CSC 3201 Compiler Construction

Department of Computer Science

SZABIST (Islamabad Campus)

Week 4 (Lecture 1, Part 2)

CFG vs CSG

- CFG (Context-Free Grammar):
  - Every production rule is of the form
    - $V \rightarrow w$, where V is a single nonterminal symbol, and w is a string of terminals and/or nonterminals (possibly empty).
  - Nonterminals can be rewritten without regard to the context in which they occur.

CFG vs CSG

- CSG (Context-Sensitive Grammar):
  - Both Left and right hand sides of any production rules may be surrounded by a context of terminal and nonterminal symbols.
  - A production rule is of the form
    - $\beta A \gamma \rightarrow \beta \alpha \gamma$

Parsing Techniques

- Top-down parsers
  - Starts at the root of the parse tree and grows towards leaves.
  - At each node, the parser picks a production and tries to match the input.
  - The parser may pick the wrong production in which case it will need to backtrack.
  - Some grammars are backtrack- free.

Parsing Techniques

- Bottom-up parsers
    - Starts at the leaves and grows toward root of the parse tree.
    - As input is consumed, the parser encodes possibilities in an internal state.
    - Starts in a state valid for legal first tokens.
    - Handle a large class of grammars.

Top-down Parsers

- Construct the root node of the parse tree.
- Repeat until the fringe of the parse tree matches input string.
  - At a node labeled A, select a production with A on its LHS.
  - For each symbol on its RHS, construct the appropriate child.
  - When a terminal symbol is added to the fringe and it does not match the fringe, backtrack.

Backtracking

```
1 goal    → expr
2 expr    → expr + term
3           | expr - term
4           | term
5 term    → term * factor
6           | term / factor
7           | factor
8 factor  → number
9           | id
```

Backtracking

- Example: x-2*y
  - Wrong production selected at step 2

P	Sentential Form	input
-	*Goal*	↑$\underline{x} - 2 * y$
1	*expr*	↑$\underline{x} - \underline{2} * \underline{y}$
2	*expr + term*	↑$x - 2 * y$
4	*term + term*	↑$\underline{x} - 2 * y$
7	*factor + term*	↑$\underline{x} - 2 * y$
9	$\langle id, \underline{x} \rangle + term$	↑$x - 2 * y$
9	$\langle id, \underline{x} \rangle + term$	x ↑$- \underline{2} * \underline{y}$

Backtracking

P	Sentential Form	input
-	*Goal*	↑$\underline{x} - \underline{2} * \underline{y}$
1	*expr*	↑$\underline{x} - \underline{2} * \underline{y}$
3	*expr* − *term*	↑$\underline{x} - \underline{2} * \underline{y}$
4	*term* − *term*	↑$\underline{x} - \underline{2} * \underline{y}$
7	*factor* − *term*	↑$\underline{x} - \underline{2} * \underline{y}$
9	<id,$\underline{x}$> − *term*	↑$\underline{x} - \underline{2} * \underline{y}$
9	<id,$\underline{x}$> − *term*	$\underline{x}$ ↑ − $\underline{2} * \underline{y}$