# CSC485 A2

Michael Dzamba - 994803806

January 25, 2010

**Note:**
I used possibly smaller then 12pt font size for the grammar rules, in order to keep them on one line. All of the rules are also printed in at least 12pt in the attached grammar print out.

# 1 Creating/Testing a grammar

## 1.1 Lexicon design

The lexicon used for this grammar is a superset of the lexicon provided in the assignment examples. The reason the lexicon also includes the lexicon from the examples that should fail (instead of not including it and trivially failing on those) is because in a more general situation this might not work. That's why it was decided to give the grammar more information about the lexicon as oposed to less, and rely on its complexity to determine which strings do or do not belong to the grammar.

## 1.2 Grammar design

### 1.2.1 Support for simple sentences

The initial part of the grammar was modeled after the pass and fail examples provided in question 1.1 of the assignment. To do this the grammar must support,

- Appending adverbs to verbs

  VP → V AdvP | AdvP V | V | VP PP

- Differntiating between types of nouns

  NP → NP_Pro (pronoun) | NP_Prp (proper noun) | NP_N (other nouns)
  This is necessary, in order to evaluate the restrictions,
  *'Can't attach a PP to a name'* and *'Pronoun can't take an adjective'*
  Without differentiating between the types of nouns the grammar could not both pass and fail and the required examples. Also in order for the

grammar evaluate what kind of determiner a noun needs (if any), it must have this information.

The grammar satisfies the design specifications by only allowing adverbs to come directly before or after a verb and adjectives to come directly before a noun (other then a pronoun).

### 1.2.2 The auxillary system

To support the auxillary system for sentences containing intransitive verbs the following rules were added,

- V → Aux_modal modal | Aux_perfect perfect | Aux_progressive progressive | Aux_passive passive | V_ps
- modal → perfect_bs | progressive_bs | passive_bs | V_bs
  perfect_bs → Aux_perfect_bs perfect
  progressive_bs → Aux_progressive_bs progressive
  passive_bs → Aux_passive_bs passive
- perfect → progressive_pp | passive_pp | V_pp
  progressive_pp → Aux_progressive_pp progressive
  passive_pp → Aux_passive_pp passive
- progressive → passive_ger | V_ger
  passive_ger → Aux_passive_ger passive
- passive → V_pp

The first allows for a verb to start with any auxillary verb in the lexicon or non at all. The rules following ensure that any condition a auxillary verb has on the following verb is satisfied. These rules are sufficient to extend the grammar from the previous section ( 1.2.1 ) to the required grammar supporting auxiallary verbs.

### 1.2.3 Subcategorization

To handle transitive verbs the following method was taken,
Given the sentence *'NP some_verb* $A_1$ *...* $A_n$*'*, create the rule,
VP → AdvP $V_{A_1..A_n}$ $A_1$ .. $A_n$ |$V_{A_1..A_n}$ AdvP $A_1$ .. $A_n$ |AdvP $V_{A_1..A_n}$ AdvP $A_1$ .. $A_n$ |$V_{A_1..A_n}$ $A_1$ .. $A_n$
And add 'some_verb' to the lexicon in this new verb category,
$V_{A_1..A_n}$ → 'some_verb'
This is similar to what was outlined in the Jurafsky text (page 402)

Another design decision implemented in this section is start explicitly stating what kinds of prepositional phrases can follow a given verb phrase. This idea was taken from the Jurafsky text (page 401, see *'fly'* and *'travel'* in figure 12.6). For example it's possible to have,
*'I would like to fly [ from Boston ] [ to Philadelphia]'*
and not,
*'I would like to fly [ of Boston ] [ for Philadelphia]'*

Although the grammar still allows 'VP → VP PP', that can be phased out

and replaced by this kind of construct if need be.
In the examples provided for part *'1.3'* of the assignment, a differential treatment of count and non-count nouns appeared. Also it allows for potential differentiation in the grammar among what kind of propositional phrases can follow given verbs, since it might not always be true that every propositional phrase can follow every verb phrase.
*'Cheese was always on the menu'*
Thus the following rules were changed/added to the grammar,

NP_N → Det N_countable | Det AdjP N_countable | NP_N PP | Det_mass AdjP N_mass | Adj N_mass | Det_mass N_mass | N_mass

And to the lexicon,

Det → 'the' | 'The' | 'a' | 'A' | 'some' | 'Some'

Det_mass → 'the' | 'The' | 'some' | 'Some'

## 1.3 Grammar limitations

Some of the current grammar limitations are,

- **Plurals not supported**
  This aspect of the grammar was not implement because it was not required. Although it can be implemented by adding additional classes and conditions for plural form of every word. An example of this limitation is that,
  The grammar rejects, *'The cats with the long soft fur slowly ate'*
  Which is a very similar sentence to one that,
  The grammar accepts, *'The cat with the long soft fur slowly ate'*

- **Subject-verb argeement not supported**
  This aspect of the grammar was not implemented because it was not required, and features would make its implementation a lot easier. Some examples of this short coming are lack of support for plurals and,
  The grammar accepts, *'I has left'*
  The grammer rejects, '*'They have left'*

- **do not allow adverbs to mix with adjectives**
  This is a large limitation of the grammar. It can be implemented in a straightforward manner by sub-dividing adverbs into those that can come before a adjective and using additional rules to allow adjective phrases to have this property. An example of this limitation is,
  The grammar rejects, *'The cat with the really long fur slowly ate'*

- **Specificity of adverbs not supported**
  The grammar does not specify which adverbs can appear before or after a verb (since adverb/adjective combination is not supported, see above). An example of this is that,
  The grammar accepts, *'Nadia left really always really really'*

- **There exist many different subcategorizations and frames that are unsupported**
  According to Jurafsky text 'modern grammars distinguish as many as 100 subcategories'. This grammar only considers a handful. This limits the usefulness of the grammar.
  For example,
  *'he brought the cat cheese'*, is not accepted by the grammar because it is missing the trivalent form of the verb brought.

- **Supports only intransitive verbs with auxillary verbs** Transitive verbs are currently not support with auxiallary verbs. This once again can be implemented in a straightforward (but tedious) fashion, but was not required by the assignment examples or text. One example of this limitation is,
  The grammar rejects, *'Nadia will leave Ross'*

- **possible prepositonal phrases attached to a verb phrase**
  Another limitation of this grammar is that it does not explicitly specify all possible propositional phrases that can follow a verb phrase. This allows the following sentence to be accepted,
  *'Nadia will run of cheese'*

And there are many many many ... more limitations of this grammar. A lot of these are straightforward to implement and would add more lines to the grammar which can cause it lose generality without the proper use of feature structures. I did not implement additional grammar fixes to the above limitations because I wanted to keep the grammar around the required size so that it might be easier to read.

## 1.4 Testing Strategy

The testing strategy used was primarily focused on passing and failing on the required examples from the assignment. All sentences from the assignment were used as part of the testing sentences. Simpler forms of these sentences were also added (These were added to aid in debugging a failure to parse a more complex sentence).

There was also many test sentences which were added based on the written specification for the grammar from the assignment. These were added increase the likelihood that the grammar is not over-fitted to the example sentences from the assignment, and is hopefully as general as the assignment specifies.

To further test generalization of the grammar, whenever a grammar change was required to accept or reject a new example from the assignment, a different but similar test sentence was added to enforce the generalization of the new grammar change.

Lastly test sentences were added that clearly demonstrate the limitations of the grammar. These sentences were created by analyzing the design of the grammar and discovering overly specific sections which are not generalized to the complete english language. Many of these sentences are descibed in more detail in the section above ( Section 1.3 ). For example,
When looking at the rules, it is clear that there is no distinguishment between 'I' and 'They' in the lexicon, because they are both only in 'NPro_sub'. Because the lexicon does not distinguish between them neither can the production rules. A example of where the English language does distinguish between the two is,
*'I was at the store'* and *'They were at the store'*
To make this example more relevant to the grammar, we can change it to have one of the versions in the grammar and the other not,
*'I was on the cheese'* and *'They were on the cheese'*

Overall these compose the following four categories of test sentences,
1) Those that should pass and do, 2) Those that should not pass and do not, 3) Those that should pass and do not and 4) Those that should not pass and do.