# Grammars and automatic syntactic analysis

Begoña Villada and Gosse Bouma

May 12, 2003

## Road map

- NLP applications require syntactic analysis
- What is parsing?
- Sentences and constituent structure
- Grammars
- Regular grammars shortcomings
- Center-embedding
- Ambiguity
- Regular vs. context-free languages and grammars
- Corpus query tool based on a context-free grammar

## Applications

Some NLP applications require syntactic analysis of the language.

- **Grammar checkers**
  - ⋆ Jan word ziek
  - ⋆ Dit kind is verwent
- **Dialogue systems**
  - ⋆ USER: wanneer gaat de volgende trein?
  - ⋆ SYSTEM: over vijftien minuten

## Applications 2

- **Machine Translation**
  - Se lo di yo (lit. 'To her it gave I')
  - *To her it gave I vs. *I gave it to her*
  - *Aan haar het geef ik vs. *Ik gaf het aan haar*
- **Machine Translation: Subtitles**
  - They slam the door [in my face]
  - ?* Ze smijten de deur dicht [in m'n gezicht]
  - Ze smijten de deur [voor m'n neus] dicht

## A natural language . . .

- is a *non-finite* set of linguistic expressions
  - *a slow train*, *a very slow train*, *a very very slow train*, . . .
  - *a dog and two cats*, *a dog, two cats and a turtle*, *a dog, two cats, a turtle and a hamster*, . . .
- How to assign a syntactic structure to all possible linguistic expressions in a natural language?

## Automatic Syntactic analysis: Parsing

- *Jan* word *ziek*
- *over vijftien minuten*
- **Recognizing constituents**
  - subject (onderwerp)
  - finite verb (persoonsvorm)
  - direct object (lijdend voorwerp)
  - adverbial modifiers (bijwoorderlijke bepalingen)
- Assigning words the adequate word category

## Grammars

- Collections of rules (and a lexicon) that describe well-formed sentences in a language
- Aims:
  - (automatically) determine whether a sentence is grammatical or not (given the rules of the grammar)
  - assign structure (constituents, meaning) to the sentences of the language

## An example grammar

Grammar rules

- S → NP VP
- NP → Det A N
- VP → V
- VP → V NP

## FSA versus CFG: Recursive rules

- Some languages or syntactic constructions cannot be described by a Finite State Grammar (regular expressions)
- $a^n b^n$: the language consisting of strings containing some number of $a$s followed by the same number of $b$s, where $n$ has non-finite upper bound
  - S → a S b
  - S → a b
- Do such strings exist in real Dutch ?

## Syntactic structure




## Apen blozen

- Alle apen blozen beestachtig (All apes blush terribly)
- Als alle apen blozen, breien boerinnen (when all apes blush, farmers knit)
- Als alle Amerikaanse apen blozen, breien boerinnen borstrokken (when all American apes blush, farmers knit underwear)

## Yet more complex

- Als alle Amerikaanse apen akelig blozen, breien boerinnen blauwe borstrokken (When all American apes blush unpleasantly, farmers knit blue underwear)
- Als alle Amerikaanse apen, alsmede alle Amerikaanse apinnen, altijd akelig blozen, breien Beirese boerinnen, behalve blauwe borstrokken, bepaald beige babysokjes

## Center-embedding

- Ik ken de fotograaf die het kind dat met de pop die rood haar waarin krullen zaten had speelde fotografeerde.
- . . . fotograaf $N_1$ die het kind $N_2$ dat met de pop$N_3$ die rood haar waarin krullen$N_4$ zaten$V_4$ had$V_3$ speelde$V_2$ fotografeerde $V_1$
- literally: I know the photograph who the child who with a doll that red hair where curls sat had played photographed

## Not powerful enough

- Apen blozen sentences pose problems for finite-state transducers
- Similarly: Deze schoenen, boeren, koeien, . . . , zijn respectievelijk goedkoop, moe, woest, . . .

## Center-embedding 2

- Ik ken de fotograaf die het kind fotografeerde.
- Ik ken de fotograaf die het kind dat met de pop speelde fotografeerde.
- Ik ken de fotograaf die het kind dat met de pop die rood haar had speelde fotografeerde.
- Ik ken de fotograaf die het kind dat met de pop die rood haar waarin krullen zaten had speelde fotografeerde.

## Center-embedding 3

- NP → Det N RelPronoun NP V
- np0: det n
- np1: det n rel det n v
- np2: det n rel det n rel det n v v
- npN: det n [ rel det n $]^n$ $v^n$
- This is not a regular language

## Generative power

- Two grammars are equivalent in weak generative power if they accept the same strings
- Two grammars are equivalent in strong generative power if they assign the same constituent-structure to strings
  - CFG and FSA NP-grammars are weakly equivalent, not strongly equivalent.
  - Ambiguity is invisible in FSA

## FSA vs. CFG: Constituent structure

- NP → Det N
- N → A N
- N → N PP
- macro(np,[det,a*,n,[p,det,a*,n]*]).

## Importance of constituent structure

- VP → V NP
- VP → VP PP
  - An eet rijst met een vork
  - An eats rice with a fork
- NP → NP PP
  - An eet rijst met kip
  - An eats rice with chicken
- macro(vp,[v,np,pp*]).

## Constituent structure




## Context-Free Grammar baseNPs

- this grammar includes no separate lexicon
- lexical entries are words in corpus; all words have a POS-tag
-
```
Henk    N(eigen,ev,neut)
Kraaijenhof     N(eigen,ev,neut)
is      V(hulp_of_kopp,ott,3,ev)
nu      Adv(gew,aanw)
haar    Pron(bez,3,ev,neut,attr)
trainer N(soort,ev,neut)
```

## CFG application

- Digging up complex prepositions from a corpus
  - in tegenstelling tot ('in constrast with')
  - op jacht naar ('in search of')
- Given
  - a part-of-speech tagged corpus
  - syntactic structure: Prep baseNP Prep
- GSEARCH a corpus query tool developed at the University of Edinburgh

## Context-Free Grammar baseNPs 2

**Grammar rules** and **lexical categories**

-
```
bnp  --> dp ap* noun
bnp  --> ap* noun
dp   --> det
dp   --> poss
ap   --> adv adj
ap   --> adj
```
-
```
noun --> <N\(.*>           % common, proper
det  --> <Art.*>           % determiner
poss --> <Pron\(.*attr\)>  % possesive det
adj  --> <Adj\(attr.*>     % attr adjective
adv  --> <Adv.*>           % adverb
```

## Extract Prep baseNP Prep strings

- `gsearch VOLKS-jan GrammarWotan -p0 -nc -M prep bnp prep > pbnpp`

- `gsearch VOLKS-jan - '<tag=Prep.* & word=op>' '<tag=N.*>' '<tag=Prep.*>' > prep_op_prep`