CA215 Languages and Computability
Spring 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) Write down the strings in the language permitted by the following grammar:

$\langle V_t = \{$sleep, walk, shout, er, s, pen, keyboard$\}$,
$V_n = \{$PL, N, V$\}$,

$$
\begin{array}{lllll}
P & = & \{ & \text{PL} & \longrightarrow & \text{N s} \\
 & & & \text{N} & \longrightarrow & \text{V er} \\
 & & & \text{N} & \longrightarrow & \text{pen}|\text{keyboard} \\
 & & & \text{V} & \longrightarrow & \text{sleep}|\text{walk}|\text{shout}\},
\end{array}
$$

$S =$ PL$\rangle$

(iii) Draw the finite-state network that accepts this language.

(iv) Given the configuration $(q_0, \textit{walk er s})$, trace the execution of the DFA in (iii).

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

(ii) Define a Pushdown Automaton in terms of the sextuple $M = (Q, \Sigma, \Gamma, q_0, F, \delta)$.

(iii) Consider the following ruleset:

- $E \rightarrow E\ +\ E$
- $E \rightarrow E\ -\ E$
- $E \rightarrow a$

(a) Show that this grammar is ambiguous by giving two distinct parse trees for the sentence $a + a - a$.

(b) For **one** of the trees that you derived in (a), show that different derivations can result in the same parse tree.

(c) Show which of the rules above could be rules in an FSG. Explain why.

(d) Which of the rules above make this grammar a CFG? Explain why.

(e) Describe how CFGs can handle types of recursion that FSGs cannot.

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:

- $W \rightarrow xyz$
- $W \rightarrow xWYz$
- $zY \rightarrow Yz$
- $yY \rightarrow yy$

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) Explain the difference between the transition functions of Turing machines and Linear Bounded Automata.

(iv) Assuming unary input, describe the transition function of the Turing machine **UN+1**, which adds one to a number. Comment on the approach you have taken compared to other possible solutions.

(v) For the initial configuration $(q_0, \underline{1}1\#)$, show how the Turing machine you provided in (v) 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)	$(\lambda p.pqp)\ a$
(ii)	$(\lambda p.qb)\ b$
(iii)	$((\lambda p.(\lambda q.(qpq)))\ a)\ b$
(iv)	$\lambda p.\lambda q.\lambda r.pqr\ a\ b\ c$
(v)	$(\lambda q.((\lambda p.qrps)\ x))\ y$
(vi)	$(\lambda x.(\lambda y.xy))\ y\ x$
(vii)	NOT FALSE
(viii)	AND TRUE FALSE
(ix)	IFTHENELSE FALSE loop exit

where

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