

Syntax and Semantics Grammars and Derivations

Defining a language

Natural languages, such as English, are defined by their syntax and semantics.

Syntax – Defines the valid symbols and well-formed sentences.

Semantics: Defines the meaning of symbols and well-formed sentences.

Examples: apple



John loves Lucy



BNF – Backus-Naur Form

BNF: a common method for to specifying the syntax of a programming language

Terminals: The words in our language

Non-terminals: Tokens that are not part of the language, but are part of the grammar

Productions: Rules that define the well-formed sentences in the language.

Example:

- **Nonterminals:** <exp> <term> <factor> <number> <digit>
- **Terminals:** + *) (0 1 2 3 4 5 6 7 8 9
- **Productions:**
 - <exp> ::= <exp> + <term> | <term>
 - <term> ::= <term> * <factor> | <factor>
 - <factor> ::= (<exp>) | <number>
 - <number> ::= <number> <digit> | <digit>
 - <digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Derivations

Start symbol: One of the non-terminal symbols is identified as the start symbol.

Derivation: An application of rules that is such that it consumes a sentence. The derivation begins with the start symbol.

Derivation tree: A graphical manner in which to show a derivation.

Language: All the well-formed formuli which can be derived from the start symbol, by application of the productions.

Example of a Derivation

```
<exp> ::= <exp> + <term> | <term>
<term> ::= <term> * <factor> | <factor>
<factor> ::= ( <exp> ) | <number>
<number> ::= <number> <digit> | <digit>
<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Derive $1 * (2 + 34)$ by developing a derivation tree.

Solve this on your handout

Terminals: $() \lambda . a b c d$

```
<LcExp> ::= <Identifier> |
          (λ <Identifier> . <LcExp>) |
          (<LcExp> <LcExp>)
```

<Identifier> ::= a | b | c | d

$((\lambda a. a) (\lambda b. b))$

Abbreviations: Extended BNF

`{ string }*` (a.k.a. Kleene *) stands for 0 or more occurrences of things derivable from *string*

`{ string }+` (a.k.a. Kleene +) stands for 1 or more occurrences of things derivable from *string*