Context-Free Grammars

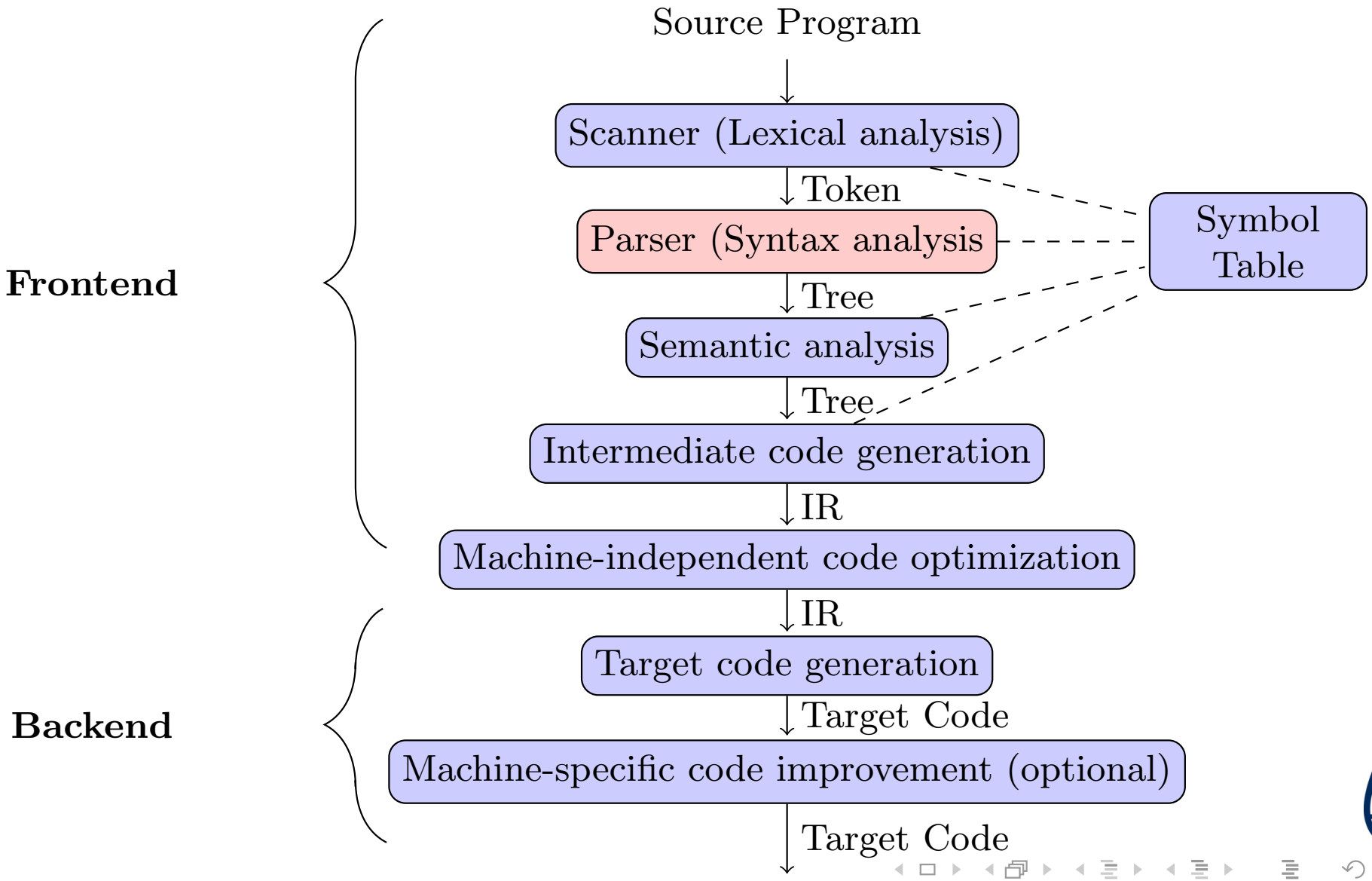
Lecture 4

Sections 4.1 - 4.2

Rokan Uddin Faruqui

Associate Professor
Dept of Computer Science and Engineering
University of Chittaong, Bangladesh
Email: *rokan@cu.ac.bd*

The Phases of Compilation



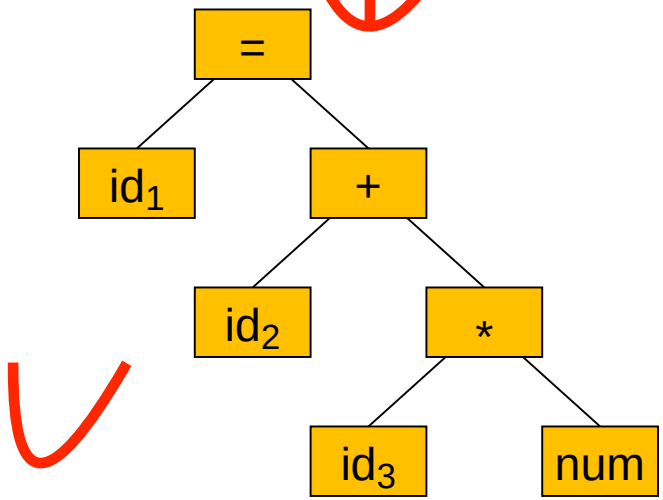
Syntax Analysis

```
position = initial + rate * 60;
```

{id}
“if”

$$\mathbf{id}_1 = \mathbf{id}_2 \; + \; \mathbf{id}_3 \; * \; \mathbf{num} \; ;$$

int if34;



Outline

Outline

How do we specify language syntax?

- Context-Free Grammars,
- uses special notation (BNF – Backus Naur Form),
- consist of set of rules (productions).

Example:

```
if (x=2) print("yep"); else print("nope");
```

Corresponds to a rule:

`stm` $\rightarrow$ `if (expr) stm else stm`

Context-Free Grammars

Definition (Context-free grammar)

A context-free grammar, or CFG, consists of set terminals, a set of nonterminals, a start symbol, and a set of production.

- The terminals are the tokens.
- The nonterminals will eventually be "replaced" by terminals (tokens), matching a grammatical pattern.
- The start symbol is a nonterminal.
- For each production,
  - The left side, or head, is a nonterminal.
  - The right side, body, is any string of terminals and nonterminals, including the empty string.

Example

G=(V,T,S,P)
V={E}
T={+, *, (,), id}
S = E

Example (Context-free grammar)

- Let the terminals be the tokens

$$\{+, *, (,), \mathbf{id}\}.$$

E => E + E => id + E => id + id

- Let the nonterminals be $\{E\}$, which is also the start symbol.
- Let the productions be

1. id * * id
2. id - id /id

id ++ id

$$E \to E + E$$
$$E \to E * E$$
$$E \to (E)$$
$$E \to \mathbf{id}$$

id+id*id
(id + id) *id
(id + id) * (id *id)

Derivations
Parse Tree

Example

Context-free grammars

- We can use the grammar to derive strings of terminals if we
  - Begin with the start symbol.
  - Repeatedly replace a nonterminal with the body of a production of which that nonterminal is the head, until the resulting string consists only of terminals.

Example

Example (Context-free grammar)

- Use the grammar of the example to derive the string

$$\mathbf{(id + id) * id.}$$

Example

Example (Context-free grammar)

- We typically group together all the productions for a single nonterminal.

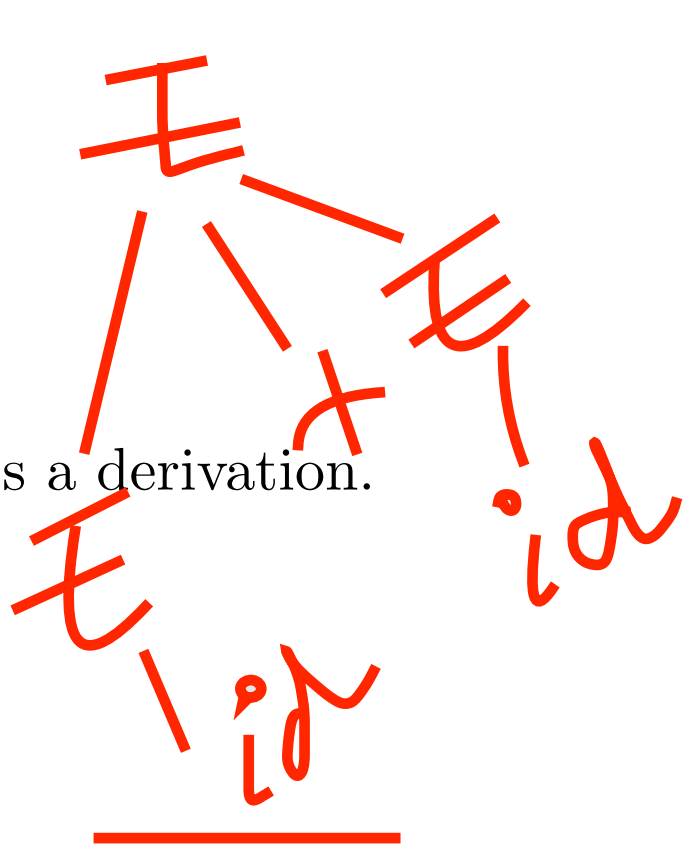
$$E \to E + E \mid E * E \mid (E) \mid \mathbf{id}$$

Outline

Parse Trees

E => E + E = > id + E => id + id

id ++ id



Definition (Parse Tree)

A parse tree is a tree structure that represents a derivation.

- The root node is the start symbol.
- Every interior node is a nonterminal.
- Every leaf node is a terminal.
- The children of each nonterminal node are the nonterminals and terminals of the body of a production for that nonterminal.

Example

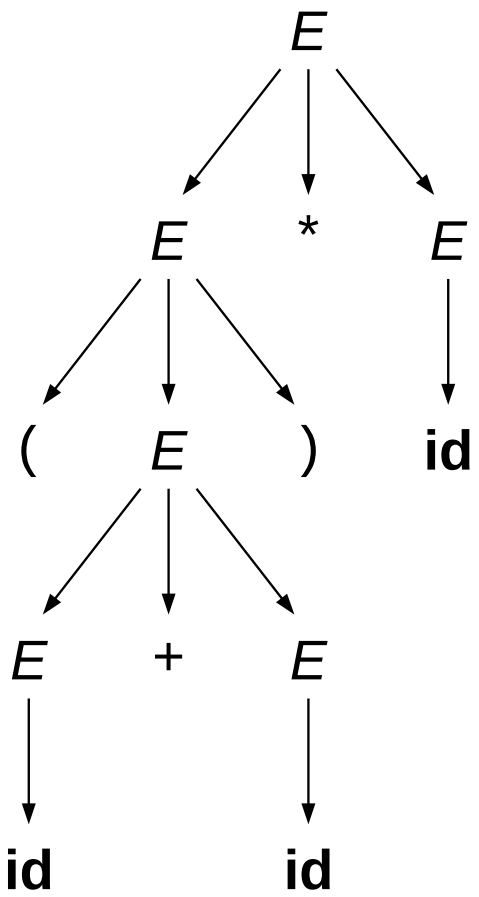
Example (Parse Tree)

- Draw a parse tree for the derivation of the string

$$(\mathbf{id} + \mathbf{id}) * \mathbf{id}.$$

Example

Example (Parse Tree)



(id + id) * id

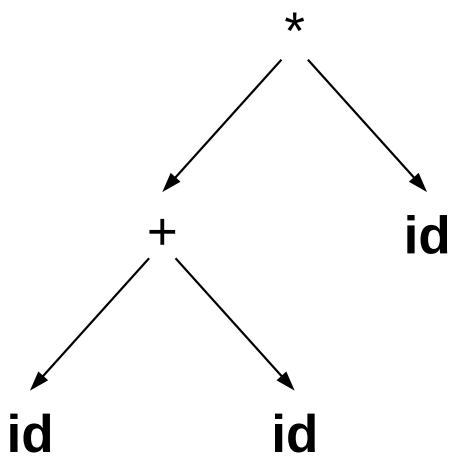
id + id * id

4 + 5 * 3 = 19 / 27

Example

Example (Parse Tree)

The parse tree is not the same as the syntax tree, which we will study later.



Outline

Leftmost and Rightmost Derivations

E => E + E => id + E => id + id

E => E + E => E + id => id +id

Definition (Leftmost derivation)

A leftmost derivation of a string is a derivation in which, at each step, the leftmost nonterminal is replaced with a string.

Definition (Rightmost derivation)

A rightmost derivation of a string is a derivation in which, at each step, the rightmost nonterminal is replaced with a string.

Example

```
E  => E * E
 => (E) * E
=> (E+E) * E
=>(id +E) * E
=> (id + id) * E
=> (id + id) * id
```

Rightmost Derivation

```
E => E * E
  => E * id
 => (E) * id
=>(E +E) * id
=>(E + id) * id
E =>(id + id) * id
```

Example (Leftmost and rightmost derivations)

Using the grammar

$$E \rightarrow E + E \mid E * E \mid (E) \mid \mathbf{id}$$

find leftmost and rightmost derivations of **(id+id)*id**.

```
id * * id
```

```
E => E * E
```

Outline

Ambiguity

- Some grammars provide more than one way to derive a string.
- For example, **id+id*id** can be derived in two different ways using the grammar rules

$$E \rightarrow E \ + \ E \mid E \ * \ E \mid (\ E \ ) \mid \mathbf{id}$$

Ambiguity

Definition (Ambiguous grammar)

A grammar is ambiguous if its language contains a string that has more than one leftmost derivation under that grammar.

Definition (Inherently ambiguous language)

A language is inherently ambiguous if every grammar for that language is ambiguous.

Example

Example (Unambiguous Grammar)

The same language can be derived unambiguously from the following grammar.

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \mathbf{id}$$

Example

Example (Unambiguous Grammar)

Using the grammar

$$E \to E + T \mid T$$
$$T \to T * F \mid F$$
$$F \to (E) \mid \mathbf{id}$$

- Find a leftmost derivation of **id+id*id**.
- Find a leftmost derivation of **id*id+id**.
- Draw the parse trees.