Ex. 1

To account for the syntax of expressions like the ones in (1-a), the following assumptions are made: a noun phrase is either a proper noun or a noun phrase followed by a conjunction phrase; a conjunction phrase is a coordinating word (a comma or the word *and*) followed by a noun phrase.

1. Write the grammar G for noun phrases following these assumptions.
2. Provide the two possible derivation trees that G associates with (1-a). Which of these two analyses seem the most appropriate to you?
3. Is G able to offer an analysis for (2)? If not, propose a grammar G' which can. Give the corresponding syntactic tree.
4. G generates the variants (3) of the expression (1-a). Propose a grammar G'' which generates (1-a), as well as (1-b), but excludes these variants. In other words, G'' would allow at most one occurrence of *and*, before the last conjunct. Note that it is not asked that G'' generates the embedded form (2).

(1) a. Paul, Marc and André
    b. Paul, Marc, Zoé and André

(2) Paul, Marc and Léa, and Luc

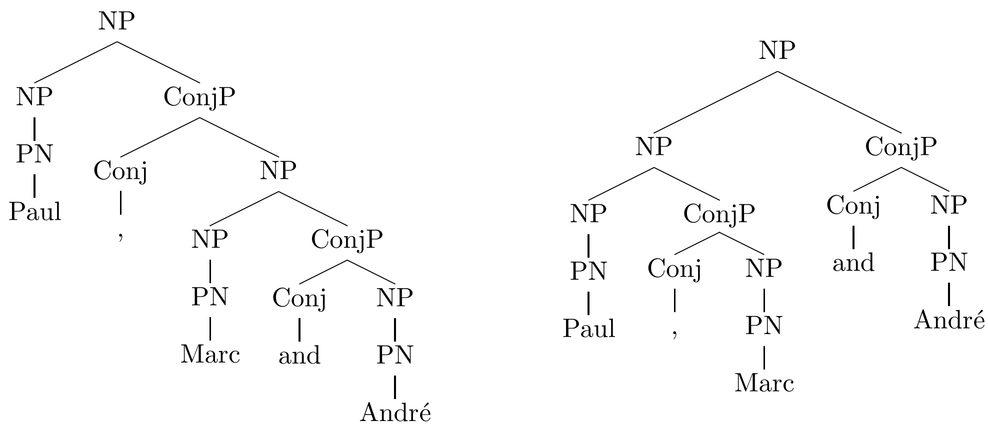
(3) a. Paul, Marc, André
    b. Paul and Marc and André

.... Answer ....

1. The grammar is explicitely defined in the question: $\begin{array}{lll} \text{NP} & \longrightarrow & \text{PN} \\ \text{NP} & \longrightarrow & \text{NP ConjP} \\ \text{ConjP} & \longrightarrow & \text{Conj NP} \end{array}$

   Lexical rules are also needed: $\begin{array}{lll} \text{PN} & \longrightarrow & \text{Paul} \mid \text{Marc} \mid \text{André} \mid \text{Zoé} \mid \text{Luc} \\ \text{Conj} & \longrightarrow & , \mid \text{and} \end{array}$

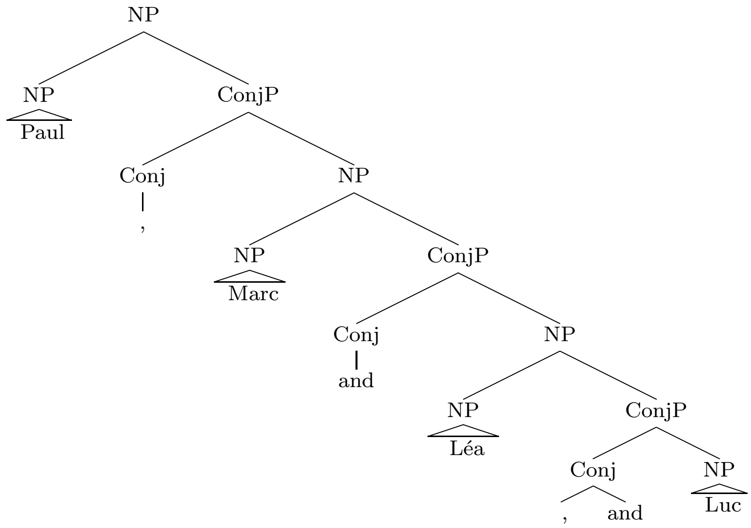
2. The two possible derivation trees are given here:

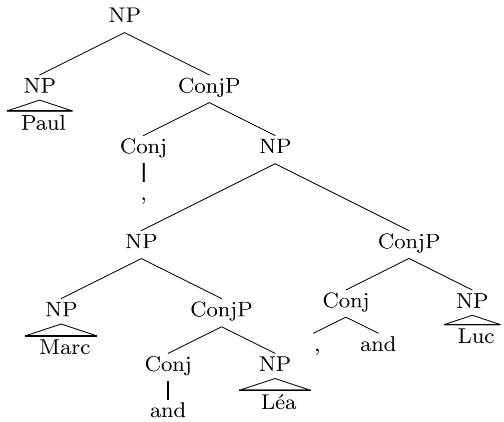


The second analysis yields a grouping between the first two conjoints which doesn't seem justified either semantically nor syntactically; the best analysis is probably the first one.

3. The answer is **no**: two coordination symbols follow one another in the sentence, and this is not allowed by the grammar. A minimal modification would be to alter the lexical rules: Conj $\longrightarrow$, | and | , and

   The grammar is still offering two syntactic analyses:





4. It was reasonable to assume that the grammar had still to be able to generate a single proper noun. Then when two of more proper nouns were generated, it was possible to interpret the question as requiring <u>exactly</u> one occurrence of *and*, or as requiring <u>at most</u> one occurrence (possibly none). The version on the left forces every conjunction to end with exactly one occurrence of *and* ; the version on the right allows for a conjunction with no *and* (but if there is one, there is only one). In both cases 'Pn' is the lexical rule for proper nouns.

$$\begin{array}{lcl} \text{NP} & \longrightarrow & \text{Pn} \\ & | & X \ \textit{and} \ \text{NP} \\ X & \longrightarrow & \text{Pn} \\ & | & \text{Pn} \ , X \end{array} \qquad\qquad \begin{array}{lcl} \text{NP} & \longrightarrow & X \\ & | & X \ \textit{and} \ \text{NP} \\ X & \longrightarrow & \text{Pn} \\ & | & \text{Pn} \ , X \end{array}$$

   Many other versions were possible, depending on the level of lexicalisation, the use of ε-productions, and the choice to depart a lot from the initial version.

Ex. 2

A context-free grammar $G = \langle \Sigma, N, S, P \rangle$ is called *simple* if it verifies the two following conditions:

- $P \subset N \times \Sigma N^*$
- $\forall A \in V, \forall x \in \Sigma, \forall u, u' \in (\Sigma \cup N)^*, (A \to xu) \in P \wedge (A \to xu') \in P \Rightarrow (u = u')$

In words, (1) right hand side parts of the rules start with a terminal letter, followed by an arbitrary number of non-terminal letters (possibly none), and (2) it's not possible to have two differents rules from the same non-terminal whose right hand side part start with the same (terminal) letter.
A context-free language is a *simple language* if there exists a simple grammar that generates it.

1. Find a simple grammar for the language $\{a^n b^{n+1}, n \geq 0\}$
2. Find a simple grammar for the language $\{a^n b^n, n > 0\}$
3. Let L be the language generated by: $S \longrightarrow aSS \mid b$.
   Build a context-free grammar that generates the language Lc^*d.
4. Show that the product of two simple languages is a simple language. Provide a rigourous explanation, not necessarily a mathematical proof.

. . . . Answer . . . .

1. The most natural grammar would be $S \to aSb \mid b$, but it is not simple. Let's introduce a non-terminal symbol whose function will be simply to rewrite into b:
   $S \longrightarrow aSB$
   $S \longrightarrow b$
   $B \longrightarrow b$
2. The most natural grammar would be $S \to aSb \mid \varepsilon$, but it's not simple (none of the two rules is simple). Instead we may want to propose $S \longrightarrow aSB \mid aB$; $B \longrightarrow b$, but even though all of its rules are simple, it's not a simple grammar since two rules from S have the same terminal symbol on the right handside. An additional non-terminal symbol seems necessary:
   $S \longrightarrow aX$
   $X \longrightarrow aXB$
   $X \longrightarrow b$
   $B \longrightarrow b$
3. Quite naturally, the following grammer can be proposed, even though it is not simple (it was not asked): $S_0 \longrightarrow SX$; $S \longrightarrow aSS \mid b$; $X \longrightarrow cX \mid d$.
   However, anticipating the following question, it was also possible to look for a simple grammar (S_0 is the new axiom in both cases):
   $S_0 \longrightarrow aSSX$
   $S_0 \longrightarrow bX$
   $S \longrightarrow aSS$
   $S \longrightarrow b$
   $X \longrightarrow cX$
   $X \longrightarrow d$
4. Let L_1 and L_2 be two simple languages, engendered by two simple grammars G_1 (axiom S_1) and G_2 (axiom S_2). We assume (without loss of generality) that the non-terminal alphabets of the two grammars are distinct.
   Let $S_1 \to u_1 \mid u_2 \mid \ldots \mid u_k$ be the rules starting from S_1 in G_1.
   Then the language L_1L_2 is engendered by a grammar G (axiom S), comprising the rules $S \to u_1S_2 \mid u_2S_2 \mid \ldots \mid u_kS_2$, as well as the rules of G_1 and those of G_2.
   All the rules from G_1 and G_2 have the right form, by hypothesis, and the new rules starting from S also have the right form, since the concatenation of a non-terminal symbol to a rule of the right form remains of the right form.
   In addition, the new rules have exactly the same prefixes that initial rules starting from S_1, among which the factorisation condition was verified by hypothesis. This condition is thus verified for rules starting from S.
   The proposed grammar is therefore simple.
   What remains to be done is to prove that the grammar thus defined engenders L_1L_2.