# Maths for Computation Theory 2003

## Paper 11 Solution

Refers to Part B section 2, "Using automata to recognise languages", Part B section 3, "Algebraic operations on languages", and Part B section 4, "Regular languages are representable events".

$LL' = \{ww' \mid w \in L, w' \in L'\}$, a language over the finite alphabet $S \cup S'$.

$L + L' = \{w \mid (w \in L) \vee (w \in L')\}$, the union of the two languages, also over the finite alphabet $S \cup S'$. ALTERNATION

[ This definition extends to any finite union. More generally, if $\{L_t\}_{t \in T}$ is an arbitrary family of languages over the same finite alphabet $S$, then

$$\sum_{t \in T} L_t = \{w \mid w \in L_t \text{ for some } t \in T\}$$

is also a language over $S$, the (possibly infinite) sum of the languages $\{L_t\}$.]

Paper 11. Solution (ctd.)

The unit language (event) 1 is defined over any alphabet S, and consists of the empty word ε of length 0 only.

The zero (empty) language (event) 0 is the null set of words.

[Now define for any language L over S

$$L^0 = 1$$

$$L^{n+1} = L^n L \qquad \forall \; n \geq 0 ]$$

$$L^* = \sum_{n=0}^{\infty} L^n,$$ the Kleene closure of L — ARBITRARY REPETITION

Paper 11 Solution (ctd)

Let $S$ be a finite alphabet. For each $s \in S$ let $s$ also represent the language (event) $\{s\}$, which consists of a single word $s$ of length 1.

Then a *regular language* over $S$ is one defined from the languages $0$, $1$ and $s \in S$ by application of the regular operators ALTERNATION, CONCATENATION and ARBITRARY REPETITION.

Paper 11 Solution (ctd.)

An NDFA over $S$ differs from a DFA in that the transition $(q, s) \to q' \in Q$ is not uniquely determined; in general the successor state is chosen from some $Q' \subseteq Q$, so that the transition function has signature

$$F : (Q \times S) \to P(Q)$$

rather than $f : (Q \times S) \to Q$ as for a DFA.

A word $w$ is accepted by $M$ if, when $M$ is started in initial state $z$ and $w$ is applied, some choice of successor states is such that $z_w \in A$.

In order to define a DFA $\bar{M}$ which accepts the same words as $M$ we must keep track of the possible active states as symbols of $w$ are applied. We define the equivalent DFA by introducing $\bar{F}$, $\bar{\imath}$, etc. Define

$$\bar{M} = (\bar{Q}, \bar{S}, \bar{F}, \bar{\imath}, \bar{A}), \quad \text{where}$$

$$\bar{Q} = P(Q), \quad \bar{S} = S, \quad \bar{\imath} = \{i\},$$

$$\bar{F}(P, s) = \bigcup_{q \in P} F(p, s)$$

$$\bar{A} = \{ P \subseteq Q \mid P \cap A \neq \emptyset \}.$$

Evidently $w$ is accepted by $\bar{M}$ if and only if it is accepted by the NDFA $M$.

---

Using similar notation, suppose that $L, L'$ are accepted by DFA $M, M'$.

Paper 11 Solution (ctd)

We define the NDFA $M_\circ$ to accept the language $LL'$ as follows:

$$(Q + Q',\ S \cup S',\ ((r \in A) \to \{r, r'\}, \{r\}),\ F_\circ,\ A')$$

where we abuse notation slightly [$in_2(A')$]. We need only define the transition function $F_\circ$:

Again abusing notation:

for $q \in Q$, $x \in (S \cup S')$

$$F_\circ(q, x) = (x \notin S) \to \emptyset,$$
$$(f(q,x) \notin A) \to \{f(q,x)\},$$

ACCEPTING! $\{f(q,x), r'\}$

for $q' \in Q'$, $x' \in (S \cup S')$

$$F_\circ(q', x') = (x' \in S') \to \{f'(q', x')\}, \emptyset$$

The words accepted by $M_\circ$ are precisely the words of $LL'$.

Paper 11 Solution (ctd.)

Suppose that $L$ is a regular language over finite alphabet $S$, hence denoted by some regular expression over $S$.

We prove by structural induction that any language denoted by a regular expression over $S$ is accepted by some DFA.

It is easy to build DFA to accept the minimal expressions: $0$, $1$, and languages $s \in S$.

Each algebraic operator $+$, $\cdot$, $*$ can be handled by constructing an NDFA to accept the language, then carrying out the construction of an equivalent DFA.