CA215 Languages and Computability
Autumn 2008

Attempt **three** questions. All questions carry equal marks.

Q 1.
(i) Define the constraints on α and β in a finite-state rewrite rule of the form $\alpha \longrightarrow \beta$.

(ii) Define a Deterministic Finite Automaton in terms of the quintuple $M = (Q, \Sigma, \delta, q_0, F)$.

(iii) For the language $x^a y^b$, give the transition function.

(iv) Draw the finite-state network for the same language corresponding to the transition function you provided in (iii).

(v) Given the configuration $(0, xxyyy)$, trace the execution of the DFA in (iii) and (iv).

(vi) Provide the formal grammar in terms of the quadruple $\langle V_t, V_n, P, S\rangle$ that describes the language $x^a y^b$.

Q 2.
(i) Define the constraints on α and β in a context-free rewrite rule of the form $\alpha \longrightarrow \beta$.

(ii) Consider the following Type 2 grammar rules:

S	$\longrightarrow$	NP VP
NP	$\longrightarrow$	DET N
NP	$\longrightarrow$	PN
VP	$\longrightarrow$	V
VP	$\longrightarrow$	V NP
DET	$\longrightarrow$	a
N	$\longrightarrow$	woman
PN	$\longrightarrow$	peter
V	$\longrightarrow$	loves

Write down the strings in the language permitted by this grammar. Show their derivations using trees.

(iii) Show which of the rules in the grammar could be rules in an FSG. Explain why.

(iv) Which rules in the set in (ii) make this grammar a Type 2 grammar? Explain why.

Q 3.
(i) Define the constraints on α and β in a context-sensitive rewrite rule of the form $\alpha \longrightarrow \beta$.

(ii) Define a Linear Bounded Automaton in terms of the quintuple $M = (Q, \Sigma, \Gamma, q_0, \delta)$.

(iii) Assuming the following ruleset:

- $N \rightarrow lmn$
- $N \rightarrow lNMn$
- $nM \rightarrow Mn$
- $mM \rightarrow mm$

What are the shortest three strings in this language?

(iv) Show the derivation of the 2nd and 3rd shortest strings using trees;

(v) Show the derivation of the 2nd and 3rd shortest strings using string manipulation.

(vi) Does it make a difference in which order you apply the rules? Explain your answer.

Q 4.
(i) Define the constraints on α and β in a Type 0 grammar rewrite rule of the form $\alpha \longrightarrow \beta$.

(ii) Give a formal definition of a Turing machine in terms of the quintuple $M = (Q, \Sigma, \Gamma, q_0, \delta)$.

(iii) Construct a Turing machine which erases the input string, assuming unary input. Comment on the approach you have taken compared to other possible solutions.

(iv) For the initial configuration $(q_0, \underline{1}11\#)$, show how the Turing machine you provided in (iii) copes with that input tape.

Q 5.

The following are expressions in the *untyped* λ-calculus. Reduce the expressions as far as possible. Remember that abstraction is right-associative and that application is left-associative.

(i)	$(\lambda p.qp)\ x$
(ii)	$(\lambda p.qr)\ x$
(iii)	$(\lambda y.\lambda x.yrxs)\ a\ b$
(iv)	$(\lambda y.((\lambda x.yrxs)\ a))\ b$
(v)	$((\lambda p.\lambda q.pxrq)\ q)\ a$
(vi)	$((\lambda p.(\lambda q.((\lambda r.psqr)\ a)))\ b)$
(vii)	OR FALSE TRUE
(viii)	NOT (NOT TRUE)
(ix)	IFTHENELSE TRUE do skip

where

TRUE :=	$(\lambda x.(\lambda y.(x)))$
FALSE:=	$(\lambda x.(\lambda y.(y)))$
NOT :=	$(\lambda p.(p\ FALSE\ TRUE))$
OR:=	$(\lambda p.(\lambda q.(p\ TRUE\ q)))$
IFTHENELSE :=	$(\lambda p.(\lambda x.(\lambda y.(p\ x\ y))))$