Practical Grammar Engineering using HPSG
Exercises day 2

Goals:

1. Getting familiar with types and features in the Matrix grammar.
2. Introduce language specific lexeme, word and phrase types.
3. Parse short sentences.

Exercises:

1. Download the Grammar Matrix
   (a) In the X-terminal where you have started LKB, type `cvs checkout matrix0.6` to retrieve the Matrix.
   (b) The "matrix.tdl" file in the "matrix0.6" directory contains a number of language independent types which can be used in constructing language specific lexical entries and rules. In this exercise we will add some language specific types in the file "my_language.tdl" which will inherit from types in the "matrix.tdl" file and which will be supertypes of language specific lexical entries and rules.
2. Start the LKB and load the Grammar Matrix
   (a) Start emacs
   (b) In emacs: `<ESC>`x lkb
   (c) The LKB top window should appear.
   (d) Load the grammar by selecting Load/Complete Grammar in the "LKB Top" window, then doubleclicking on the directory "matrix0.6", then on "lkb" and then on the file "script".
3. Personalize your copy of the Matrix
   (a) Rename the file "my_language".tdl to reflect the language you are working on, e.g., english.tdl.
   (b) Consider opening it in emacs and adding some comments to the beginning. The comment character is ;.
   (c) Open the file lkb/script and replace "my_language.tdl" with your new file name. (It only occurs twice, once in a comment (;;;) that you don't need to change, and once in the code.)
   (d) Save the changes to script
4. Add language-specific information.
   Note The default place to make changes is your renamed my_language.tdl file. In principle, you should not need to make any changes to matrix.tdl. You may need to use some other files (e.g., for lexical entry and rule instances). In the following instructions, if no particular file is mentioned, use your renamed my_language.tdl file.
   (a) Add some subtypes of the type head for nouns, verbs, and determiners:

```
noun := head.
verb := head.
det := head.
```

   (For some languages, the type *det* might not be necessary. For others, you might want to add a type postposition or some sort. (For example, one analysis of Japanese case particles is that they are all in fact postpositions. Thus vanilla transitive verbs don't take NP arguments, but rather PP arguments.))
   (b) Now introduce a subtype of *lex-item* (as you're going to use a determiner), as follows: