# CIS 425 - Week 1 - Lecture 1

Alexander Eldemir

1 April 2019

## 1 Syntax vs. Semantics

*Syntax* refers to the structure of the sentences of the language.

*Semantics* refers to the meaning of the sentences of the language.

When talking about PL, a sentence refers to a program.

**Example 1.** - Sentences with the **same** syntax might have **different** meaning:

- Consider the syntactic rule defining a date:

  $$\text{dd / dd / dddd}$$

  where $d$ stands for a digit. The date

  $$01\ /\ 12\ /\ 2018$$

  is syntactically correct. However, it comes with two interpretations: it could either mean December 1, 2018 or January 12, 2018. In other words, the same string can be read as:

  $$\text{day/month/year} \quad \text{or} \quad \text{month/day/year}$$

- Consider the expression 2 + 3 * 9. It could either mean 45 or 29.

- Sentences with **different** syntax might have **same** semantics: Consider the assignment statements

  $$x = a + b \text{ and } x = b + a$$

  they both assign the same value to variable x.

## 2 Notion of a grammar

How do we distinguish the sentences *I like apples* and *like I apples*?

By making use of a grammar. For example, we have the rules learned in school:

(1) sentence $\to$ subject verb object
(2) subject $\to$ *I*
(3) verb $\to$ *like*
(4) object $\to$ *apples*

The above rules state the following: a sentence starts with a subject followed by a verb and an object, a subject is *I*, a verb is the string *like*, and an object refers to *apples*.

We can now show that the string *I like apples* is syntactically correct by building a derivation starting from the root symbol which in this case is "sentence:"

$$\begin{array}{lll} \underline{\text{sentence}} & \to \underline{\text{subject}}\ \text{verb object} & (1) \\ & \to I\ \underline{\text{verb}}\ \text{object} & (2) \\ & \to I\ like\ \underline{\text{object}} & (4) \\ & \to I\ like\ apples & (3) \end{array}$$

In each step we have underlined the symbol we are rewriting and specified the rule we are using.

We can now distinguish our two original sentences by saying that the first is syntactically correct and the second one is not. In trying to build a derivation for the sentence *like I apples* we start with:

$\underline{\text{sentence}} \quad \to \text{subject verb object} \quad (1)$

but then we get stuck since there is no way to derive *like* from subject, *I* from verb, and *apples* from object.

**Definition 2.** *A* grammar *G is a quadruple G = (T, NT, P, R), where:*

- *T is a finite set of terminal symbols;*
- *NT is a finite set of nonterminals symbols;*
- *P is a finite set of productions of the form:*

$$\{(l, r) \mid l \in (T \cup NT)^+, r \in (T \cup NT)^*\}$$

*where each pair $(l, r)$ expresses the fact that $l$ can be rewritten to $r$; $l$ and $r$ are strings over the set of terminals and nonterminals. Note that $l$ cannot be the empty string.*

- *R is for the root symbol and must belong to NT.*

Notation:

- A production $(l, r)$ can be written as: $l \to r$ or $l ::= r$.

- If not specified, the left-hand side of the first production is the start symbol
- Productions with the same left-hand side are usually combined with |
- If a production has an empty right-hand side it means that the right-hand side is the empty string $\epsilon$

**Example 3.** Referring to our example language consisting of a single sentence *I like apples*, we have:
T = { *I, like, apples*}
NT = {sentence, subject, verb, object}
$R$ = sentence
$P$ is the set with the productions $1 - 4$ given above.

**Definition 4.** *The language generated by a grammar G = (T,NT,P,R), written as L(G), is defined as follows*

$$L(G) = \{s \in T^* \mid R \to^+ s\}$$

In other words, the language consists of all the strings that can be derived from the start symbol via one or more rewrite steps. We refer to the sequence of successive rewritings, $R \to^+ s$, as a derivation. For example, we have:

$$\text{sentence} \to^4 \textit{I like apples}$$

since it takes four rewriting steps to derive the string.

## 2.1 Parse Trees

A Parse tree shows how a string is produced by a grammar. In a parse tree:
- The root node is the start symbol
- Every internal node is a nonterminal
- Children of an internal node T, read from left to right, correspond to the right-hand side of a production having T as its left-hand side
- Every leaf node is a terminal

**Example 5.** See Figure **??** for the parse tree associated to the sentence *I like apples.*

## 2.2 Regular Grammar

Let us now impose some restriction on the form of productions.

**Definition 6.** *A* regular grammar *has $l \in NT$ and $r = T$ or $T$ followed by $NT$.*

**Example 7.** The following is an example of a regular grammar:
N ::= 0
N ::= 1
N ::= 0N

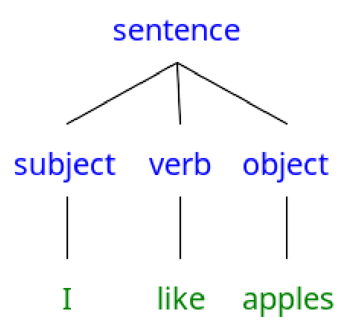


Figure 1: Parse tree