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 pumping 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 isomorphic.

3. The Viterbi matrix is

	man	bites	dog
N	0.6x0.5 = 0.3	← 0.3x0.5x0.2 = 0.03, since 0.3x0.5 > 0.16x0.8	↙ 0.06x0.8x0.3 = 0.0144 since 0.03x0.5 < 0.06x0.8
V	0.4x0.4 = 0.16	↖ 0.3x0.5x0.4 = 0.06 since 0.3x0.5 > 0.16x0.2	↖ 0.03x0.5x0.2 = 0.003 since 0.03x0.5 > 0.06x0.2

   So the most probable tag sequence is NVN.

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

   (S Which (NP (N orange)) (VP (V flies) like (N bananas)) ?)
   (S Which (NP (A orange)(N flies)) (VP (V 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 NP, VP, N, V should be parameterized on a number attribute with values s, p. A complete solution (not asked for here) would be:

$$
\begin{array}{rcl}
\mathsf{S} & \rightarrow & \text{Which } \mathsf{NP}[\text{x}]\ \mathsf{VP}[\text{x}]\ ? \\
\mathsf{NP}[\text{x}] & \rightarrow & \mathsf{N}[\text{x}] \ \mid\ \mathsf{A}\ \mathsf{N}[\text{x}] \\
\mathsf{VP}[\text{x}] & \rightarrow & \mathsf{V}[\text{x}]\ \mathsf{N}[\text{p}] \ \mid\ \mathsf{V}[\text{x}] \text{ like } \mathsf{N}[\text{p}] \\
\mathsf{N}[\text{s}] & \rightarrow & \text{orange} \\
\mathsf{N}[\text{p}] & \rightarrow & \text{flies} \ \mid\ \text{bananas} \\
\mathsf{V}[\text{s}] & \rightarrow & \text{flies} \ \mid\ \text{throws} \\
\mathsf{V}[\text{p}] & \rightarrow & \text{like} \\
\mathsf{A} & \rightarrow & \text{orange}
\end{array}
$$

5. (a) The format is $\alpha \rightarrow \beta$ where $\alpha, \beta \in (\Sigma \cup N)^*$, α contains at least one nonterminal and $|\alpha| \leq |\beta|$.

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

$$
\begin{aligned}
\underline{\mathsf{S}} &\Rightarrow \mathsf{exp} = \mathsf{0}\ \underline{\mathsf{As}} \\
&\Rightarrow \mathsf{exp} = \mathsf{0}\ \mathsf{A}\ \underline{\mathsf{As}} \\
&\Rightarrow \mathsf{exp} = \underline{\mathsf{0}\ \mathsf{A}}\ \mathsf{A} \\
&\Rightarrow \mathsf{exp}\ \underline{=\ \mathsf{A}}\ \mathsf{0}\ \mathsf{0}\ \mathsf{A} \\
&\Rightarrow \mathsf{exp}\ \mathsf{0} = \mathsf{0}\ \underline{\mathsf{0}\ \mathsf{A}} \\
&\Rightarrow \mathsf{exp}\ \mathsf{0} = \underline{\mathsf{0}\ \mathsf{A}}\ \mathsf{0}\ \mathsf{0} \\
&\Rightarrow \mathsf{exp}\ \mathsf{0}\ \underline{=\ \mathsf{A}}\ \mathsf{0}\ \mathsf{0}\ \mathsf{0}\ \mathsf{0} \\
&\Rightarrow \mathsf{exp}\ \mathsf{0}\ \mathsf{0} = \mathsf{0}\ \mathsf{0}\ \mathsf{0}\ \mathsf{0}
\end{aligned}
$$

(c) The language is:

$$\{\mathsf{exp}\ \mathsf{0}^n = \mathsf{0}^{2^n} \mid n \geq 1\}$$

PART B

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

   (b) The equations are:

$$\begin{aligned} X_0 &= 1X_1 + {-1}X_2 + \epsilon \\ X_1 &= 1X_2 + {-1}X_0 \\ X_2 &= 1X_0 + {-1}X_1 \end{aligned}$$

   The language we are interested in is X_0.
   Substituting equation for X_2 in X_1, we get:

$$\begin{aligned} X_1 &= 1{-1}X_1 + (11 + {-1})X_0 \\ &= (1{-1})^*(11 + {-1})X_0 && \text{(by AR)} \end{aligned}$$

   So, substituting back in X_2

$$X_2 = (1 + {-1}(1{-1})^*(11 + {-1}))X_0$$

   Now substituting in X_0, we get

$$\begin{aligned} X_0 &= 1(1{-1})^*(11 + {-1})X_0 + {-1}(1 + {-1}(1{-1})^*(11 + {-1}))X_0 + \epsilon \\ &= (1(1{-1})^*(11 + {-1}) + {-1}(1 + {-1}(1{-1})^*(11 + {-1})))X_0 + \epsilon \\ &= (1(1{-1})^*(11 + {-1}) + {-1}(1 + {-1}(1{-1})^*(11 + {-1})))^* && \text{(by AR)} \end{aligned}$$

   (c) We show $\neg$ 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 $uv^0w = uw = 1^m$ for some $m < k$. Whence $xyv^0wz = 1^m({-1})^k \notin L$ since $m < k$.
   Thus the pumping property fails for $i = 0$.

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

$$\begin{array}{rcl@{\qquad\qquad}rcl} 1,\ \perp &:& 1\perp & {-1},\ \perp &:& {-1}\perp \\ 1,\ 1 &:& 11 & {-1},\ {-1} &:& {-1}{-1} \\ 1,\ {-1} &:& \epsilon & {-1},\ 1 &:& \epsilon \\ \epsilon,\ \perp &:& \epsilon & & & \end{array}$$

   (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

	cows	,	goats	and	sheep
cows	I,AL,CL		CL		CAL,L
,		I,AL,CL			
goats			I,AL,CL		AL,CAL,L
and				I,AL,CL	
sheep					I,AL,CL

(b) 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.)

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

$$\begin{array}{rcl} \mathsf{L} & \rightarrow & \mathsf{I\ Rest} \\ \mathsf{Rest} & \rightarrow & \epsilon \mid \textit{and}\ \mathsf{I\ ATail} \mid ,\ \mathsf{I\ CTail}\ \textit{and}\ \mathsf{I} \\ \mathsf{ATail} & \rightarrow & \epsilon \mid \textit{and}\ \mathsf{I\ ATail} \\ \mathsf{CTail} & \rightarrow & \epsilon \mid ,\ \mathsf{I\ CTail} \end{array}$$

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.

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

$$\begin{array}{rclcrcl} \text{First}(\mathsf{L}) & = & \{\ \mathsf{I}\ \} & & \text{Follow}(\mathsf{L}) & = & \{\ \$\ \} \\ \text{First}(\mathsf{Rest}) & = & \{\ \epsilon,\ \textit{and},\ ,\ \} & & \text{Follow}(\mathsf{Rest}) & = & \{\ \$\ \} \\ \text{First}(\mathsf{ATail}) & = & \{\ \epsilon,\ \textit{and}\ \} & & \text{Follow}(\mathsf{ATail}) & = & \{\ \$\ \} \\ \text{First}(\mathsf{CTail}) & = & \{\ \epsilon,\ ,\ \} & & \text{Follow}(\mathsf{CTail}) & = & \{\ \textit{and}\ \} \end{array}$$

(e) For the above grammar, the parse table is

	I	*and*	,	$
L	I Rest			
Rest		*and* I ATail	, I CTail *and* I	ϵ
ATail		*and* I ATail		ϵ
CTail		ϵ	, I ATail	

(f) 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):

$$\begin{array}{rcll} \mathsf{S} & \rightarrow & \text{There is a } \mathsf{NP} & \{\ \exists\,\text{x}.\ \mathsf{NP.Sem}(\text{x})\ \} \\ \mathsf{NP} & \rightarrow & \mathsf{N} & \{\ \mathsf{N.Sem}\} \\ \mathsf{NP} & \rightarrow & \mathsf{ANP} & \{\ \mathsf{ANP.Sem}\} \\ \mathsf{NP} & \rightarrow & \mathsf{ANP}\ \text{that}\ \mathsf{VP} & \{\ \lambda\text{x}.\mathsf{ANP.Sem}(\text{x}) \wedge \mathsf{VP.Sem}(\text{x})\} \\ \mathsf{ANP} & \rightarrow & \mathsf{AP\ N} & \{\ \lambda\text{x}.\mathsf{N.Sem}(\text{x}) \wedge \mathsf{AP.Sem}(\text{x})\ \} \\ \mathsf{AP} & \rightarrow & \mathsf{A} & \{\ \mathsf{A.Sem}\} \\ \mathsf{AP} & \rightarrow & \mathsf{A}_1\ \text{or}\ \mathsf{A}_2 & \{\ \lambda\text{x}.\mathsf{A}_1.\mathsf{Sem}(\text{x}) \vee \mathsf{A}_2.\mathsf{Sem}(\text{x})\ \} \\ \mathsf{VP} & \rightarrow & \text{touches every } \mathsf{NP} & \{\ \lambda\text{x}.\forall\ \text{y}.\ \mathsf{NP.Sem}(\text{y}) \Rightarrow \textit{touches}(\text{x,y})\ \} \\ \mathsf{VP} & \rightarrow & \text{touches some } \mathsf{NP} & \{\ \lambda\text{x}.\exists\ \text{y}.\ \mathsf{NP.Sem}(\text{y}) \wedge \textit{touches}(\text{x,y})\ \} \\ \mathsf{N} & \rightarrow & \text{sphere} & \{\ \lambda\text{x}.\textit{sphere}(\text{x})\ \},\ \text{etc.} \\ \mathsf{A} & \rightarrow & \text{red} & \{\ \lambda\text{x}.\textit{red}(\text{x})\ \},\ \text{etc.} \end{array}$$

(b) The logical formula is

$\exists\text{x}.\ \textit{cube}(\text{x}) \wedge \textit{red}(\text{x}) \wedge (\forall\text{y}.\ (\textit{sphere}(\text{y}) \wedge (\textit{blue}(\text{y}) \vee \textit{green}(\text{y}))) \Rightarrow \textit{touches}(\text{x,y}))$

(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.

$$\begin{array}{rcll} \mathsf{VP} & \rightarrow & \text{touches } \mathsf{QNP} & \{\ \lambda\text{x}.\ \mathsf{QNP.Sem}(\lambda\text{y}.\textit{touches}(\text{x,y}))\ \} \\ \mathsf{QNP} & \rightarrow & \text{every } \mathsf{NP} & \{\ \lambda\text{P}.\ \forall\ \text{y}.(\mathsf{NP.Sem}(\text{y}) \Rightarrow \text{P}(\text{y})) \\ \mathsf{QNP} & \rightarrow & \text{some } \mathsf{NP} & \{\ \lambda\text{P}.\ \exists\ \text{y}.(\mathsf{NP.Sem}(\text{y}) \wedge \text{P}(\text{y})) \end{array}$$

(d) The raw lambda expression given by the semantics above is:
λx. (λP.$\exists$y. (λx.*cube*(x))(y) $\wedge$ P(y)) (λy.*touches*(x,y))
This reduces via four β-reduction steps to:
λx.$\exists$y. *cube*(y) $\wedge$ *touches*(x,y)