COMPUTER SCIENCE TRIPOS, Part II (General)
DIPLOMA IN COMPUTER SCIENCE

Mathematics for Computation Theory **(KM 1999)**

Kleene's Theorem

The aim of the lectures on Finite Automata is to prove important results in theoretical Computer Science fairly rigorously, using the techniques introduced in Part A. The material is not always easy, but I hope that the intuition is clear.

There are a number of conceptual sticking points, but the first and probably the most serious is the introduction of NDFA. The presentations via *random choice of action* and *parallel execution* are essentially equivalent, but it's easier to kill off a machine that's exploring paths in parallel, and that's the view that I adopt when proving Kleene's synthesis theorem (that "one can build a DFA to recognise any regular language").

Arden's rule for events is that the event A^*B is the **least** solution of the event equation

$$X \;=\; B + A.X$$

Stanat and McAllister present this result beautifully, including the demonstration that if A does not contain the null string, then the solution is unique. One way of thinking about this result is to view it as an inductive definition that enables the construction of better and better approximations to X: start off by defining

$X_0 = \phi$ the *null event*, the empty set of strings

and by induction extend to

$$X_{n+1} \;=\; B + A.X_n \;.$$

We prove Kleene's analysis theorem by extending Arden's rule for events to the case of event matrices. There is no particular technical difficulty in the extension, but again the notation may be intimidating. Once Arden's Rule has been established for matrices we apply it to the action of the given DFA whose event transition matrix is M: the entries of M are (possibly empty) events containing only strings of length 1. If you like, M gives the operational description for the action of M. The language recognised by a particular output of M is an entry in the matrix M^*.

Arden's rule for event matrices allows us to characterise M^* as the **least** solution of the (k x k) event matrix equation

$$X \;=\; I + M.X$$

and the inductive view generates successively longer strings in each element of M^*. Diploma 1996, Paper 1, Q 9 shows that in this case the solution is in fact unique.

That's it, really. The formula for the Kleene closure of an $(m+k)$ x $(m+k)$ matrix M partitioned symmetrically follows at once from the uniqueness of the solution; from that result it's simple to show by induction that the entries of M^* are regular expressions over the entries of M.

The last lecture covers the *Pumping Lemma*, the first tool encountered for proving systematically that particular languages are **not** regular. Examples of such languages usually depend on the fact that a DFA can't maintain a count of arbitrary size.

Kleene's theorem shows that regular languages are closed under (symmetric) difference, and it is easy to establish algorithms for determining the equivalence of two languages described by different regular expressions over the same alphabet. Unfortunately taking differences requires the formation of complements, and so explicitly defined NDFA are ruled out. The notes show why the complexity of the equivalence problem may well be high: a formal proof that the problem is *non-elementary* is given in Hopcroft & Ullman's 1974 book "The Design and Analysis of Computer Algorithms".

Note The inductive view of Arden's rule for event matrices constructs a sequence of approximations to M^* :

$$X_0 = \phi \qquad \text{the \textit{null event matrix}, each element the empty set}$$

by induction extending to

$$X_{n+1} \;=\; I + M.X_n \;, \qquad \text{so that} \quad X_{n+1} \;=\; I + M + M^2 + M^3 + \;...\; + M^n \,..$$

Why does the sequence $\{X_n\}$ converge ? Two different approaches, both familiar from denotational semantics, will work in this case. In each we consider the function defined on $\mathfrak{M}_k$, the set of all $(k$ x $k)$ event matrices, by

$$f(X) \;=\; I + M.X$$

with binary operation + representing set-theoretic union elementwise on matrices M_1 , M_2 .

a) consider $\mathfrak{M}_k$ as a complete partially ordered set, as suggested in the notes. Then $f : \mathfrak{M}_k \rightarrow \mathfrak{M}_k$ is a continuous function, hence has a least fixed-point: this fixed-point is the least upper bound of the countable chain $\{X_n\}$.

b) consider $\mathfrak{M}_k$ as a complete metric space, the **finite** product of k^2 copies of the set of events $\mathfrak{E}$. Within each copy of $\mathfrak{E}$ define the distance between distinct events E_1 , E_2 by

$$\rho(E_1, E_2) = 2^{-l} \;, \quad \text{where} \quad l \;=\; Min\;\{length(w) \mid w \in (E_1 \vartriangle E_2)\}.$$

Then $(\mathfrak{E},\rho)$ is a complete metric space, hence so also is $\mathfrak{M}_k$, taking the distance between matrices M_1 , M_2 to be the maximum of the distances elementwise between component events. The binary operation + is actually the **symmetric difference** in the case being considered, since each element in the matrix M^n contains words of length n only. The mapping $f : \mathfrak{M}_k \rightarrow \mathfrak{M}_k$ is a contraction, hence has a unique fixed-point: this fixed-point is the limit of the Cauchy sequence $\{X_n\}$.