Indian Institute of Technology Delhi
Department of Computer Science and Engineering

CSL302/ CS232F | Programming Languages | Quiz I
January 19, 2006 | 12:40–12:55 | Maximum Marks: 10

Q1 [4+4+2 marks] **Regular Languages and Context Free Grammars.**
Give a Context Free Grammar for generating exactly those strings which represent *Natural Numerals* (decimal representations of natural numbers). [Hint: Identify a non-terminal with each automaton state, and give a production for each transition.]

Answer: Let S, N be the non-terminals; $0 \ldots 9$ are the terminals, and S is the start state.

S	$\rightarrow$	0	Zero, from initial state
S	$\rightarrow$	nN	$n \in \{1 \ldots 9\}$ Initial non-zero, then further digits
N	$\rightarrow$	dN	$d \in \{0 \ldots 9\}$ Further digits, stay in state
N	$\rightarrow$	ϵ	No further digits

△

Generalize this idea to outline a construction which proves that any set of strings (over an alphabet Σ) that is recognized by a Finite State Automaton (such sets are called *regular languages*) can be generated a Context Free Grammar (such languages are called *Context Free Languages*).

Answer: Let $\mathcal{A}$ be an automaton which recognizes the given regular language over alphabet Σ. With each state q_i of the automaton, associate a *non-terminal* Q_i.

For each arc $q \xrightarrow{c} q$ (where q, q' are states in the automaton $\mathcal{A}$ and $c \in \Sigma$), we have a grammar rule $Q \rightarrow cQ'$ where Q, Q' correspond to automaton states q, q' respectively.

For each final state q of $\mathcal{A}$, we introduce a rule $Q \rightarrow \epsilon$. △

If every regular language is also a context-free language, why do we separate lexical analysis from parsing? (One sentence only)

Solution: For the same reason that one does not consider a spelling mistake to be a grammatical erro (;-) — the framework and algorithms for lexical analysis are simpler and more efficient (they are linear in terms of the input size) than those for parsing. △