HW #2

LING 571
Deep Processing Techniques for NLP

Goals

- Begin development of CKY parser

- First stage: Conversion to CNF
  - Develop Representation for CFG
  - Manipulate/Transform Grammars
  - Investigate weakly equivalent grammars

Task

- Conversion:
  - Read in grammar rules from arbitrary CFG
  - Convert to CNF
  - Write out new grammar
- Validation:
  - Parse test sentences with original CFG
  - Parse test sentences with CFG in CNF

Approach

- May use existing models/packages to represent rules
  - Need RULE, RHS, LHS, etc
  - e.g. nltk.grammar.Production
- ***Conversion code must be your own***

Data

- ATIS (Air Travel Information System) data
  - Grammar provided in nltk-data and in the dropbox
  - Terminals in double-quotes
    - *the* → “the”
  - All required files on patas dropbox
- **NOTE**:
  - Grammar is fairly large (~193K Productions)
  - Grammar is fairly ambiguous (Test sentences may have 100 parses)
  - You will likely want to develop against a smaller grammar (e.g. toy.cfg)
  - You must submit a *condor* .cmd file
  - Also readme.{txt | pdf}

NLTK Grammars

```
>>> gr1 = nltk.data.load('grammars/large_grammars/atis.cfg')

>>> gr1.productions()[0]
ABBCL_NP -> QUANP_DTI QUANP_DTI QUANP_CD AJP_JJ NOUN_NP PRPRTCL_VBG

>>> gr1.productions()[0].lhs()
ABBCL_NP

>>> gr1.productions(lhs=gr1.productions()[1].lhs())
[ADJ_ABL -> only, ADJ_ABL->such]
```

Writing / Saving NLTK Grammar

- No built-in methods from NLTK to write your grammar to a file, so will need to roll your own
- Needs to be loadable by nltk.data.load
  - **NOTE!** NLTK determines the **start symbol** of a grammar in one of two ways:
    - Either: it's the *first nonterminal* it encounters when reading the grammar
    - Or: if one line says "%start SYMBOL", then SYMBOL will be the start
- Use the examples we provide as templates