Homework 3	COS/LIN 280

Parsing

Due: November 7, 2008

(Each subpart is worth 10 points)

1 Regular Languages

For each of the following languages, either show that it is regular by showing the FSA that accepts it or apply the pumping lemma to show that it is not regular (while there are languages that fall somewhere in between, we won't ask you about them). Be sure to clearly label the start and stop states.

1. this $(\text{sunday})^n$
2. $(\text{powerful} \mid \text{fast} \mid \text{nitro-burning})^n$ (funny) cars
3. $(\text{monster})^n$ truck $(\text{rally})^n$

Build a regular expression parser using tag pattern to find noun phrases containing plural head nouns, e.g. "many/JJ researchers/NNS", "two/CD weeks/NNS", "both/DT new/JJ positions/NNS". Try to do this by generalizing the tag pattern that handled singular noun phrases:

```
tagged_tokens = [("The", "DT"), ("enchantress", "NN"),
            ("clutched", "VBD"), ("the", "DT"), ("beautiful", "JJ"), ("hair", "NN")]
cp1 = nltk.RegexpParser(r"""
  NP: {<DT><JJ><NN>}      # Chunk det+adj+noun
      {<DT|NN>+}          # Chunk sequences of NN and DT
  """)
  print cp1.parse(tagged_tokens, trace=1)
```

2 Context-Free Grammars

Consider the following context free grammar:

Intermediate Rules	Terminal Rules
S → NP VP	D → a | the
NP → (D) NOM	V → slept | disappeared | loved | gave | relied
VP → V (NP) (NP)	N → cat | dog | roof | man
NOM → N	P → in | on | with
NOM → NOM PP	CONJ → and | or
VP → VP PP	
PP → P NP	
X → X CONJ X	

We designate the start state to be S, and X in the last rule stands for any part of speech (thus the last rule allows NP → NP CONJ NP but not NP→ NP CONJ VP).

1. Give a grammatical English sentence that has one an only one valid interpretation in this grammar. Draw the tree corresponding to it.
2. Give a grammatical English sentence that has more than one valid interpretation in this grammar. Draw two trees, and discuss whether the meaning in English is ambiguous.
3. Give a sentence that has a valid interpretation in this grammar but is not grammatical.
4. Give a grammatical English sentence using only these terminals that does not have a valid interpretation in this grammar. Why doesn't it have a valid interpretation?
5. Can you give an upper bound for the number of unique sequences of terminals this grammar admits? If not, why not?

3 Parsing with Features

The ungrammatical sentences produced in the first question can be addressed by adding features to our CFG. Examine the German grammar supplied with NLTK by using the command

```
>>> import nltk
>>> nltk.data.show_cfg('grammars/german.fcfg')
\% start S
# Grammar Rules
S -> NP[CASE=nom, AGR=?a] VP[AGR=?a]
NP[CASE=?c, AGR=?a] -> PRO[CASE=?c, AGR=?a]
NP[CASE=?c, AGR=?a] -> Det[CASE=?c, AGR=?a] N[CASE=?c, AGR=?a]
VP[AGR=?a] -> IV[AGR=?a]
VP[AGR=?a] -> TV[OBJCASE=?c, AGR=?a] NP[CASE=?c]
```

```
# Lexical Rules
# Singular determiners
# masc
Det[CASE=nom, AGR=[GND=masc,PER=3,NUM=sg]] -> 'der'
Det[CASE=dat, AGR=[GND=masc,PER=3,NUM=sg]] -> 'dem'
Det[CASE=acc, AGR=[GND=masc,PER=3,NUM=sg]] -> 'den'
...
```

Modify this feature-based grammar so that the following types of sentences are allowed:

- Gestern sah der hund die katze. / The dog saw the cat yesterday. (*past*)
- Heute kam ich. / I came today. (*past*)
- Heute folgen die Katzen ihm. / The cats follow him today. (*present*)

To do this, you will have to do the following:

1. Download the grammar from

   `http://www.cs.princeton.edu/~jbg/COS280/CP-V2-german.fcfg`

   If you start from the file provided by NLTK, it omits the plural rules for transitive verbs – you won't be docked points for this, but your grammar will be somewhat incomplete. If you download the file, it will also remove "mögen" and "helfen".
2. Add "past" and "present" tenses as features.
3. Add production rules for the past tense of "kommen", "folgen", and "sehen". (You can ignore or remove "mögen" and "helfen".)
4. Add "heute" and "gestern" to the lexicon as ADV; in addition, add the tense feature. "heute" can be either past or present tense, but "gestern" must only be past tense.
5. Add new production rules to allow the adverb to appear in first position.

For your reference, the preterite past for these German verbs is:

	sehen	kommen	folgen
ich (NUM=sg, PER=1)	sah	kam	folgte
du (NUM=sg, PER=2)	sahst	kamst	folgtest
er (NUM=sg, PER=3)	sah	kam	folgte
wir (NUM=pl, PER=1)	sahen	kamen	folgten
ihr (NUM=pl, PER=2)	saht	kamt	folgtet
sie (NUM=pl, PER=3)	sahen	kamen	folgten

Submit your grammar as a separate file readable by NLTK with your homework called "oitusername-german.fcfg".

4 Penn Treebank

In this section, let's consider the 10% sample of the Penn Treebank included in NLTK.

1. What is the minimum and maximum height of sentences in the Treebank? Give an example tree for both. How does the depth correlate with sentence length?

2. In the following sentence:

```
( (S
    (NP-SBJ-1
      (NP (NNP Rudolph) (NNP Agnew) )
      (, ,)
      (UCP
        (ADJP
          (NP (CD 55) (NNS years) )
          (JJ old) )
        (CC and)
        (NP
          (NP (JJ former) (NN chairman) )
          (PP (IN of)
            (NP (NNP Consolidated) (NNP Gold) (NNP Fields) (NNP PLC) ))))
      (, ,) )
    (VP (VBD was)
      (VP (VBN named)
        (S
          (NP-SBJ (-NONE- *-1) )
          (NP-PRD
            (NP (DT a) (JJ nonexecutive) (NN director) )
            (PP (IN of)
              (NP (DT this) (JJ British) (JJ industrial) (NN conglomerate) ))))))
    (. .) ))
```

   what does "(-NONE- *-1)" mean? Explain both in terms of this sentence specifically and what it means in general linguistically.

3. In order to build a PCFG, we need to estimate the probability of all of the production rules. What is the probability of each of the following production rules? This is a rare case where

we actually want the MLE estimate, as zeros rule out possible parse trees. How many unique non-terminal productions are there? How about terminal productions?

Production Rule	Probability
NP → DT JJ NNS	
VP → VB NP	
ADJP → JJ PP	
VP → MD VP	
NN → 'stock'	
IN → 'like'	
IN → 'on'	
IN → 'with'	
IN → 'about'	
IN → 'over'	

5 Probabilistic Context-Free Grammars

Consider the following sentence and the following PCFG:

time flies like an arrow

Intermediate Rules		Terminal Rules	
S → NP VP	0.8	N → time	0.5
S → VP	0.2	N → flies	0.3
VP → V NP	0.5	N → arrow	0.2
VP → V PP	0.3	V → time	0.3
VP → VP PP	0.2	V → flies	0.3
NP → Det N	0.3	V → like	0.4
NP → N	0.3	P → like	1.0
NP → N N	0.2	Det → an	1.0
NP → NP PP	0.2		
PP → P NP	1.0		

1. What are the four possible parse trees for this sentence?
2. Draw a CKY chart parse for the sentence and determine the most likely parse of this sentence. Show your work.
3. Suppose we didn't use the rule "P → like" but instead used the probabilities for the production rules for IN computed in the previous question. How would that affect the disambiguation? Is this linguistically reasonable?