# Natural Language Processing

## Lab 10: Context-Free Grammar

---

## 1 Introduction

In this lab, we will write a context-free grammar for a small fragment of English and test the grammar using a CFG parser from the NLTK toolkit. All files that are needed are available from `/local/kurs/nlp/syntax/`.

## 2 Data

As our fragment of English, we will use the first five sentences from the file `en10-anno.conll`, which we annotated in Lab 8 and parsed in Lab 9:

1. He worked for the BBC for a decade.
2. She spoke to CNN Style about the experience.
3. Global warming has caused a change in the pattern of the rainy seasons.
4. I also wonder whether the Davis Cup played a part.
5. The scheme makes money through sponsorship and advertising.

## 3 Write a Grammar

Our task is to write a context-free grammar that generates the five sentences with reasonable syntactic analyses. The file `grammar.txt` contains a grammar that covers the first sentence. Extending it will involve adding new terminals, new rules and (probably) new nonterminals. Consult Jurafsky and Martin if you are uncertain about what nonterminals and rules to add.

```
# Grammar
S -> S Punct
S -> NP VP
NP -> Pron | Det Noun
VP -> Verb PP PP
PP -> Prep NP
# Lexicon
Noun -> 'BBC' | 'decade'
Verb -> 'worked'
Pron -> 'He'
Det -> 'the' | 'a'
Prep -> 'for'
Punct -> '.'
```

The grammar format is similar to that used in Figure 13.1 on page 462 of Jurafsky and Martin, but note the following:

1. The left-hand side of a rule is separated from the right-hand side by an "arrow" consisting of two characters (`->`), which must be surrounded by spaces.
2. Alternative right-hand sides are separated by a vertical bar (`|`), which must also be surrounded by spaces.
3. Terminal symbols are enclosed in single quotes (`' '`).
4. Comment lines start with the hash symbol (`#`).

## 4 Parse Sentences

To test your grammar, you can use a simple parser, built on top of the NLTK chart parser. In the Python interpreter, you first import the module `cfg_parser`, which allows you to read the grammar from the text file. You can then start the interactive parser, which allows you to type in one sentence at a time and prints out all parse trees that the grammar assigns to the sentence, as shown below.

```
>>> import cfg_parser as cfg
>>> g = cfg.read_grammar('grammar.txt')
>>> cfg.parse_sentences(g)
Parse a sentence (Q to quit): He worked for the BBC for a decade.
(S
  (S
    (NP (Pron He))
    (VP
      (Verb worked)
      (PP (Prep for) (NP (Det the) (Noun BBC)))
      (PP (Prep for) (NP (Det a) (Noun decade)))))
  (Punct .))
Parse a sentence (Q to quit): Q
>>>
```

To make your life easier, you can also run the parser directly in a terminal using a slightly different script:

```
python cfg_parser_b.py
```

The script parses the first sentence. You will need to modify it to parse the other sentences. Note that it produces a tree which is easier to visualize.

## 5 Discuss Ambiguities

It is likely that, once you have extended the grammar to cover all sentences, you will get more than one parse tree for some sentences. Check these analyses and discuss with your classmates whether they all correspond to reasonable interpretations of the sentence. Also discuss whether there are sentences that only receive one analysis but should have several because they are ambiguous.