Module Title: Informatics 2A
Exam Diet (Dec/April/Aug): Dec 2012 2012–13

Brief notes on answers:

PART A

1. (a) Regular, context-free, context-sensitive, unrestricted languages.
   (b) English theoretically allows arbitrary deep nesting, as in
       Jack built the house.
       Jack built the house the malt lay in.
       Jack built the house the malt the rat ate lay in.

       In these sentences, a sequence of \( n \) noun phrases (after ‘Jack built’) must be
       followed by \( n - 1 \) verbs for the sentence to be admissible. But the language
       \( \{a^n b^{n-1} \mid n \geq 1\} \) is a classic example of a non-regular language (proof by pump-
       ing lemma).
   (c) Context-sensitive languages are believed to be sufficient.

2. (a) (i). 1 state accepting.
        (ii). 1 state non-accepting.
        (iii). 2 states, one accepting.
   (b) Minimize both \( M_1 \) and \( M_2 \). Check whether the minimal automata are isomor-

3. The Viterbi matrix is

<table>
<thead>
<tr>
<th>man</th>
<th>bites</th>
<th>dog</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>0.6x0.5 = 0.3 ← 0.3x0.5x0.2 = 0.03, ∨ 0.06x0.8x0.3 = 0.0144</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>since 0.3x0.5 &gt; 0.16x0.8 since 0.03x0.5 &lt; 0.06x0.8</td>
</tr>
<tr>
<td>V</td>
<td>0.4x0.4 = 0.16 &lt; 0.3x0.5x0.4 = 0.06 &lt; 0.03x0.5x0.2 = 0.003</td>
<td></td>
</tr>
<tr>
<td></td>
<td>since 0.3x0.5 &gt; 0.16x0.2 since 0.03x0.5 &gt; 0.06x0.2</td>
<td></td>
</tr>
</tbody>
</table>

   So the most probable tag sequence is \textbf{NVN}.

4. (a) The parse trees are those represented by

   \( (S \text{ Which } (NP \text{ (N orange)) (VP (V flies) like (N bananas)) }) ? ) \)

   \( (S \text{ Which } (NP \text{ (N orange)) (VP (V flies) like (N bananas)) }) ? ) \)

   The first has probability 1.0 x 0.7 x 0.3 x 0.1 x 0.3 x 0.3 = 0.00189
   The second has probability 1.0 x 0.3 x 1.0 x 0.4 x 0.9 x 0.4 x 0.3 = 0.01296

   (b) To take account of singular/plural distinctions, the phrase categories \textbf{NP}, \textbf{VP}, \textbf{N},
   \textbf{V} should be parameterized on a number attribute with values \textbf{s}, \textbf{p}. A complete
   solution (not asked for here) would be:

   \[
   \begin{align*}
   S & \rightarrow \text{ Which } NP[x] \ VP[x] \\
   NP[x] & \rightarrow \ N[x] \mid A \ N[x] \\
   VP[x] & \rightarrow \ V[x] \ N[p] \mid V[x] \text{ like } N[p] \\
   N[s] & \rightarrow \text{ orange} \\
   N[p] & \rightarrow \text{ flies } \mid \text{ bananas} \\
   V[s] & \rightarrow \text{ flies } \mid \text{ throws} \\
   V[p] & \rightarrow \text{ like} \\
   A & \rightarrow \text{ orange}
   \end{align*}
   \]
5. (a) The format is $\alpha \rightarrow \beta$ where $\alpha, \beta \in (\Sigma \cup N)^*$, $\alpha$ contains at least one nonterminal and $|\alpha| \leq |\beta|$.

(b) Underlining the sequence to be expanded on the next line:

\[
\begin{align*}
S & \Rightarrow \text{exp} = 0 \text{ A}s \\
& \Rightarrow \text{exp} = 0 \text{ A A}s \\
& \Rightarrow \text{exp} = 0 \text{ A A} \\
& \Rightarrow \text{exp} = \text{A} 0 0 \text{ A} \\
& \Rightarrow \text{exp} 0 = 0 0 \text{ A} \\
& \Rightarrow \text{exp} 0 = 0 \text{ A} 0 0 \\
& \Rightarrow \text{exp} 0 = \text{A} 0 0 0 0 \\
& \Rightarrow \text{exp} 0 0 = 0 0 0 0
\end{align*}
\]

(c) The language is:

\[
\{\text{exp } 0^n = 0^{2^n} \mid n \geq 1\}
\]
PART B

6. (a) The smallest DFA has 3 states, 1 of them accepting.
(b) The equations are:

\[
\begin{align*}
X_0 &= 1X_1 - 1X_2 + \epsilon \\
X_1 &= 1X_2 + -1X_0 \\
X_2 &= 1X_0 + -1X_1
\end{align*}
\]

The language we are interested in is \(X_0\).

Substituting equation for \(X_2\) in \(X_1\), we get:

\[
X_1 = 1 - 1X_1 + (11 + -1)X_0
= (1-1)^*(11 + -1)X_0 \quad \text{(by AR)}
\]

So, substituting back in \(X_2\)

\[
X_2 = (1 - -1(1-1)^*(11 + -1))X_0
\]

Now substituting in \(X_0\), we get

\[
X_0 = 1(1-1)^*(11 + -1)X_0 + -1(1 + -1(1-1)^*(11 + -1))X_0 + \epsilon
= (1(1-1)^*(11 + -1) + -1(1 + -1(1-1)^*(11 + -1)))X_0 + \epsilon
= (1(1-1)^*(11 + -1) + -1(1 + -1(1-1)^*(11 + -1))))^* \quad \text{(by AR)}
\]

(c) We show \(\neg \text{P}\) (the negation of the pumping property).

Suppose \(k \geq 0\).

Consider \(x = \epsilon\), \(y = 1^k\) and \(z = (-1)^k\). Then \(xyz = 1^k(-1)^k \in L\) and clearly \(|y| \geq k\).

Suppose \(y = uvw\) where \(|v| \geq 1\).

Then \(uv^0w = uvw = 1^m\) for some \(m < k\). Whence \(xyv^0wz = 1^m(-1)^k \not\in L\) since \(m < k\).

Thus the pumping property fails for \(i = 0\).

(d) We use a single control state \(q\) and stack alphabet \(\Gamma = \{\bot, 1, -1\}\). The transitions from \(q\) to \(q\) are:

\[
\begin{align*}
1, \bot &: 1\bot \\
1, 1 &: 11 \\
1, -1 &: \epsilon \\
\epsilon, \bot &: \epsilon
\end{align*}
\]

(notation \(a, x : \alpha\) where \(a \in \Sigma \cup \{\epsilon\}\) is read symbol, \(x \in \Gamma\) is symbol popped off stack, and \(\alpha \in \Gamma^*\) is sequence pushed right-to-left onto stack).

7. (a) The CYK chart is

<table>
<thead>
<tr>
<th>cows</th>
<th>goats</th>
<th>sheep</th>
</tr>
</thead>
<tbody>
<tr>
<td>cows</td>
<td>I,AL,CL</td>
<td>CL</td>
</tr>
<tr>
<td>goats</td>
<td>I,AL,CL</td>
<td>I,AL,CL</td>
</tr>
<tr>
<td>and sheep</td>
<td>I,AL,CL</td>
<td>I,AL,CL</td>
</tr>
</tbody>
</table>
The grammar is ambiguous: *goats and sheep* can be parsed as either an AL or a CAL. (This example already features in the CYK chart above.)

The following is a typical LL(1) grammar of the required kind (minor variations are possible).

\[
\begin{align*}
L & \rightarrow I \ Rest \\
Rest & \rightarrow \epsilon \mid I \ ATail \mid I \ CTail \ and \ I \\
ATail & \rightarrow \epsilon \mid I \ ATail \\
CTail & \rightarrow \epsilon \mid I \ CTail
\end{align*}
\]

The requirement to use CTail is intended to ensure that at least one Follow set is non-trivial; it should also constrain the form of possible solutions so as to ease marking.

For the above LL(1) grammar, the First and Follow sets are

\[
\begin{align*}
\text{First}(L) & = \{ I \} \\
\text{Follow}(L) & = \{ \$ \} \\
\text{First}(Rest) & = \{ \epsilon, \ and, \ , \} \\
\text{Follow}(Rest) & = \{ \$ \} \\
\text{First}(ATail) & = \{ \epsilon, \ and \} \\
\text{Follow}(ATail) & = \{ \$ \} \\
\text{First}(CTail) & = \{ \epsilon, \ , \} \\
\text{Follow}(CTail) & = \{ \ and \}
\end{align*}
\]

For the above grammar, the parse table is

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>and</th>
<th>,</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>Rest</td>
<td>and</td>
<td>I ATail</td>
<td>I CTail \ and \ I</td>
</tr>
<tr>
<td>Rest</td>
<td>\epsilon</td>
<td>and</td>
<td>I ATail</td>
<td>I ATail</td>
</tr>
<tr>
<td>ATail</td>
<td>I ATail</td>
<td>\epsilon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTail</td>
<td>\epsilon</td>
<td>I ATail</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In general, LL(1) grammars aren’t appropriate for NLP. NL parsing typically involves non-trivial lookahead, in which case LL(1) can’t be used. But even when an NL grammar can be made LL(1), doing so may eradicate genuine ambiguities in an artificial way: a single interpretation will be selected, but the user can’t be expected to know which one, and the ambiguity won’t be flagged up.

8. (a) The semantics is as follows (only trivial variations are possible):

\[
\begin{align*}
S & \rightarrow \text{There is a NP} \quad \{ \exists x. \ NP.Sem(x) \} \\
NP & \rightarrow \text{N} \quad \{ \ N.Sem \} \\
NP & \rightarrow \text{ANP} \quad \{ \ ANP.Sem \} \\
NP & \rightarrow \text{ANP that VP} \quad \{ \ \lambda x. \ ANP.Sem(x) \land VP.Sem(x) \} \\
ANP & \rightarrow \text{AP N} \quad \{ \ \lambda x. \ N.Sem(x) \land AP.Sem(x) \} \\
AP & \rightarrow \text{A} \quad \{ \ A.Sem \} \\
AP & \rightarrow \text{A}_1 \ or \ A_2 \quad \{ \ \lambda x. A_1.Sem(x) \lor A_2.Sem(x) \} \\
VP & \rightarrow \text{touches every NP} \quad \{ \ \lambda x. \forall y. \ NP.Sem(y) \Rightarrow touches(x,y) \} \\
VP & \rightarrow \text{touches some NP} \quad \{ \ \lambda x. \exists y. \ NP.Sem(y) \land touches(x,y) \} \\
N & \rightarrow \text{sphere} \quad \{ \ \lambda x. sphere(x) \}, \text{etc.} \\
A & \rightarrow \text{red} \quad \{ \ \lambda x. red(x) \}, \text{etc.}
\end{align*}
\]

(b) The logical formula is

\[
\exists x. \ cube(x) \land red(x) \land (\forall y. (sphere(y) \land (blue(y) \lor green(y))) \Rightarrow touches(x,y))
\]

(c) The last two parts of the question are harder, but similar to things they have seen. They are deliberately allocated only a modest number of marks.

\[
\begin{align*}
VP & \rightarrow \text{touches QNP} \quad \{ \ \lambda x. \ QNP.Sem(\lambda y. touches(x,y)) \} \\
QNP & \rightarrow \text{every NP} \quad \{ \ AP. \forall y. (NP.Sem(y) \Rightarrow P(y)) \} \\
QNP & \rightarrow \text{some NP} \quad \{ \ AP. \exists y. (NP.Sem(y) \land P(y)) \}
\end{align*}
\]
(d) The raw lambda expression given by the semantics above is:
\[ \lambda x. (\lambda P. \exists y. (\lambda x. \text{cube}(x))(y) \land P(y)) (\lambda y. \text{touches}(x,y)) \]
This reduces via four \( \beta \)-reduction steps to:
\[ \lambda x. \exists y. \text{cube}(y) \land \text{touches}(x,y) \]