Mandatory assignment 4, INF2820, 2013

- **Deadline: April 25, 6:00 pm**
- **To be delivered in devilry.**
- **You should put your username in the comments of the submitted files, like**
  `# Oblig 5, jtl`
- **The mentioned files can be found at /projects/nlp/inf2820/**

Exercise 1 – Parsing algorithms and efficiency (10 points)

In obligatory assignment 3 we saw how inefficient the original sr_parser() is. We made an improved version sr_parse_refined and saw how much more efficient it was by comparing

```
>>> timing("""sr_parse(grammar,("Mary saw a man"+n*" in the park").split())""")
```

for n =1, 2, …, 5 for the original and the refined version of the parser. But we also saw that the refined version slowed down if we continued with n=6, 7, 8, 9

We will now study what is gained by using some sort of table parser, e.g. chart parsing. Start with sr_parse_refined, and in particular with the same example grammar. Make a chart parser, cp, for the grammar using nltk.ChartParser. Then test

```
>>> timing("""cp.nbest_parse(("Mary saw a man"+n*" in the park").split())""")
```

With n=6,7,8,9. How does this compare to the shift-reduce parser? Then test the cp with n=10, 20, 30, 40, 50. What do you see?

Submission: Answers to the questions. A 2*4 table comparing the times of the two parsers for n=6, 7, 8, 9. A table for the times of cp for n=10, 20, 30, 40, 50.

Exercise 2 – Subsumption and unification (20 points)

a) Consider the five feature structures at the next page. For each of the 10 pair of structures A, B: does A subsume B? Does B subsume A?

b) For each pair A, B, are A, B unifiable? For the pairs that are unifiable, give the unification both as an attribute-value matrix (AVM) and as a directed acyclic graph (DAG).

Submission: Answers to the questions. The structures in question (b). You do not have to spend time on fancy type-setting. The best preparation for the exam is to solve (b) by pen and paper. You may then scan or take a digital photo of your solution and deliver this.

Exercise 3 – Understanding unification grammar (20 points)

We will work on the grammar v.2.1 from the lecture. For the simple sentence “John slept” it yields the analysis in fig. 1. In a simplified notation this corresponds to the tree (fig. 2). As a grammatical analysis one should be aware about certain choices that have been made, in particular the strict X-bar schema:

1. A sentence is considered to be a maximal projection of the verb, i.e. of category V’’.
2. To follow a strict X-bar approach, there will be several unary (non-branching) trees for phrases without specifiers or complements.
3. The treatment of determiners is simplified

$$1)\ \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix}\text{PERS} & \text{3rd}\\ \text{NUM} & \text{sg}\end{bmatrix} \\ \text{CAT} & \text{n}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{BAR} & 0 \\ \text{SPEC} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{CAT} & \text{det}\end{bmatrix}\\ \text{VAL} & \begin{bmatrix}\text{SPEC} & \text{-}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

$$2)\ \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix}\text{PERS} & \text{3rd}\\ \text{NUM} & \text{sg}\end{bmatrix} \\ \text{CAT} & \text{n}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{BAR} & 1 \\ \text{SPEC} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{CAT} & \text{det}\end{bmatrix}\\ \text{VAL} & \begin{bmatrix}\text{SPEC} & \text{-}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

$$3)\ \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{3} \\ \text{CAT} & \text{n}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{BAR} & 0 \\ \text{SPEC} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{CAT} & \text{det}\\ \text{AGR} & \boxed{3}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

$$4)\ \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CAT} & \text{det} \\ \text{AGR} & \begin{bmatrix}\text{PERS} & \text{3rd}\\ \text{NUM} & \text{sg}\end{bmatrix}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{BAR} & 1 \\ \text{SPEC} & \text{-}\end{bmatrix}\end{bmatrix}$$

$$5)\ \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{BAR} & 0\end{bmatrix}\end{bmatrix}$$

(For completeness:

- To identify S with V’’ has been proposed in the (early) X-bar literature, although most newer proposals have chosen a more complex analysis. One reason is the analysis of inflection and auxiliaries.
- In other approaches the unary trees may be avoided. For example, in HPSG where there is no explicit BAR feature, where instead the BAR-level is determined by whether the COMPS and SPEC-lists are empty.)

We have implemented grammar v.2.1 in the NLTK fcfg-format in the file fcfg/grammar_2_1.fcfg. The result of analyzing “John slept” is

```
>>> slept= mod.nbest_parse("John slept".split())
>>> len(slept)
1
>>> print slept[0]
(X[FS=[HEAD=[CAT='V'], VAL=[BAR=2, -MOD]]]
  (X[FS=[HEAD=[CAT='N'], VAL=[BAR=2, -MOD]]]
    (X[FS=[HEAD=[CAT='N'], VAL=[BAR=1, -MOD, -SPEC]]]
      (X[FS=[HEAD=[CAT='N'], VAL=[BAR=0, -COMP1, -MOD, -SPEC]]] John)))
  (X[FS=[HEAD=[CAT='V'], VAL=[BAR=1, -MOD, SPEC=[HEAD=[CAT='N'],
VAL=[BAR=2]]]]]
    (X[FS=[HEAD=[CAT='V'], VAL=[BAR=0, -COMP1, -MOD, SPEC=[HEAD=[CAT='N'],
VAL=[BAR=2]]]]]
      slept)))
```

a) For each of the numbered nodes in (fig. 1): which rule is responsible for the construction of this node from its daughters?

b) We will see how the grammar describes some other constructions. Consider the sentence

3) A small girl saw John

This sentence is covered by the grammar. The simplified analysis is given by the tree (fig. 3). We want the full analysis which corresponds to (fig.3) as (fig.1) corresponds to (fig.2). To save you some repetitive work, you only have to draw the structures corresponding to the nodes within the red boundary.

Submission: Answers to the questions. The structures in question (b). Also here you may use pen and paper and take a picture or scan.

Some useful tools:

Before we proceed we will introduce some useful tools.

- First repeat the class Tree from Exercise set 6, and in particular the method <name_of_tree>.draw() which may be used when you want to inspect your analyses. (It is also possible to save the tree to postscript from which you may include it in your delivery if you find it convenient.)
- Another tool which we have not used so far : It is possible to draw several trees at the same time if you like to compare them by

  ```
  >>> from nltk.draw.tree import draw_trees
  >>> draw_trees(t1, t2, t3)
  ```

  where t1, t2 and t3 are trees. The function takes any number of trees as arguments. If you have a list of trees, say *ts* you may pass them as arguments by

  ```
  >>> draw_trees(*ts)
  ```
- Unfortunately even though the tree drawing methods work for feature structure trees, there seems to be some problems with the vertical alignment and the results are hard to read.
- I have written a small function which converts a tree of the form (fig.1) to form (fig.2), and a version that rewrites a list of trees. Combining this with draw_trees will give you a quick indication of whether your analyses are on the right track. These tools are in the file fcfg/wrap.py. For example, figure 1 is produced by

  ```
  >>> cp = load_parser("file:grammar_2_1.fcfg")
  >>> ts = cp.nbest_parse("John slept".split())
  >>> tn = simple_trees(ts)
  >>> draw_trees(*tn)
  ```

Exercise 4 – Extending a unification grammar (50 points)

a) We will now extend the fragment in grammar_2_1.fcfg. First we will include some more verbs, and start with the analysis of

4) A small girl told that John slept

a long the lines of fig.4. We should not add new rules, only lexical items. Two entries are needed:

- The verb *told* should take a complement of type C'' (=CP, complement phrase)
- The complementizer *that* should be of category C and take a V'' as complement.

Complete the details and check that you get the same analysis for the sentence.

b) Going back to earlier obligs, we also want to include some more verbs and constructions like (* means ungrammatical, should not be in the grammar)

4b)Mary told a girl that John slept
4c) Mary said that John slept
4d) *Mary said a girl that John slept

c) We will also include a construction like the one in

5) John gave a toy to Mary

We will analyze it as in the tree in fig. 5. To do this we will have two different entries in the lexicon for verbs like *give*. (In a more elaborate model we would have rules relating the two.) Make a new lexical entry for *give* and for *to* to get the analysis.

d) There is one problem. We will also include other prepositions like *by, in, with, of,…* but avoid constructions like

> 5b) John gave a toy by Mary

One possibility is to introduce a new head-feature for each preposition which distinguishes the prepositions from each other. We may call it FORM and let FORM='to' for *to*, FORM='by' for *by* etc. And then let *give* only take a PP as COMP2 if it has the form 'to'. Fill in the details.

e) We also want PPs as modifiers to nouns as in (cf. fig. 6)

> 6) A girl with a toy slept.

Look at how adjectives as modifiers are analyzed and make the MOD feature on PPs appropriately.

f) We also want PPs as modifiers of V'. Ideally we would have the same lexical entry for a preposition whether the PP is an argument (COMP) of a verb, it modifies an N' or it modifies a V'. It is not easy with the machinery we have introduced so far. The easiest is to have two different lexical entries, one where the PP modifies an N' and another one where it modifies a V'. This solution suffices for this exercise. There is a possible danger that you will get a spurious ambiguity in the analysis of sentence (5), but we ignore that.

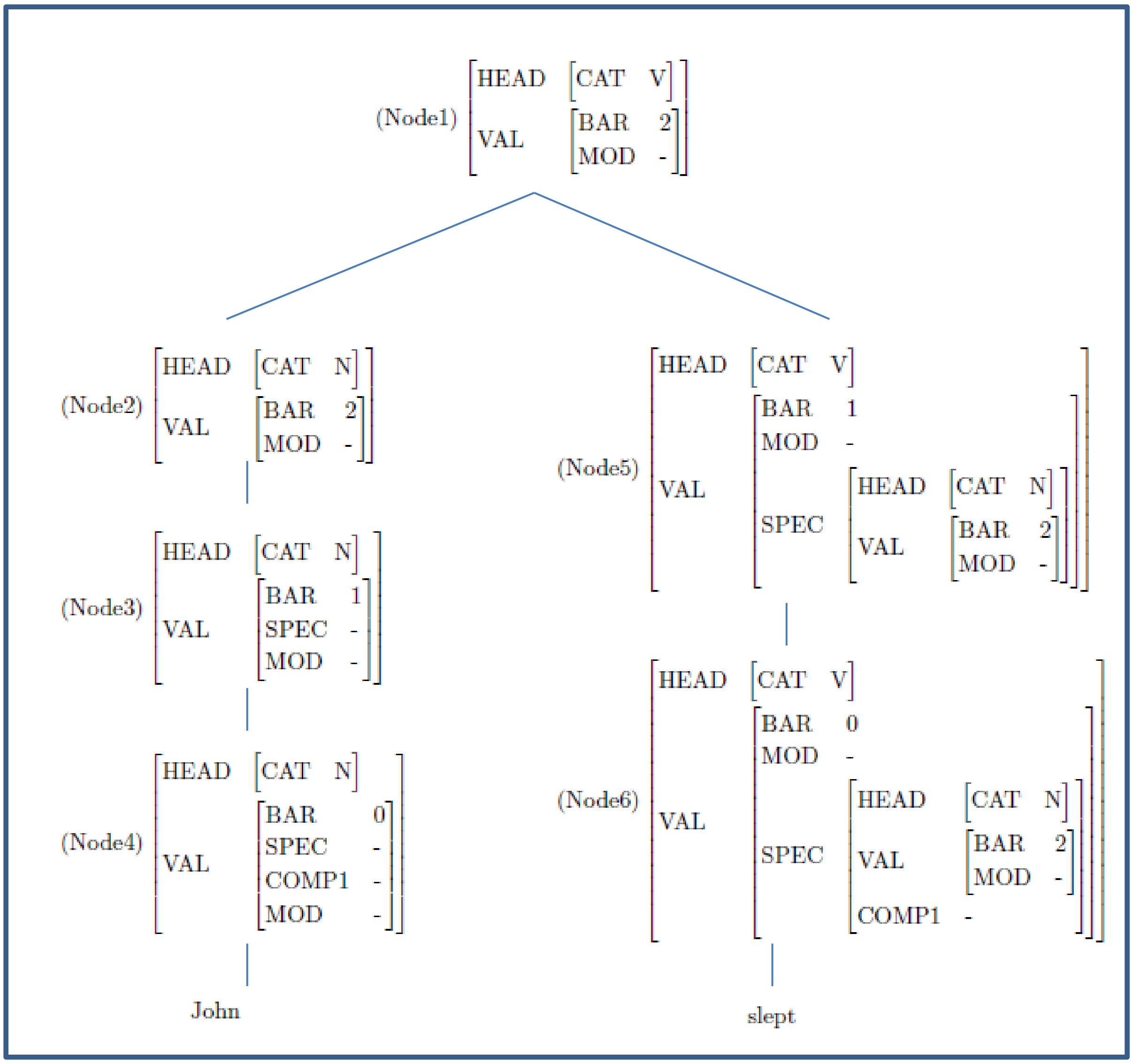
g) Finally a reasonable analysis of sentences like

> 7a) John went home.
> 7b) John saw Mary here.

is to consider *home* and *here* to be an intransitive preposition which makes up a full PP by itself. Fill in the details.

Submission: The extended grammar. At some points you may find the exercise underspecified. Make your own choices for how you will solve them and explain your choices.

Fig.1


(Node1) HEAD [CAT V] VAL [BAR 2, MOD -]
(Node2) HEAD [CAT N] VAL [BAR 2, MOD -]
(Node3) HEAD [CAT N] VAL [BAR 1, SPEC -, MOD -]
(Node4) HEAD [CAT N] VAL [BAR 0, SPEC -, COMP1 -, MOD -]
John
(Node5) HEAD [CAT V] VAL [BAR 1, MOD -, SPEC [HEAD [CAT N] VAL [BAR 2, MOD -]]]
(Node6) HEAD [CAT V] VAL [BAR 0, MOD -, SPEC [HEAD [CAT N] VAL [BAR 2, MOD -]] COMP1 -]
slept

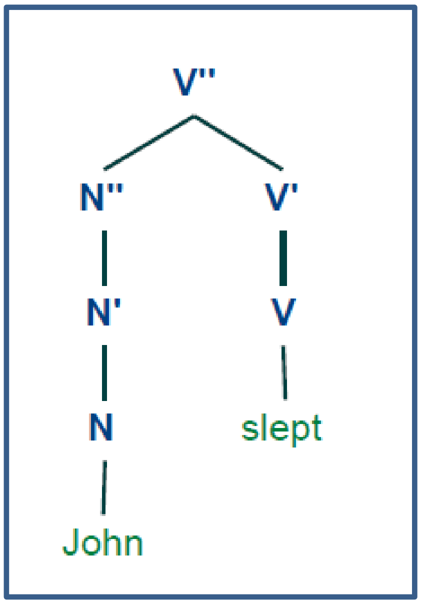


Fig.2

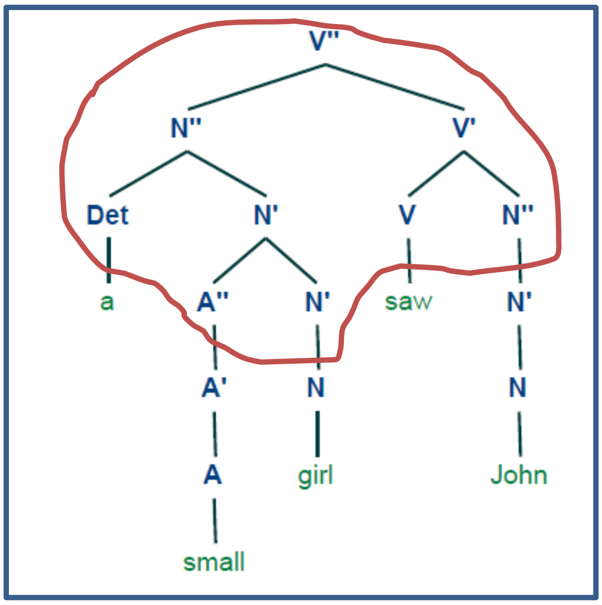


Fig.3

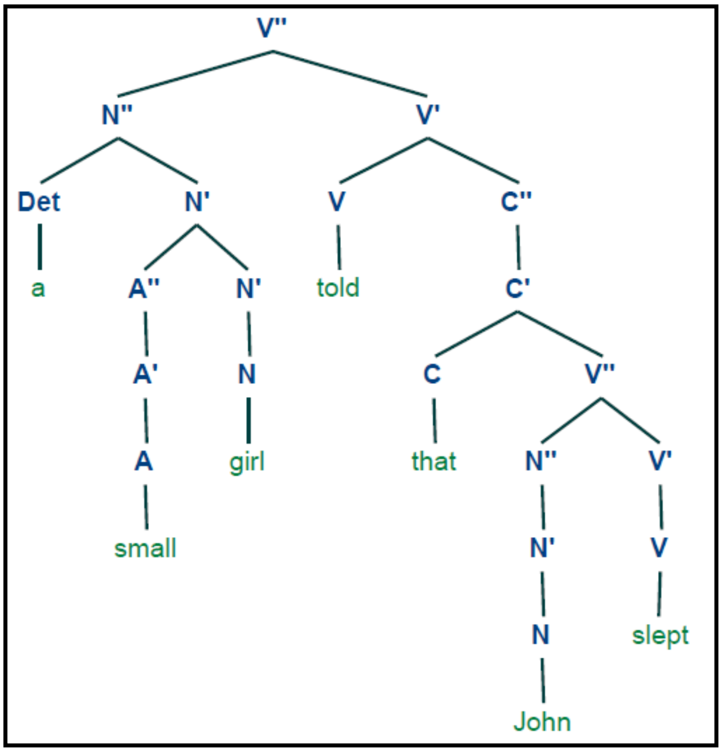
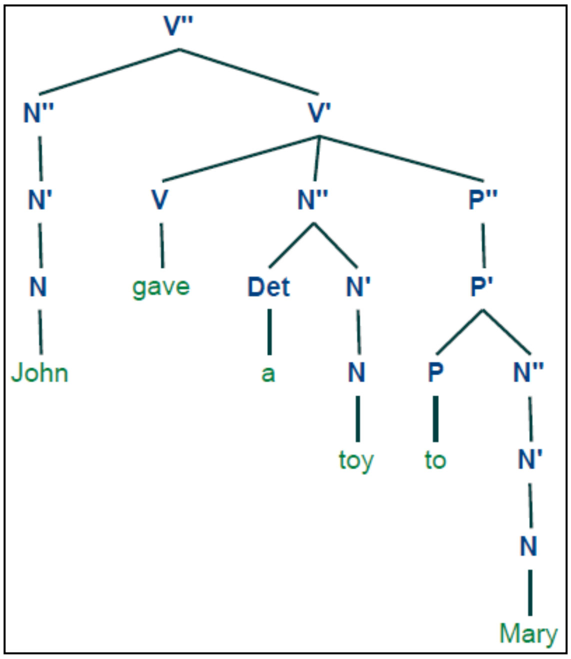


Fig.4



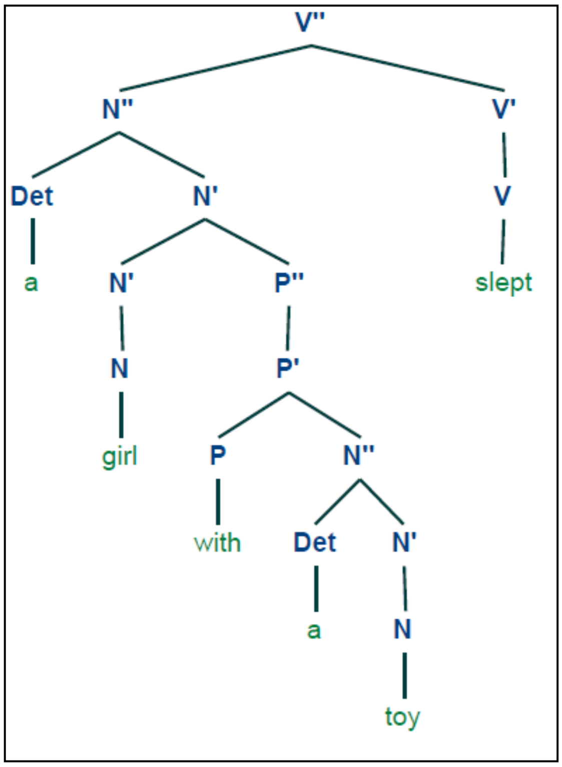


Fig.5

Fig.6