, then no subsequent step may be of the form

$$\delta_1\delta_2\delta_3\lambda \mapsto \delta_1 B \delta_3 \lambda$$

where $A\beta \overset{*}{\mapsto} \lambda$ and $\delta_1\delta_2\delta_3 = \alpha$.

Thus constrained, such a reduction is the reverse of a right-most derivation.

Bottom-Up Parsing, Again

Definition

Given a CFG $G = \langle V, \Sigma, R, S\rangle$ and $w \in \Sigma^*$, bottom-up parsing, with a left-to-right scan of the input, is the problem of constructing a reduction from w to S such that if

$$\alpha\gamma\beta \mapsto \alpha A\beta$$

is a step in the reduction, then no subsequent step may be of the form

$$\delta_1\delta_2\delta_3\lambda \mapsto \delta_1 B\delta_3\lambda$$

where $A\beta \overset{*}{\mapsto} \lambda$ and $\delta_1\delta_2\delta_3 = \alpha$.

Thus constrained, such a reduction is the reverse of a right-most derivation.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.
   - 1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   - 1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   - 1c. $SF = \alpha_1 A \delta$.
   - 1d. $\alpha = \alpha_1$.

   - 2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   - 2b. Choose $(A \longrightarrow \delta) \in R$.
   - 2c. $SF = \alpha\gamma A \lambda$.
   - 2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A \delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A\lambda$.
   2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.
   - 1a. Choose $\gamma (\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   - 1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   - 1c. $SF = \alpha_1 A \delta$.
   - 1d. $\alpha = \alpha_1$.
   - 2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   - 2b. Choose $(A \longrightarrow \delta) \in R$.
   - 2c. $SF = \alpha\gamma A\lambda$.
   - 2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.
   - 1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   - 1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   - 1c. $SF = \alpha_1 A\delta$.
   - 1d. $\alpha = \alpha_1$.
   - 2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   - 2b. Choose $(A \longrightarrow \delta) \in R$.
   - 2c. $SF = \alpha\gamma A\lambda$.
   - 2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.
   - 1a. Choose $\gamma (\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   - 1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   - 1c. $SF = \alpha_1 A \delta$.
   - 1d. $\alpha = \alpha_1$.

   - 2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   - 2b. Choose $(A \longrightarrow \delta) \in R$.
   - 2c. $SF = \alpha\gamma A \lambda$.
   - 2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.
   - 1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   - 1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   - 1c. $SF = \alpha_1 A \delta$.
   - 1d. $\alpha = \alpha_1$.
   - 2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   - 2b. Choose $(A \longrightarrow \delta) \in R$.
   - 2c. $SF = \alpha\gamma A\lambda$.
   - 2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S\rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon), \delta, \alpha_1$, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A\delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A\lambda$.
   2d. $\alpha = \alpha\gamma$.

4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

    1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
    1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
    1c. $SF = \alpha_1 A\delta$.
    1d. $\alpha = \alpha_1$.

    2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
    2b. Choose $(A \longrightarrow \delta) \in R$.
    2c. $SF = \alpha\gamma A\lambda$.
    2d. $\alpha = \alpha\gamma$.

4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A\delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A\lambda$.
   2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S\rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A\delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A\lambda$.
   2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon), \delta, \alpha_1$, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A \delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A \lambda$.
   2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A\delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A\lambda$.
   2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S\rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A\delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A\lambda$.
   2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S\rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A\delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A\lambda$.
   2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S \rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

   1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
   1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
   1c. $SF = \alpha_1 A\delta$.
   1d. $\alpha = \alpha_1$.

   2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
   2b. Choose $(A \longrightarrow \delta) \in R$.
   2c. $SF = \alpha\gamma A\lambda$.
   2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

General Structure of a Bottom-Up Parser

Given $G = \langle V, \Sigma, R, S\rangle$ and w.

1. $SF = w$.
2. $\alpha = \varepsilon$.
3. For $SF = \alpha\beta$, choose one the following.

    1a. Choose $\gamma(\neq \varepsilon)$, δ, α_1, and α_2 such that $\beta = \gamma\delta$ and $\alpha = \alpha_1\alpha_2$.
    1b. Choose $(A \longrightarrow \alpha_2\gamma) \in R$.
    1c. $SF = \alpha_1 A\delta$.
    1d. $\alpha = \alpha_1$.

    2a. Choose γ, δ and, λ such that $\beta = \gamma\delta\lambda$.
    2b. Choose $(A \longrightarrow \delta) \in R$.
    2c. $SF = \alpha\gamma A\lambda$.
    2d. $\alpha = \alpha\gamma$.
4. If no choices are available, then fail.
5. If $SF = S$ then return.
6. Goto 3.

$\alpha_2\gamma$ in 1b and δ in 2b are referred to as handles of SF.

Outline

PDA Perspective

- You may think of a shift-reduce parser as a PDA P.
- P has three *main* states: q_s, q_{loop}, and q_a.
- In q_s, it pushes \$ and enters q_{loop}.
- While in q_{loop}
  - nondeterministically, do one of the following:
    1. Shift, by reading an input symbol and pushing it onto the stack.
    2. Reduce, by popping zero or more stack symbols $s_1 \cdots s_n$, with s_n on top of the stack, and pushing a variable A, where $A \longrightarrow s_1 \cdots s_n$ is a nondeterministically chosen rule of the grammar.
    3. Pop the start symbol and then \$ and enter q_a.

Exercise

Example

Draw the state diagram of the shift-reduce PDA which is equivalent to the CFG G_2.

Do it yourself.

Exercise (Cont'd)

Example

STACK	INPUT	ACTION
$\$$	$\mathbf{id}_1 * \mathbf{id}_2 \,\$$	shift
$\$\,\mathbf{id}_1$	$* \mathbf{id}_2 \,\$$	reduce by $F \to \mathbf{id}$
$\$\,F$	$* \mathbf{id}_2 \,\$$	reduce by $T \to F$
$\$\,T$	$* \mathbf{id}_2 \,\$$	shift
$\$\,T *$	$\mathbf{id}_2 \,\$$	shift
$\$\,T * \mathbf{id}_2$	$\$$	reduce by $F \to \mathbf{id}$
$\$\,T * F$	$\$$	reduce by $T \to T * F$
$\$\,T$	$\$$	reduce by $E \to T$
$\$\,E$	$\$$	accept

Conflicts

- Given the nondeterminism of a shift-reduce parser, two types of conflicts may be encountered during parsing:
  1. Shift/Reduce conflict, which occurs when it is possible to shift and to reduce. This can only happen when there is at least one remaining input symbol.
  2. Reduce/Reduce conflict, which occurs when there is more than one way to reduce. This depends on the grammar rules.
     - In particular, it will occur whenever there are two rules $A \longrightarrow \alpha$ and $B \longrightarrow \beta$ with β a suffix of α.

Dangling Else (I)

Example

$$
\begin{array}{lcl}
\mathit{stmnt} & \longrightarrow & \textbf{if}\ \mathit{bexpr}\ \textbf{then}\ \mathit{stmnt} \\
\mathit{stmnt} & \longrightarrow & \textbf{if}\ \mathit{bexpr}\ \textbf{then}\ \mathit{stmnt}\ \textbf{else}\ \mathit{stmnt} \\
\mathit{stmnt} & \longrightarrow & \ldots \\
\vdots & & \vdots \\
\mathit{bexpr} & \longrightarrow & \ldots \\
\vdots & & \vdots
\end{array}
$$

What should the PDA do in the following configuration?

STACK	INPUT
$\ldots$ **if** *bexpr* **then** *stmnt*	**else** $\ldots$

Dangling Else (II)

Example

- This is a shift/reduce conflict.
- We may impose the common disambiguation strategy by preferring shifting over reducing.

Arrays and Procedures (I)

Example

$$
\begin{array}{lcl}
stmnt & \longrightarrow & \textbf{id}\ (\ parameter\_list\ )\ |\ expr := expr \\
parameter\_list & \longrightarrow & parameter\_list\ ,\ parameter\ |\ parameter \\
parameter & \longrightarrow & \textbf{id} \\
expr & \longrightarrow & \textbf{id}\ (\ expr\_list\ )\ |\ \textbf{id} \\
expr\_list & \longrightarrow & expr\_list\ ,\ expr\ |\ expr
\end{array}
$$

What should the PDA do in the following configuration?

STACK	INPUT
$\ldots$ **id** (**id**	, **id**) $\ldots$

Arrays and Procedures (II)

Example

- Assume the PDA can cheat and detect that shifting is no good.
- Which rule should be used to reduce **id**?
- This is a reduce/reduce conflict.