Formal Syntax & Grammar Engineering (Optional Exercise)

High-Level Goals

- Act as an autonomous researcher in formal syntax.
- Investigate the structure of the Swedish noun phrase.
- Use an LKB implementation to validate your analysis.

1 Obtaining the Starting Grammar

In this exercise, we will use the English grammar (*sans* semantics) that we had produced upon completion of Exercise # 4 as our starting point and adapt it to Swedish. Even though there are many interesting differences between the two languages, specifically in clause structure (i.e. the way verbal heads and their dependents are realized), we will focus in this exercise on NP-internal syntax and, hence, hope to get a jump start from using a grammar of English. This strategy is often (and euphemistically) called 'grammar porting' in the formal grammar engineering literature.

(a) To set yourself up for work on this exercise, make sure you are content with your solution to Exercise # 4, or obtain a copy of our model solution from the course web page. Assuming you have a good grammar of English in a directory '`grammar4`', copy it into a new directory '`swedish`' and then work there:

```
cd ~
cp -pr grammar4 swedish
```

(b) Next, within the new '`swedish`' directory, obtain a few new files we supply for this exercise. While most of the files of the English grammar will require editing as part of the adaptation to Swedish, the original English test data will not do us much good. Obtain two sets of test items from the course web page, viz. '`base.items`' for the first part of this exercise and '`kompleks.items`' for the optional second part. Put both files into your '`swedish`' directory, and when you use the LKB batch parsing machinery, choose corresponding file names with the '`.results`' suffix to record the system results.

As a general strategy, use our test items as the development target for your grammar, i.e. aim to provide analyses for all the wellformed examples and have your grammar reject any of the ungrammatical inputs.

(c) As always, make generous use of the comment syntax in TDL to document the files of your grammar. Additionally, for this exercise, we ask you to prepare a brief discussion document (in the spirit of a scientific abstract submitted to a linguistics workshop) commenting on the relevant syntactic aspects of the Swedish NP and the choices made in your analysis. Aim for between half a page and a full page of text.

2 Some Pencil-and-Paper Linguistics

- Unlike some grammar engineers we know, you would know better than start with the actual implementation before you had obtained a good theoretical understanding of the data you want to analyze with your grammar. Take a look at the '`base.items`' file and obtain an intuitive idea of what we are trying to accomplish. What appear to be the relevant dimensions of variation within the Swedish noun phrase, and which of them have correspondences in our English grammar already? What is different in the Swedish NP, when contrasted with our good understanding of the English noun phrase?

(a) In the light of the above, name the three morphological categories that appear to play a role in agreement relations among members of the Swedish NP. For each of these categories, identify the range of possible values.

(b) Sketch at least two constituent trees for (grammatical) examples from '`base.items`', for example *en snäll katt hoppar* and *katter hoppar* would seem like good analytical test cases.

In a sentence or two, discuss what is novel about NPs like *katter*. Assuming we want to avoid making fundamental changes to our analysis of modification, what could be the consequences for the NP-internal structure in this case? Try thinking in terms of the three projection levels we have assumed: lexical, intermediate, and maximal.

3 The Actual Exercise

- In a step-wise fashion, adapt the English grammar to turn into a grammar of a real (if somewhat funny) language. For example, to encode the richer inventory of morphological categories, decide on the range of agreement features and corresponding types, or maybe consider an application of the technique we used for English, i.e. accounting for all morphological variation in a single (though likely fairly complex) type hierarchy. Both options should work well.

  With the necessary set of distinctions you are expecting to make at hand, maybe next focus on the structure of the lexicon and the inflectional rule machinery (we will neither need the dative shift nor agentive nominalization rules in the grammar of Swedish). Make sure you have all of the vocabulary from our test data available and, if you are enjoying yourself, maybe add a few more. You will likely produce a few more inflectional rules than we saw for English (Swedish appears to have a slightly richer morphology, though it is not quite German or Russian); keep the '`%suffix`' annotations on those rules simple, i.e. do not necessarily attempt to cover all possible inflected forms for all words of the language.

- As you had observed in part (1b) above, the determiner – noun relation in Swedish, can take several forms. Try to make your analysis of forms like *katterna* (plural) and *katten* (singular) similar in terms of the tree structure.

- At this point, give some thought to what goes on in examples like **snälla katten*? Look for a way of enabling the modifer (aka non-head daughter) to contribute to the category of the mother constituent (so that you get control over what can happen further up in its derivation tree).

- *A note of caution:* Even in the 21^{th} century, languages that use scripts exceeding the original (1960s or so) 7-bit ASCII range often present challenges to computer software, even more so when multiple operating systems and philosophies (MacOS and Solaris, in our case) are involved. The LKB itself is UniCode-enabled and capable of dealing with languages as exoctic as Norwegian, Japanese, and Korean. However, if you find that among our ensemble of tools—the MacOS X Server, emacs(1), and the LKB—you encounter problems relating to the accented characters in our test data, please feel free to (i) be frustrated with the state of the computational universe and (ii) replace the adjective *snäll* with another adjective that (preferably) inflects the same way but avoids accented characters. Or consider writing a grammar of Norwegian instead, where we would say *snill* instead.

4 A Finer-Grained Classification of Nouns (Optional)

- Find the discussion(s) of the mass vs. count distinction for nouns in Sag, Wasow, & Bender (2003). Look at the additional data in our '`kompleks.items`' test file and adapt your grammar to draw the additional distinction and account for the grammatical contrasts presented in your data.

Submit your results in email to Stephan and Lilja by 12:00 h on Friday, December 24.