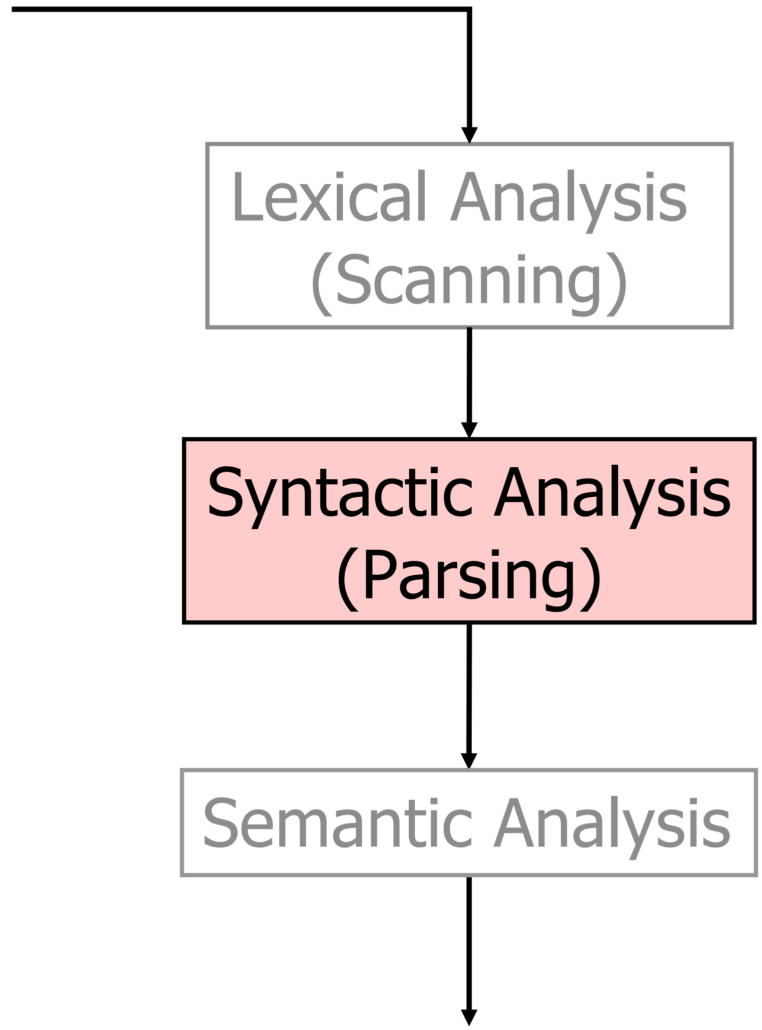
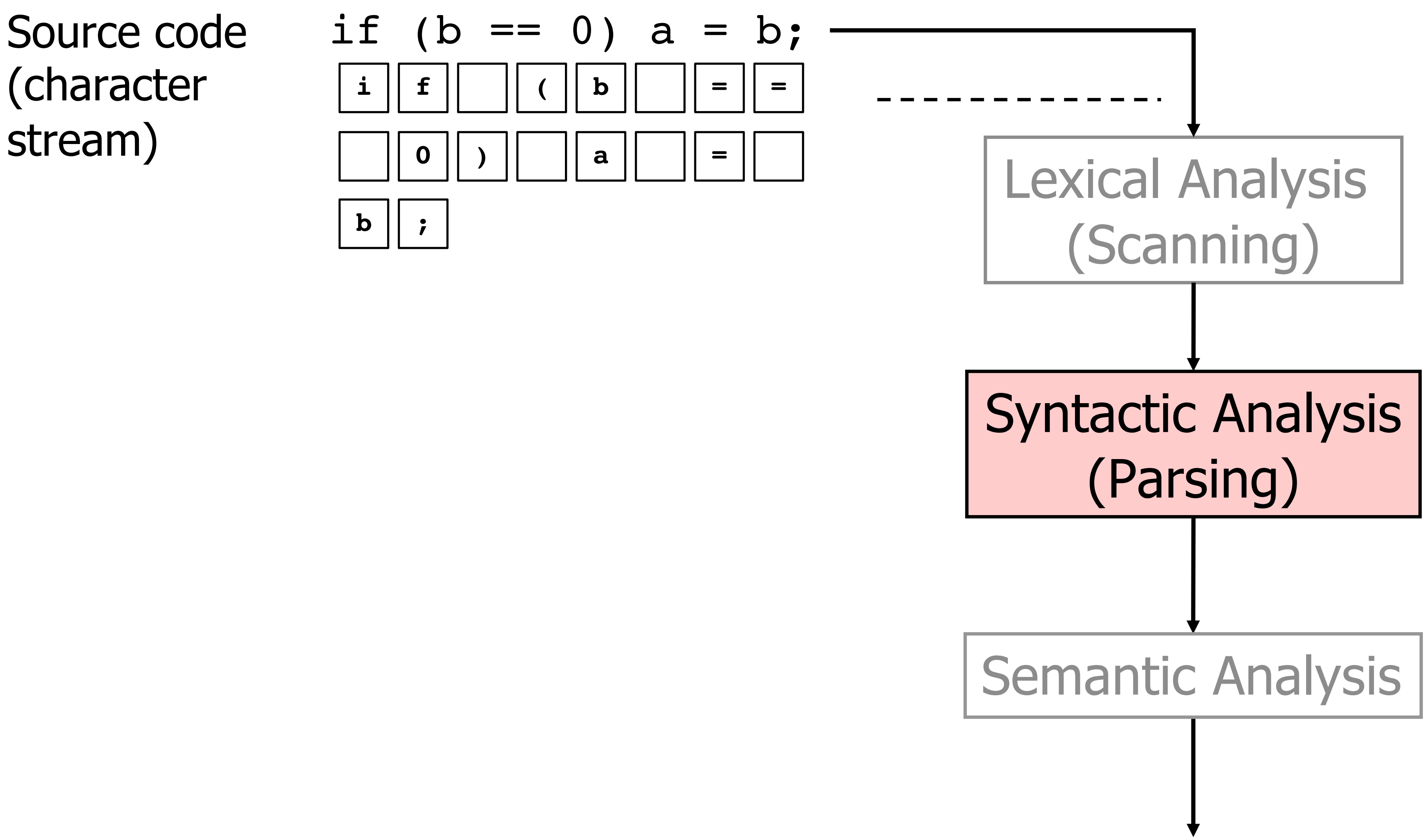
Where We Are

Source code (character stream)

```
if (b == 0) a = b;
```

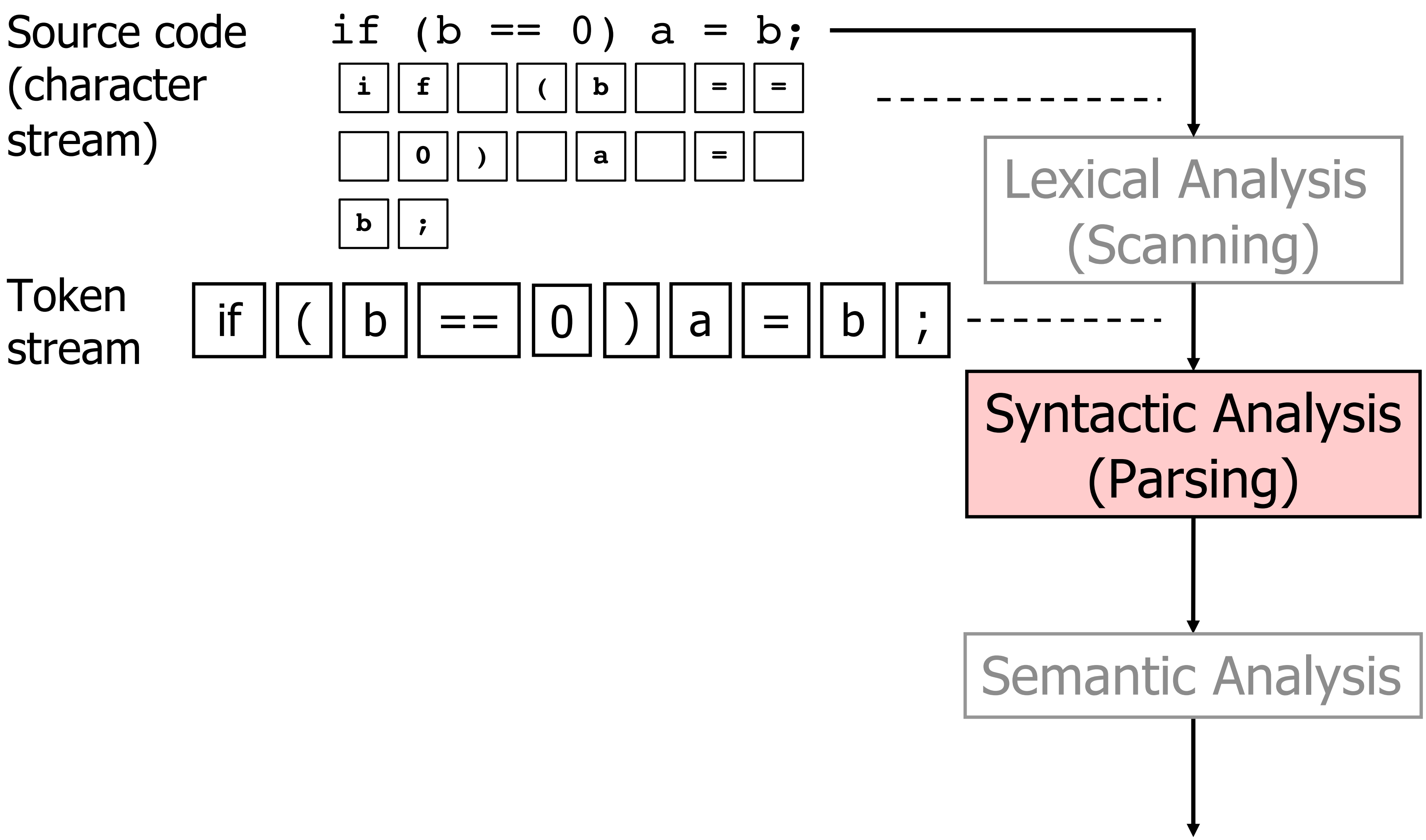


Where We Are

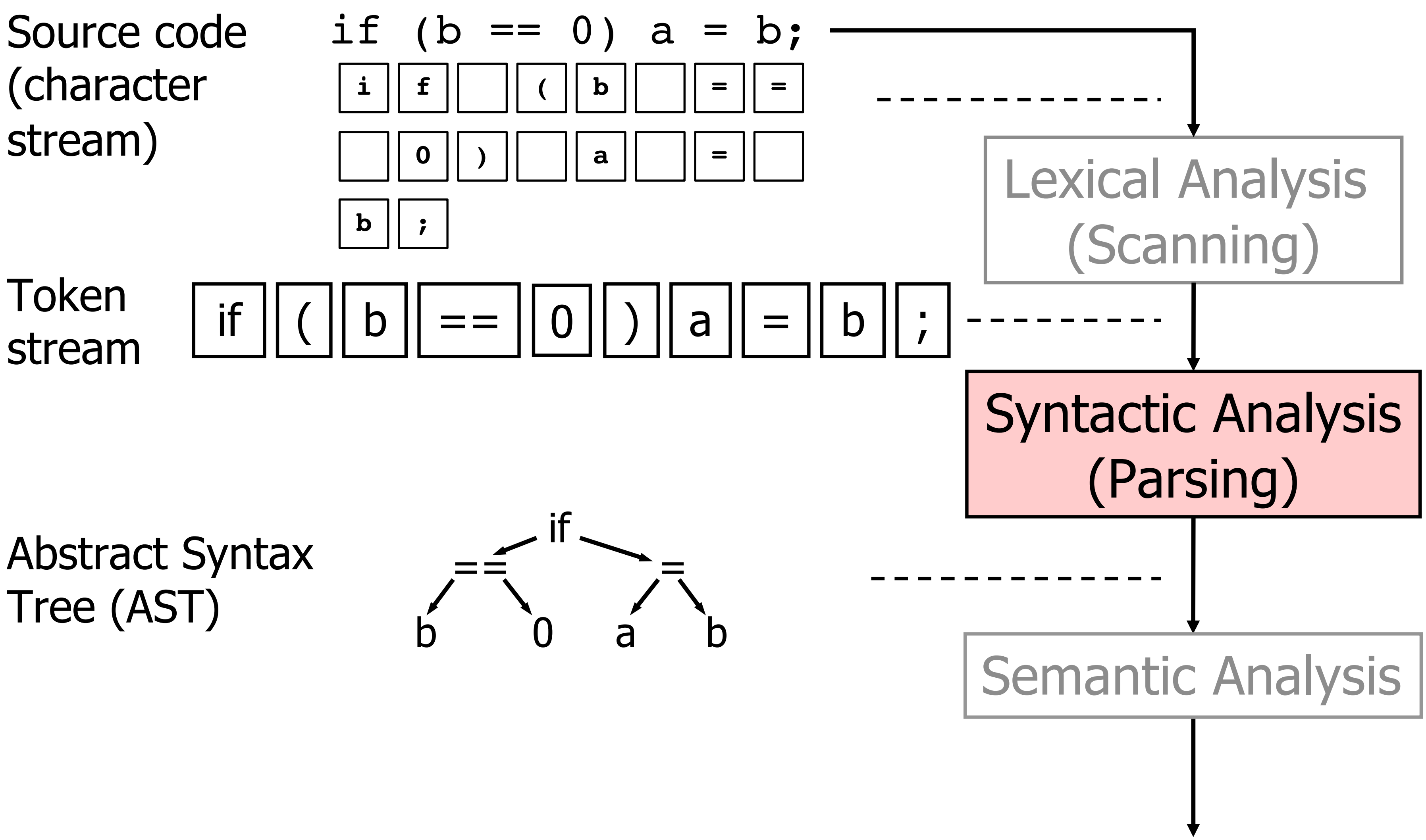


Source code (character stream)
if (b == 0) a = b;
i
f
(
b
=
=
0
)
a
=
b
;
Lexical Analysis (Scanning)
Syntactic Analysis (Parsing)
Semantic Analysis

Where We Are


Source code
(character
stream)
if (b == 0) a = b;
i f (b = =
0) a =
b ;
Token
stream
if (b == 0) a = b ;
Lexical Analysis
(Scanning)
Syntactic Analysis
(Parsing)
Semantic Analysis

Where We Are



Source code (character stream)
if (b == 0) a = b;
i f (b = =
0) a =
b ;
Token stream
if (b == 0) a = b ;
Abstract Syntax Tree (AST)
if
==
=
b 0 a b
Lexical Analysis (Scanning)
Syntactic Analysis (Parsing)
Semantic Analysis

Formalism

- Language vs. grammar
  - Language: A set (generally infinite) of strings over some alphabet.
  - Grammar: A finite generative description of a language.
  - Given a grammar G, $L(G)$ is the language that it generates.

Formalism

- Language vs. grammar
  - Language: A set (generally infinite) of strings over some alphabet.
  - Grammar: A finite generative description of a language.
  - Given a grammar G, $L(G)$ is the language that it generates.
- Context-Free Grammar $G = (N, T, P, S)$, where
  - N is a set of non-terminals;
  - T is a set of terminals (aka tokens);
  - P is a finite set of productions (rewrite rules) of the form $A \rightarrow \alpha$, where $A \in N$ and α is a sentential form;
  - $S \in N$ is the start symbol.

Formalism

- Language vs. grammar
  - Language: A set (generally infinite) of strings over some alphabet.
  - Grammar: A finite generative description of a language.
  - Given a grammar G, $L(G)$ is the language that it generates.
- Context-Free Grammar $G = (N, T, P, S)$, where
  - N is a set of non-terminals;
  - T is a set of terminals (aka tokens);
  - P is a finite set of productions (rewrite rules) of the form $A \to \alpha$, where $A \in N$ and α is a sentential form;
  - $S \in N$ is the start symbol.
- Sentential forms and sentences
  - Sentential form: A string that can be obtained by starting with S and using productions as rewrite rules to rewrite non-terminals.
  - Sentence: A sentential form without non-terminals, i.e., a word in the language $L(G)$.

Recognition vs. Parsing

- Given a grammar G and a sentence s
  - Recognition is a decision problem: $s \in L(G)$?
  - Parsing is a construction problem: Show a derivation (proof) that $s \in L(G)$.

Recognition vs. Parsing

- Given a grammar G and a sentence s
  - Recognition is a decision problem: $s \in L(G)$?
  - Parsing is a construction problem: Show a derivation (proof) that $s \in L(G)$.

- Derivation of string using grammar
  - Start from S and repeatedly re-write one non-terminal at a time using the productions of the grammar, until there are no non-terminals left to re-write.
  - Leftmost/rightmost derivation: A derivation in which the leftmost/rightmost non-terminal of the current sentential form is rewritten at each step.

Example: Simple Expression Grammar

- Consider

  Grammar: $E \rightarrow (E + E) \mid$ **num**

  String: (2 + 3)

Example: Simple Expression Grammar

- Consider

  Grammar: $E \rightarrow (E + E) \mid \textbf{num}$

  String: $(2 + 3)$

- Leftmost derivation
  - $E \Longrightarrow (E + E) \Longrightarrow (2 + E) \Longrightarrow (2 + 3)$

- Rightmost derivation
  - $E \Longrightarrow (E + E) \Longrightarrow (E + 3) \Longrightarrow (2 + 3)$

Ambiguity in Grammars

- Ambiguous grammar
  - A grammar in which there are two or more leftmost derivations for some sentence $s \in L(G)$.
- Consider

  Grammar: E → E + E | E * E | (E) | **num**

- The string 2 + 3 * 5 has two distinct leftmost derivations.
  - $E \Longrightarrow E + E \Longrightarrow 2 + E \Longrightarrow 2 + E * E \Longrightarrow 2 + 3 * E \Longrightarrow 2 + 3 * 5$
  - $E \Longrightarrow E * E \Longrightarrow E + E * E \Longrightarrow 2 + E * E \Longrightarrow 2 + 3 * E \Longrightarrow 2 + 3 * 5$
- However, the strings (2 + 3) * 5 and 2 + (3 * 5) do have unique leftmost derivations.
  - $E \Longrightarrow E * E \Longrightarrow (E) * E \Longrightarrow (E + E) * E \Longrightarrow \cdots \Longrightarrow (2 + 3) * 5$
  - $E \Longrightarrow E + E \Longrightarrow E + (E) \Longrightarrow E + (E * E) \Longrightarrow \cdots \Longrightarrow 2 + (3 * 5)$