

Application and Research Highlights

Context-Free Grammars

Lecturer : Giorgio Satta

Programming languages : syntax of programming languages, automatic construction of compilers

Extensible markup language : syntactic specification

Natural language processing : models of natural language syntax

Programming languages follow a context-free, non-regular structure: curly braces in blocks, if and else in C, etc.

CFG is used as a specification in tools for building compilers: for instance Yacc and Flex

XML is a markup language that defines encoding of documents in a format that is both human-readable and machine-readable

Context-free grammar used to describes the allowable HTML tags and the ways in which these tags may be nested (document type definition feature)

CFG for natural language

CFG for a fragment of English; nonterminals such as NP, VP, PP represent **linguistically informed** syntactic categories

S → NP VP

NP → NP PP | Det N | N

VP → VP PP | V NP

PP → P NP

N → chocolate | I | fork | strawberries

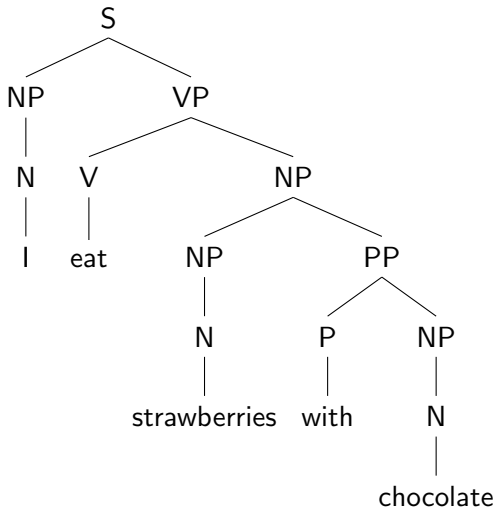
V → eat

Det → a

P → with

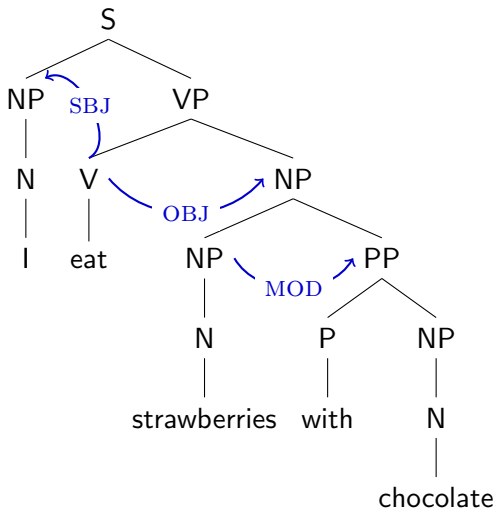
CFG for natural language

Parse tree



CFG for natural language

Underlying **grammatical relations** can be retrieved

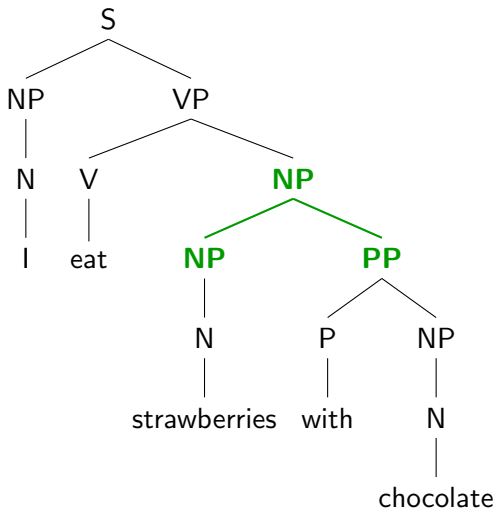


In contrast with programming languages, natural language is **highly ambiguous**

Lexical, **semantic** and **pragmatic** knowledge needed to rule out undesired/unlikely interpretations

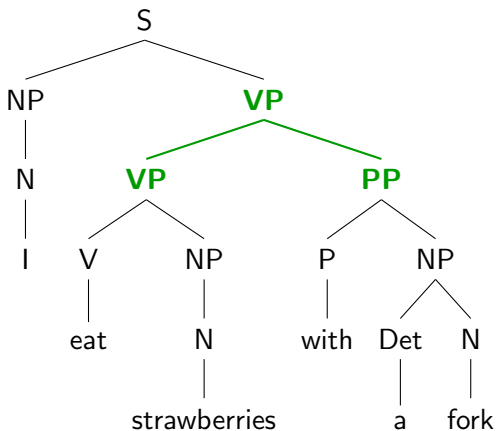
Ambiguity

Correct parse tree



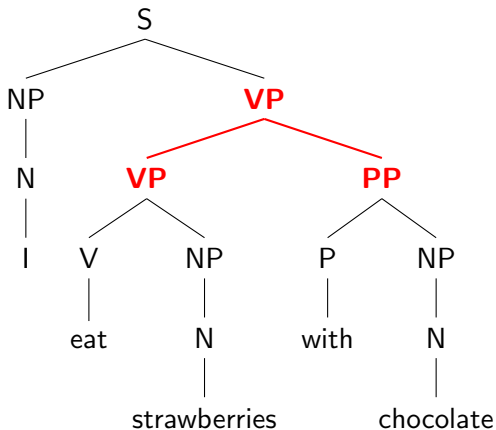
Ambiguity

Correct parse tree



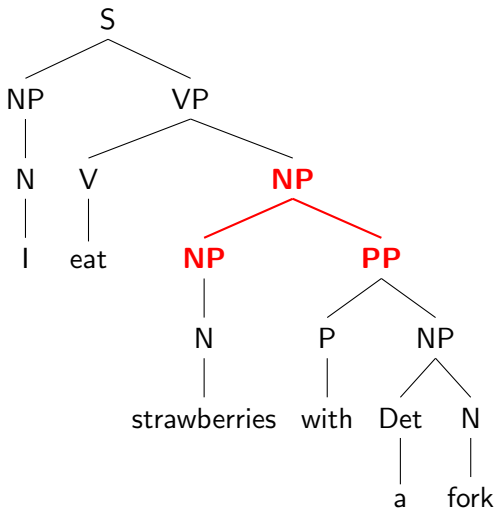
Ambiguity

Wrong parse tree



Ambiguity

Wrong parse tree



CFG productions are associated with **probabilities** that can be estimated by means of

- **supervised** methods: log-likelihood maximization, cross-entropy minimization, convex problem
- **unsupervised** methods: expectation maximization, local maxima