

# HW #2

## 1. Goals

Through this assignment you will:

- Begin development of an automatic parser.
  - Homework #3 will require the implementation of the CKY algorithm.
- Develop and manipulate a representation for context-free grammars.
- Improve your understanding of Chomsky Normal Form (CNF) and weak grammatical equivalence through implementation.

## 2. Background

Please review the class slides and readings in the textbook on Chomsky Normal Form conversion.

## 3. Converting a Grammar to Chomsky Normal Form

As noted in the text, the CKY algorithm requires a grammar in Chomsky Normal Form (CNF). While it is not always intuitively clear how to write a grammar from scratch in CNF, it is fairly straightforward to convert a context-free grammar into a weakly equivalent grammar in CNF.

Following the approach outlined in class, implement a procedure to convert an input grammar of the form used for the first assignment to a new weakly equivalent grammar in CNF.

You will want to create data structures corresponding to RULE, RHS, LHS, etc. You may use whatever programming language you like, in accordance with the [Course Policy \(Links to an external site.\)](#). You may use existing implementations of these data structures in NLTK or other NLP toolkits (e.g. the Stanford Parser), but you ***must implement the grammar conversion algorithm yourself.***

## Requirements

The program you submit should do the following:

1. Read in an original context-free grammar.
2. Convert this grammar to Chomsky Normal Form.
3. Print out the rules of the converted grammar to a file.

## Programming

Create a program named `hw2_tocnf` to perform the conversion described above. Your program should be able to be invoked with:

```
hw2_tocnf.sh <input_grammar_file> <output_grammar_file>
```

where

- `<input_grammar_file>` is the name of the file holding the grammar to convert to CNF.
- `<output_grammar_file>` is the name of the output file for the CNF conversion. The file should be in the NLTK grammar format, but now in Chomsky Normal Form.

## Verification & Parse Comparison

Using your system from HW#1, you will parse a set of sentences with

- A General NLTK Context-Free Grammar, and...
- A weakly equivalent Chomsky Normal Form context-free grammar, created by your CNF conversion program.

**NOTE:** You will need to run your code yourself in both these conditions with the files specified below. You will include the output parse files in your submission tar file. You do **not** need to run this code as part of your condor file.

## Files:

### In the Dropbox

You will find the following files in `/dropbox/20-21/571W/hw2`:

- **atis.cfg:**
  - CFG grammar to test your CNF conversion program on.
- **sentences.txt:**
  - Set of sentences for validation of your CNF conversion.

- You will use your HW #1 system to parse these sentences both:
    - With the original **atis.cfg**
    - With your CNF-converted version of the ATIS grammar.
- **toy.cfg:**
  - Toy example NLTK-format grammar file.
- **other\_grammars:**
  - Directory with other NLTK-format CFG grammars for you to test with.

## Submission Files

- **hw.tar.gz**, containing the following:
  - **hw2\_tocnf.sh:**
    - Primary program file.
  - **hw2\_grammar\_cnf.cfg:**
    - Results of running your CNF conversion program on the atis.cfg grammar file.
  - **hw2\_orig\_output.txt:**
    - Results of parsing the test sentences using the original atis.cfg grammar file.
  - **hw2\_cnf\_output.txt:**
    - Results of parsing the test sentences using your CNF-converted grammar file (hw2\_grammar\_cnf.cfg from above).
  - **hw2.cmd:**
    - condor\_submit job definition that drives your parser, by calling the hw2\_tocnf.sh script.
    - Your CNF conversion program must run on patas using:
 

```
$ condor_submit hw2.cmd
```
  - Please see the [CLMS wiki pages \(Links to an external site.\)](#) for help on using condor.
  - All files created by the condor run should appear in the top level of the directory.
- **readme.{txt|pdf}:**
  - This file should describe and discuss your work on this assignment. Include problems you came across and how (or if) you were able to solve them, any insights, special features, and what you learned. Give examples if possible. If you were not able to complete parts of the project, discuss what you tried and/or what did not work.

**NOTE:** Also, please review the parses generated by the original grammar and those from the CNF grammar. Provide a brief discussion of similarities and differences.