Informatics 2A: Tutorial Sheet 7 (Week 9)

Earley Algorithm, Probabilistic CFGs and Parameter Estimation

Shay Cohen

1. Consider the following context free grammar with start symbol S:

S → NP VP	N → *saw*
S → Pro V	V → *saw*
NP → Pro	Pro → *I*
VP → V Det N	Det → *the*

   (a) Use the Earley algorithm from Lecture 19 to parse the sentence:

   I saw the saw

   (Note that you don't have to convert the grammar to CNF this time.)

   (b) Compare the behaviour of the Earley parser on this sentence to that of CYK. (You need not create the entire CYK chart for this.) How does the latter handle the ambiguity in *saw*?

2. Consider the following probabilistic context free grammar:

S	→	Subj VP (1.0)
VP	→	V Obj (0.5) | V Obj Obj (0.3) | V Small (0.2)
Small	→	Obj V (1.0)
Subj	→	*I* (0.3) | NP (0.7)
Obj	→	*her* (0.2) | NP (0.8)
NP	→	N (0.5) | Det N (0.5)
V	→	make (0.6) | duck (0.4)
N	→	duck (0.5) | goose (0.5)
Det	→	her (1.0)

   Use a probabilistic CYK-style algorithm to find the most probable parse for the sentence

   I make her duck

   and determine its probability. (For hints on the meanings of the various possible parses, see the Lecture 19 slides.)

   N.B. Do not convert the grammar to Chomsky Normal Form — instead, you should draw up a CYK-*style* parse chart for the grammar as it stands in the manner illustrated in the lectures. For small examples this is perfectly feasible, but you should check that you understand why a CYK-style algorithm has difficulties with non-CNF grammars in general. (Hint: think about what additional inferences you might need to make in order to work with grammars in this form.)

3. (a) Derive a PCFG grammar from the corpus below. Write down the rules of the grammar and calculate their probabilities. Assume for the purpose of the grammar that the corpus is lowercased (e.g. *The* has been replaced by *the*).

```
(S
    (NP She)
    (VP
        (VP
            (V saw)
            (NP (Det the) (N man)))
        (PP
            (P from)
            (NP (Det a) (N distance)))))

(S
    (NP Here)
    (VP
        (V is)
        (NP (Det a) (N telescope))))

(NP
    (NP (Det the) (N man))
    (PP (P with) (NP (Det the) (N guitar))))

(S
    (NP He)
    (VP
        (V saw)
        (NP (Det the) (N girl))))

(S
    (NP (Det The) (N man))
    (VP
        (V saw)
        (NP
            (NP (Det the) (N girl))
            (PP
                (P with)
                (NP (Det the) (N flowers))))))
```

(b) Draw all possible parse trees for the following sentence, and compute their probabilities using the probabilistic grammar you have created.

```
He saw the man with the telescope
```

(c) You can also use your probabilistic grammar for text prediction much like the suggestion box in a search engine. If a user typed `The girl saw`, what is the most likely suggestion for a possible completion?