QUASIAUTOMATA AND APPLICATIONS*

by

Ernst Leiss

Research Report CS-77-34

Department of Computer Science

University of Waterloo
Waterloo, Ontario, Canada

November 1977

* This research was supported by the National Research Council of Canada under Grant No.A-1617.

for all $q \in Q$, $a \in A$, $\underset{\sim}{A}$ is called deterministic. M is extended to $P_0(Q) \times A^* \to P_0(Q)$ in the usual fashion. A word $x \in A^*$ is said to be accepted by $\underset{\sim}{A}$ iff $M(q_0, x) \cap F \neq \phi$. $L(\underset{\sim}{A})$ denotes the set of words accepted by $\underset{\sim}{A}$. A language L is regular iff $L = L(\underset{\sim}{A})$ for some finite automaton $\underset{\sim}{A}$.

(Unrestricted) regular expressions (over the alphabet A) are defined inductively:

(a) Basis: If $a \in A$ then a is a regular expression denoting the language $\{a\}$; λ is a regular expression denoting the language $\{\lambda\}$; ϕ is a regular expression denoting the empty language ϕ .

(b) Induction: If α , β are regular expressions denoting the languages $L(\alpha)$, $L(\beta)$ respectively, then $\alpha\circ\beta$, $\alpha\cdot\beta$, $\overline{\alpha}$, α^* are regular expressions denoting the languages $L(\alpha)\circ L(\beta)$, $L(\alpha)\cdot L(\beta)$, $\overline{L(\alpha)}$, $(L(\alpha))^*$, respectively, where $\circ$ is any binary boolean function.

(c) Any regular expression can be obtained by a finite number of applications of (a) and (b).

We do not distinguish between boolean functions of different models for boolean algebras. For example, inthe above definition, in order to be precise, it would be necessary to say that $\circ$ in $\alpha\circ\beta$ is a boolean function in the model of regular expressions, that $\circ$ in $L(\alpha)\circ L(\beta)$ is a boolean function in the model of sets, that the two models are isomorphic as boolean algebras, and that the two functions are to be identified via this isomorphism. Besides the two models already mentioned we will also use the model $\{0, 1\}$. It should be clear from the context which model is actually referred to.

QUASIAUTOMATA AND THEIR RELATION TO FINITE AUTOMATA

In this section we introduce the notion of quasiautomaton which is a generalization of the notion of finite automaton. Then we will show that for each quasiautomaton there exists a deterministic finite automaton, called the derived deterministic automaton, which accepts the same language, thereby establishing that the language accepted by any quasiautomaton is regular.

We would like to mention at this point that we do not aim for utmost generality but rather try to formulate this section in order to simplify the presentation of the following sections.

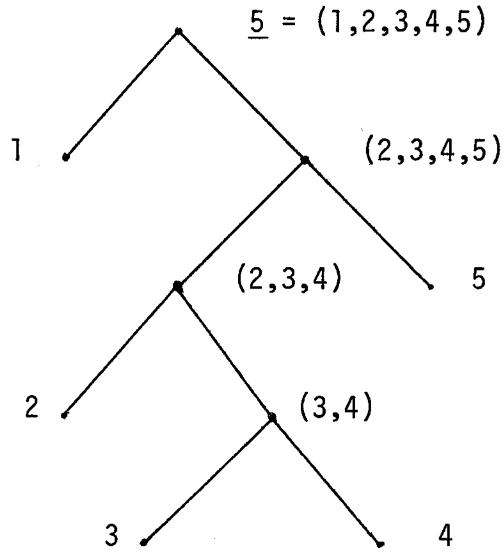
Given any natural number $c \geq 1$, we denote by $\underline{c}$ the ordered sequence $\underline{c} = (1,2,\ldots,c)$. Let $\tau_{\underline{c}}$ be an ordered binary tree such that the leaf profile reads $\underline{c}$. Furthermore let the label of each node n of $\tau_{\underline{c}}$ be the leaf profile of the subtree with roots n. One easily verifies: If a node n has n_1 (n_2) as its left (right) son, and s_i is the label of node n_i, $i = 1,2,$ then n has the label

$$s = s_1 \cup s_2$$

where $s_1 \cup s_2$ is the ordered sequence consisting of the elements of s_1 (in order) followed by the elements of s_2 (in order). (Note that $\cup$ here is not commutative.) Furthermore if a is the last element of s_1, b the first element of s_2 then $a+1 = b$. In particular, the root n_0 of $\tau_{\underline{c}}$ has the label $\underline{c} = (1,2,\ldots,c)$. It follows immediately that each label is of the form $(i, i+1, \ldots, i+d)$, $d \geq 0$.

Let $S_{\tau_{\underline{c}}}$ (or S if $\tau_{\underline{c}}$ is understood) be the set of labels of the tree $\tau_{\underline{c}}$; each $s \in S$ is called a type.

For example, let $c = 5$. Then $\tau_{\underline{c}}$ might be



The corresponding set S of labels is

$$\{1,2,3,4,5,\ (3,4),\ (2,3,4),\ (2,3,4,5),\ \underline{5}\} \ .$$

Let Q be a finite, nonempty set of states, $c \geq 1$, and assume

$$Q = Q_1 \cup \dots \cup Q_c$$

such that $Q_i \cap Q_j = \phi$ for all $i \neq j$, and $Q_i \neq \phi$ for all $i = 1,\dots,c$. For each $s \in S_{\tau_{\underline{c}}}$ we define

$$Q_s = \bigcup_{i \in s} Q_i \ .$$

(Note that the sequence s is treated here as a set; since s contains no element more than once, no confusion should arise.)

For instance, if $s = (2,3,4)$ $Q_s = Q_2 \cup Q_3 \cup Q_4$. For each $s \in S_{\tau_{\underline{c}}}$, q_0^s is a new state, not in Q_s , which will be called the initial state of Q_s .

For each $i \in \{1,\ldots,c\}$ let $F_i \subseteq Q_i \cup \{q_0^i\}$. Furthermore, for each $s \in S_{\tau_{\underline{c}}}$, s containing more than one element, fix F_s as follows:

If $s = s_1 \cup s_2$ then F_s is one of the following four sets:

$$F_{s_2} - \left\{q_0^{s_2}\right\}$$

$$(F_{s_1} \cup F_{s_2}) - \left\{q_0^{s_1},\ q_0^{s_2}\right\}$$

$$(F_{s_2} - \left\{q_0^{s_2}\right\}) \cup \left\{q_0^s\right\}$$

$$((F_{s_1} \cup F_{s_2}) - \left\{q_0^{s_1},\ q_0^{s_2}\right\}) \cup \left\{q_0^s\right\} .$$

Clearly, $F_s \subseteq Q_s \cup \left\{q_0^s\right\}$ for all $s \in S$.

We now define the set $\mathcal{E}_{Q_\tau}$ or $\mathcal{E}$ of well formed expressions. The variables of the expressions in $\mathcal{E}$ will be the elements of Q . Furthermore let BOP be the set consisting of the following boolean operators:

+ (addition), $\cdot$ (multiplication), $\bar{\ }$ (complement), and any other binary boolean operator, one might want to add.

Finally, BR is a set of auxiliary symbols,

$$BR = \{(,\ ),\ [,\ \{\} \cup \{]_q,\ ]'_q,\ \}_q \mid q \text{ an initial state for some } Q_s\}$$

Then $\mathcal{E}$ is defined as follows:

(a) Any boolean expression over Q_i is in $\mathcal{E}$, having type i, for $i \in \{1,\ldots,c\}$.

(b) If $f,g \in \mathcal{E}$ having types s, t, respectively, then $\overline{f}$ is in $\mathcal{E}$ having type s and $f \circ g$, $(f \circ g)$ are in $\mathcal{E}$ having type $s \cup t$ if $s \cup t \in S$, for $\circ \in BOP$.

(c) If $f \in \mathcal{E}$ having type s, and s' is such that s' is maximal with respect to $t = s \cup s' \in S$ (i.e. there is no s'' such that $t' = s \cup s'' \in S$ and s'' has more elements than s'; note that this uniquely determines s' and t!)† then

$$[f]_{q_0^{s'}}\ ,\quad [f]'_{q_0^{s'}}\ \text{ are in } \mathcal{E} \text{ having type } t\ .$$

(d) If $f \in \mathcal{E}$ having type s, then $\{f\}_{q_0^s}$ is in $\mathcal{E}$ having type s.

(e) Any element of $\mathcal{E}$ can be obtained in a finite number of applications of (a) through (d).

To continue with our example, let

$$Q_1 = \{A,B\},\quad Q_2 = \{C\}\ ,\quad Q_3 = \{D,E\},\quad Q_4 = \{F,G\},\quad Q_5 = \{H\}\ ,$$

and let X^s be the initial state of Q_s, $s \in S_{\tau_{\underline{5}}}$. Then the following expressions are in $\mathcal{E}$:

$$A + \overline{H}\ ,\quad \text{type} \quad (1,2,3,4,5);$$
$$[\overline{D} + E]_{X^4}\ ,\quad \text{type} \quad (3,4)\quad ;$$

---

† Alternatively, if n_s is the node of τ with label s, then n_s is the left son of the node n_t, and the right son of n_t is the node $n_{s'}$.

$\overline{[C\cdot D + E]'}_{X^5}$, type (2,3,4,5) ;

$\overline{[C]_{X(3,4)}}$, type (2,3,4) ;

$\{[C + G]_{X^5} \cdot \overline{H}\}_{X(2,3,4,5)}$, type (2,3,4,5) ;

$\overline{H}$, type 5 .

On the other hand, the following are not in $\mathcal{E}$:

$[A + B]_C$, $\{H\}_{X^s}$ for any $s \neq 5$,

$[H]_{X^s}$ for any $s \in S_{\tau_{\underline{5}}}$,

$[\overline{C}]'_{X^s}$ for any $s \neq (3,4)$.

We now define a relation $\equiv$ on $\mathcal{E}$ as follows:

(a) Expressions of different type are never related.

(b) The boolean operators maintain their usual properties, in particular $+$ and $\cdot$ are associative, commutative, distribute over each other, etc.

This implies that $\equiv$ restricted to boolean expressions is precisely equivalence of boolean expressions (boolean functions).

(c) $[f]_q + [g]_q \equiv [f+g]_q$, $[f]'_q + [g]'_q \equiv [f+g]'_q$, $\{f\}_{q'} + \{g\}_{q'} \equiv \{f+g\}_{q'}$, if the left hand side is defined.

(d) $\{\{f\}_{q'}\}_{q'} \equiv \{f\}_{q'}$.

It is clear that $\equiv$ is an equivalence relation since it is reflexive, symmetric, and transitive.

Define

$$\mathcal{F} = \mathcal{E}/\equiv \quad .$$

Any element f of $\mathcal{F}$ will be called a function, however for convenience, we will always write expressions. Furthermore we define the type of a function to be the type of any expression denoting this function.

Thus

$$\overline{[\bar{D} + E]_{X^4}} \cdot (\overline{[\bar{E}]_{X^4}} + \overline{[\bar{D}]_{X^4}}) \equiv \overline{[\bar{D} + E]_{X^4} + [\bar{E} + \bar{D}]_{X^4}}$$

$$\equiv \overline{[\bar{D} + E + \bar{E} + \bar{D}]_{X^4}} \equiv \overline{[1]_{X^4}} \ ;$$

note that $\overline{[1]_{X^4}} \not\equiv 0$.

For the sake of completeness, q_0^s is considered to be a boolean function of type s for all $s \in S$.

We are now in a position to define quasiautomata. A quasiautomaton $\underset{\sim}{Q}$ is a quintuple

$$\underset{\sim}{Q} = (A, Q_\tau, M, q_0, F_\tau) \quad .$$

A is the alphabet of input symbols.

Q_τ is the tree of states, defined as follows:

τ is a tree, $\tau = \tau_{\underline{c}}$ for some natural number c . $S = S_{\tau_{\underline{c}}}$ is the corresponding set of labels of $\tau_{\underline{c}}$. Q is the set of states of the quasiqutomaton, $Q = Q_1 \cup \ldots \cup Q_c$ as described above. For each $s \in S$, Q_s is defined as outlined above.

Also, for each Q_s there is a distinct initial state $q_0^s \notin Q$.

q_0 is the initial state of the quasiautomaton, $q_0 = q_0^{s_0}$ $(s_0 = \underline{c})$.

F_τ is the tree of final states, defined as outlined above.

M , finally, is the transition function. It is a function from

$$(Q \cup \bigcup_{s \in S} \{q_0^s\}) \times A \quad \text{into} \quad \mathcal{F}$$

defined as follows:

For all $q \in Q_s$, $s \in S$, $M(q, a)$ is a <u>boolean</u> function ($\not\ni q_0^s$) of type s, and $M(q_0^s, a)$ is a function of type s.

We extend M to

$$M : \mathcal{F} \times A^* \to \mathcal{F}$$

as follows:

(1) $M(f, \lambda) = f$ for all $f \in \mathcal{F}$, $M(q_0, \lambda) = q_0$.

(2) $M(f, a)$ for $f \in \mathcal{F}$ is defined as follows:

(a) If f is a boolean function, $f = f(q_1, \ldots, q_j)$ then

$$M(f, a) = f(M(q_1, a), \ldots, M(q_j, a)).$$

(b) If $f = \overline{g}$ then $M(f, a) = \overline{M(q, a)}$.

(c) If $f = f_1 \circ f_2$, $\circ$ a binary boolean operator in BOP, then

$$M(f, a) = M(f_1, a) \circ M(f_2, a).$$

(d) If $f = [g]_q$, and s is the type of g, then

$$M(f, a) = \begin{cases} [M(g, a)]_q & , \text{ if } g =_{F_s} 0 \\ ([M(g, a)]_q + M(q, a)) & , \text{if } g =_{F_s} 1 \end{cases}$$

(e) If $f = [g]'_q$, and s is the type of g, then

$$M(f,\ a) = \begin{cases} [M(g,\ a)]'_q & ,\ \text{if}\ \ g =_{F_s} 0 \\ ([M(g,\ a)]'_q + M(q,\ a)), & \text{if}\ \ g =_{F_s} 1 \end{cases} .$$

(f) If $f = \{g\}_q$, and s is the type of g , then

$$M(g,\ a) = \begin{cases} \{M(g,\ a)\}_q & ,\ \text{if}\ \ g =_{F_s} 0 \\ \{M(g,\ a) + \tilde{g}\}_q & ,\ \text{if}\ \ g =_{F_s} 1 \end{cases}$$

and $\tilde{g}$ is defined as follows:

$$\tilde{g} = \begin{cases} M(q,\ a) & \text{if}\quad M(q,\ a) \not\equiv \{g'\}_q \ \text{ for some }\ g' \\ g' & \text{if}\quad M(q,\ a) \equiv \{g'\}_q \ \text{ for some }\ g' \end{cases} .$$

(3) $M(f,\ xa) = M(M(f,\ x),\ a)$ for $f \in \mathcal{F}$, $x \in A^*$, $a \in A$.

$=_{F_s}$ is an equivalence relation on functions of type at most s. It is called evaluation under F_s and is defined as follows:

(a) If f is a boolean function of type at most s , i.e.

$$f = f(F_s;\ Q_s - F_s)$$

then $f =_{F_s} \alpha$ where $\alpha = f(1,\ldots,1;\ 0,\ldots,0)$ $(\alpha \in \{0,\ 1\})$.

(b) If $f = \overline{g}$ then

$$f =_{F_s} \begin{cases} 0\ ,\ \text{if}\ \ g =_{F_s} 1 \\ 1\ ,\ \text{if}\ \ g =_{F_s} 0 \end{cases} .$$

(c) If $f = f_1 \circ f_2$, $\circ$ a binary boolean operator in BOP, then

$f =_{F_s} \alpha$ where $\alpha = \alpha_1 \circ \alpha_2$ and $f_i =_{F_s} \alpha_i$, $i = 1,2$.

(d) If $f = [g]_q$ then $f =_{F_s} 0$.

(e) If $f = [g]'_q$ then $f =_{F_s} \alpha$ where $g =_{F_s} \alpha$.

(f) If $f = \{g\}_q$ then $f =_{F_s} \alpha$ where $g =_{F_s} \alpha$.

To illustrate these concepts, recall the quasiautomaton $\underset{\sim}{Q}_1$. Let us compute $M(X^3, 0100)$.

$$M(X^3, 0010) = M(M(M(M(X^3, 0), 1), 0), 0) \ .$$

$M(X^3, 0) = [A + B]'_{X^2}$ by definition of M ;

$M([A + B]'_{X^2}, 1) = [M(A + B, 1)]'_{X^2}$ since 1 is the type of $A+B$ and $A+B =_{F_1} 0$

$$= [B + \overline{B}]'_{X^2} \equiv [1]'_{X^2} \ ;$$

$M([1]'_{X^2}, 0) = ([M(1, 0)]'_{X^2} + M(X^2, 0))$ since $1 =_{F_1} 1$

$$= ([1]'_{X^2} + \{C\}_{X^2}) \ ;$$

$$M([1]'_{X^2} + \{C\}_{X^2}, 0) = ([M(1, 0)]'_{X^2} + \{C\}_{X^2}) + \{M(C, D)\}_{X^2}$$

$$= [1]'_{X^2} + \{C\}_{X^2} + \{\overline{C}\}_{X^2}$$

$$\equiv [1]'_{X^2} + \{C + \overline{C}\}_{X^2} \equiv [1]'_{X^2} + \{1\}_{X^2} \ .$$

Thus $M(X^3, 0100) = [1]'_{X^2} + \{1\}_{X^2}$.

Now we can define acceptance of a word $x \in A^*$ by a quasiautomation $\underset{\sim}{Q} = (A, Q_\tau, M, q_0, F_\tau)$:

$$x \in A^* \text{ is accepted by } \underset{\sim}{Q} \text{ iff } M(q_0, x) =_{F_{s_0}} 1 \ .$$

The set of words accepted by $\underset{\sim}{Q}$ is denoted by $L(\underset{\sim}{Q})$,

$$L(\underset{\sim}{Q}) = \{ x \in A^* \mid M(q_0, x) =_{F_{s_0}} 1 \} .$$

For example, λ is accepted by $\underset{\sim}{Q}_1$ since $F_{s_0} = F_{(1,2)} = \{X^3\}$, and X^3 is the initial state of $\underset{\sim}{Q}_1$. 0 is not accepted by $\underset{\sim}{Q}_1$ since $[A + B]'_{X^2} =_{F_{s_0}} 0$. 01 is accepted by $\underset{\sim}{Q}_1$, for $[1]'_{X^2} =_{F_{s_0}} 1$, similarly for 010 and 0100 . In fact, it should be clear that every word starting with 01 is accepted by $\underset{\sim}{Q}_1$ since for any $w \in A^*$, $M(X^3, 01w)$ will have an additive term $[1]'_{X^2}$, and

$$[1]'_{X^2} =_{F_{s_0}} 1 .$$

<u>Theorem 1</u> Every regular language is accepted by some quasiautomaton, and conversely, every quasiautomaton accepts a regular language.

<u>Proof</u> Let R be a regular language; we have to show there exists a quasiautomaton $\underset{\sim}{Q}$ such that $R = L(\underset{\sim}{Q})$.

Let $\underset{\sim}{A}$ be a deterministic finite automaton such that $\underset{\sim}{A}$ accepts R . Since R is regular such an automaton always exists; denoted it by $\underset{\sim}{A} = (A, Q, M, q_0, F)$. Define

$$\underset{\sim}{Q} = (A, P_\tau, N, p_0, G)$$

as follows: $c = 1$, τ is the (degenerate) tree with one node, $P = P_1 = Q$,

$$G = \begin{cases} F & , \text{ if } q_0 \notin F \\ F \cup \{p_0\} & , \text{ otherwise} \end{cases} .$$

Finally N is defined as follows:

for all $q \in Q$, $N(q, a) = M(q, a)$, and $N(p_0, a) = M(q_0, a)$.

It is easy to verify that, in fact,

$$L(\underset{\sim}{Q}) = L(\underset{\sim}{A}) .$$

This proves the first claim of the theorem.

The second claim will be shown in the following way: Given a quasiautomaton $\underset{\sim}{Q} = (A, Q_\tau, M, q_0, F_\tau)$ we will construct a deterministic finite automaton $\underset{\sim}{A}$ such that $L(\underset{\sim}{Q}) = L(\underset{\sim}{A})$. Clearly this implies that $L(\underset{\sim}{Q})$ is a regular language.

Define $\underset{\sim}{A} = (A, P, N, p_0, G)$ as follows:

$P = \{f \in \mathcal{F} \mid M(q_0, x) = f$ for some $x \in A^*\}$,

$G = \{p \in P \mid p =_{F_{s_0}} 1\}$,

$p_0 = q_0$, and

$N : P \times A \to P$ is defined as follow:

If $a \in A$ and $p \in P$, i.e. $M(q_0, x) = p$ for some $x \in A^*$ then

$$N(p, a) = M(p, a) = M(q_0, xa) .$$

We now have to verify that $\underset{\sim}{A}$ is indeed a finite automaton. However, this follows immediately, if we can prove that $\mathcal{F}$ is finite. This will be done below.

It remains to show that $L(\underset{\sim}{A}) = L(\underset{\sim}{Q})$. This is not hard to see, since $x \in L(\underset{\sim}{Q})$ iff $M(q_0, x) =_{F_{s_0}} 1$ iff $N(q_0, x) \in G$ iff $x \in L(\underset{\sim}{A})$. This concludes the proof. $\square$

The deterministic finite automaton which was constructed in the proof from a given quasiautomaton $\underset{\sim}{Q}$ will be called the derived deterministic automaton, denoted by $\underset{\sim}{A}_{\underset{\sim}{Q}}$.

<u>Lemma</u> $\mathcal{F}$ is finite.

<u>Proof</u> By induction on the height h of τ .

Basis: $h = 0$, i.e. $Q = Q_i$. Assume Q has n states. Thus there are at most 2^{n+2} functions of type i , since there are $n+1$ variables $(Q \cup \{q_0\})$ and a function is either a boolean function or a function of the type $\{g\}_{X^1}$ where g is a boolean function.

Induction step: Let s_1, s_2 be types in S_τ such that $s = s_1 \cup s_2 \in S_\tau$, and assume that there are finitely many functions of type s_i , $i = 1,2$. Now, every function of type s can be considered as a function of two variables x_1 and x_2 , where for x_i functions of type s_i can be substituted, $i = 1,2$. There are only finitely many possibilities to do this. Hence there are only finitely many functions of type s .

This shows that $\mathcal{F}$ is finite. □

Let us construct the derived automaton $\underset{\sim}{A}_{\underset{\sim}{Q}_1}$ for $\underset{\sim}{Q}_1$.

	0	1	$=F_{s_0}$
X^3	$[A + B]'_{X^2}$	$[\bar{B}]'_{X^2}$	1
$[A + B]'_{X^2}$	$[A + B]'_{X^2}$	$[1]'_{X^2}$	0
$[\bar{B}]'_{X^2}$	$[\bar{A} + \bar{B}]'_{X^2} + \{C\}_{X^2}$	$[\bar{B}]'_{X^2} + \{\bar{C}\}_{X^2}$	1
$[1]'_{X^2}$	$[1]'_{X^2} + \{C\}_{X^2}$	$[1]'_{X^2} + \{\bar{C}\}_{X^2}$	1
$[\bar{A} + \bar{B}]'_{X^2} + \{C\}_{X^2}$	$[\bar{A} + \bar{B}]'_{X^2} + \{1\}_{X^2}$	$[1]'_{X^2} + \{\bar{C}\}_{X^2}$	1
$[\bar{B}]'_{X^2} + \{\bar{C}\}_{X^2}$	$[\bar{A} + \bar{B}]'_{X^2} + \{1\}_{X^2}$	$[\bar{B}]'_{X^2} + \{1\}_{X^2}$	1
$[1]'_{X^2} + \{C\}_{X^2}$	$[1]'_{X^2} + \{1\}_{X^2}$	$[1]'_{X^2} + \{\bar{C}\}_{X^2}$	1
$[1]'_{X^2} + \{\bar{C}\}_{X^2}$	$[1]_{X^2} + \{C\}_{X^2}$	$[1]'_{X^2} + \{1\}_{X^2}$	1
$[A + B]'_{X^2} + \{1\}_{X^2}$	$[\bar{A} + \bar{B}]'_{X^2} + \{1\}_{X^2}$	$[1]'_{X^2} + \{1\}_{X^2}$	1
$[\bar{B}]'_{X^2} + \{1\}_{X^2}$	$[\bar{A} + \bar{B}]'_{X^2} + \{1\}_{X^2}$	$[\bar{B}]'_{X^2} + \{1\}_{X^2}$	1
$[1]'_{X^2} + \{1\}_{X^2}$	$[1]'_{X^2} + \{1\}_{X^2}$	$[1]'_{X^2} + \{1\}_{X^2}$	1

Obviously, as already remarked, this could have been shortened by using the observation that any function containing $[1]'_{X^2}$ as additive term evaluates under F_{s_0} to 1 ; similarly for $\{1\}'_{X^2}$. The reduced automaton is given by $\underset{\sim}{A}_0 = (\{0,1\}, \{1,2,3\}, M_0, 1, \{1,3\})$, M_0 defined by

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 1 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 3 & 3 \end{array} \; .$$

QUASIAUTOMATA ARE LINEARLY CLOSED UNDER REGULAR OPERATIONS

In this section we will show that the class of quasiautomata is linearly closed under all regular operations i.e. all boolean operations, concatenation, and star. By this we mean the following: Given an m-ary operation f , $m \geq 1$, and m quasiautomata $\underset{\sim}{Q}_i$, there exists a quasi-automaton $\underset{\sim}{Q}$ such that the following holds:

(1) $L(\underset{\sim}{Q}) = f(L(\underset{\sim}{Q}_1), \ldots, L(\underset{\sim}{Q}_m))$.

(2) If $\underset{\sim}{Q}_i$ has n_i states, $i = 1,\ldots,m$, then $\underset{\sim}{Q}$ has $O(n_1 + \ldots + n_m)$ states.

Clearly, if f is a regular operation (1) can be satisfied by constructing the derived deterministic automaton $\underset{\sim}{A}_{\underset{\sim}{Q}_i}$ for $\underset{\sim}{Q}_i$, $i = 1,\ldots,m$, and then applying standard constructions. However, the result of this approach will not satify (2), in general.

Theorem 2 The class of quasiautomata is linearly closed under all boolean operations, concatenation, and star.

Proof We start with boolean operations. Without loss of generality we consider only complement and binary boolean operations.

Complement: Let $\underset{\sim}{Q}' = (A, Q_\tau, M', q_0, F'_\tau)$ be a quasiautomaton. Define $\underset{\sim}{Q} = (A, Q_\tau, M, q_0, F_\tau)$ where F_τ is the same as F'_τ with the exception of F_{s_0} which is given by

$$F_{s_0} = \begin{cases} F'_{s_0} - \{q_0\} & \text{if} \quad q_0 \in F'_{s_0} \\ F'_{s_0} \cup \{q_0\} & \text{if} \quad q_0 \notin F'_{s_0} \end{cases},$$

and M is as follows: $M(q_0, a) = \overline{M'(q_0, a)}$ for all $a \in A$, and $M(q, a) = M'(q, a)$ for all $a \in A$, $q \neq q_0$.

Clearly, this defines a quasiautomaton. By the definition of acceptance by a quasiautomaton we have $x \in L(\underset{\sim}{Q})$ iff $x \notin L(\underset{\sim}{Q}')$ for all $x \neq \lambda$, and due to the definition of F_{s_0} we also have $\lambda \in L(\underset{\sim}{Q})$ iff $\lambda \in L(\underset{\sim}{Q}')$. Therefore $L(\underset{\sim}{Q}) = \overline{L(\underset{\sim}{Q}')}$ which proves the first requirement for linear closure. The second one is obviously fulfilled.

Binary boolean operations: Let $\underset{\sim}{Q}_i = (A, Q^i_{\tau_i}, M_i, q^i_0, F^i_{\tau^i})$ be a quasi-automaton, $i = 1,2$, and let $\circ$ be a binary boolean operation. Without loss of generality assume $\left(Q_{s^1_0} \cup \bigcup_{s^1 \in S_1} \left\{q_0^{s^1}\right\}\right) \cap \left(Q_{s^2_0} \cup \bigcup_{s^2 \in S_2} \left\{q_0^{s^2}\right\}\right) = \phi$.

We now define $\underset{\sim}{Q} = (A, Q_\tau, M, q_0, F_\tau)$:

Let $\tau_0 = \tau_{\underline{c}}$, and for simplicity assume that the leaf profile of τ_2 reads $(c+1, \ldots, c+d)$. Then $\tau = \tau_{\underline{c+d}}$ where the root (with label $\underline{c+d}$) has τ_1 as left and τ_2 as right subtree. Clearly $S = S_\tau \supseteq S_1 \cup S_2$, S_i being the set of labels of τ_i.

Q_τ is as follows: For all $s^i \in S_i$, $Q_{s^i} = Q^i_{s^i}$, $i = 1,2$, and for $s_0 = \underline{c+d}$, $Q_{s_0} = Q^1_{s^1_0} \cup Q^2_{s^2_0}$ (s^1_0 being $\underline{c}$, s^2_0 being $(c+1, \ldots, d)$).

F_τ is as follows: For all $s^i \in S_i$, $F_{s^i} = F^i_{s^i}$, $i = 1,2$, and

$$F_{s_0} = \begin{cases} \left(F^1_{s^1_0} \cup F^2_{s^2_0}\right) - \left\{q^1_0, q^2_0\right\} & , \text{ if } \alpha_1 \circ \alpha_2 = 0 \\ \left(F^1_{s^1_0} \cup F^2_{s^2_0} \cup \{q_0\}\right) - \left\{q^1_0, q^2_0\right\} & , \text{ if } \alpha_1 \circ \alpha_2 = 1 \end{cases} ,$$

$$\alpha_i = \begin{cases} 0 & , \text{ if } q^i_0 \notin F^i_{s^i_0} \\ 1 & , \text{ if } q^i_0 \in F^i_{s^i_0} \end{cases} , \qquad i = 1, 2 .$$

Finally, M is as follows: $M(q_0, a) = M_1(q^1_0, a) \circ M_2(q^2_0, a)$ for all $a \in A$, and $M(q^i, a) = M_i(q^i, a)$ for all $a \in A$, if $q^i \in Q^i_{s^i_0} \cup \bigcup_{s^i \in S_i} \left\{q^{s^i}_0\right\}$, $i = 1,2$. Clearly this defines a quasiautomaton.

We claim $L(\underset{\sim}{Q}) = L(\underset{\sim}{Q}_1) \circ L(\underset{\sim}{Q}_2)$. Again this follows immediately from the definition of acceptance and the fact that the two quasiautomata have no states in common; the definition of F_{s_0} ensures that $\lambda \in L(\underset{\sim}{Q})$ iff $\lambda \in L(\underset{\sim}{Q}_1) \circ L(\underset{\sim}{Q}_2)$. Therefore the first condition for linear closure is satisfied. As for the second, we observe that $\underset{\sim}{Q}$ has all the states of $\underset{\sim}{Q}_1$ and $\underset{\sim}{Q}_2$ plus a new initial state q_0, thus (2) is clearly satisfied.

We proceed with concatenation. As in the previous case for binary boolean operations, let $\underset{\sim}{Q}_1$ and $\underset{\sim}{Q}_2$ be quasiautomata with no states in common. We define

$$\underset{\sim}{Q} = (A, Q_\tau, M, q_0, F_\tau) .$$

τ and Q_τ are defined as in the previous case. For $s^i \in S_i$ we have $F_{s^i} = F^i_{s^i}$, $i = 1,2$, and

$$F_{s_o} = \begin{cases} F^2_{s_o^2} - \{q_o^2\} & \text{, if } q_o^2 \notin F^2_{s_o^2} \\ \left(F^2_{s_o^2} \cup F^1_{s_o^1}\right) - \{q_o^1, q_o^2\} & \text{, if } q_o^2 \in F^2_{s_o^2} \text{ and } q_o^1 \notin F^1_{s_o^1} \\ \left(F^2_{s_o^2} \cup F^1_{s_o^1} \cup \{q_o\}\right) - \{q_o^1, q_o^2\} & \text{, otherwise} \end{cases} .$$

For the definition of M we distinguish two cases, $\lambda \notin L(Q_2)$, and $\lambda \in L(Q_2)$:

(a) $q_o^2 \notin F^2_{s_o^2}$:

$$M(q_o, a) = \begin{cases} [M_1(q_o^1, a)]_{q_o^2} & \text{, if } q_o^1 \notin F^1_{s_o^1} \\ [M_1(q_o^1, a)]_{q_o^2} + M_2(q_o^2, a) & \text{, otherwise} \end{cases}$$

and $M(q^i, a) = M_i(q^i, a)$ for $q^i = Q^i_{s_o^i} \cup \bigcup_{s^i \in S_i} \left\{q_o^{s^i}\right\}$, $i = 1,2$.

(b) $q_o^2 \in F^2_{s_o^2}$:

$$M(q_o, a) = \begin{cases} [M_1(q_o^1, a)]'_{q_o^2} & \text{, if } q_o^1 \notin F^1_{s_o^1} \\ [M_1(q_o^1, a)]'_{q_o^2} + M_2(q^2, a) & \text{, otherwise} \end{cases}$$

and $M(q_o^i, a) = M_i(q^i, a)$ for $q^i \in Q^i_{s_o^i} \cup \bigcup_{s^i \in S_i} \left\{q_o^{s^i}\right\}$, $i = 1,2$.

It is easily verified that $\underset{\sim}{Q}$ is in fact a quasiautomaton.

We claim: $L(\underset{\sim}{Q}) = L(\underset{\sim}{Q}_1)L(\underset{\sim}{Q}_2)$.

We will prove this for the case $\lambda \notin L(\underset{\sim}{Q}_2)$, the other case is similar.

$w \in L(Q_1)L(Q_2)$ and $\lambda \notin L(Q_2)$ implies $w = w_1w_2$, $|w_2| \geq 1$, $w_i \in L(\underset{\sim}{Q}_i)$ for $i = 1,2$. $w_1 \in L(\underset{\sim}{Q}_1)$ implies $M_1(q_0^1, w_1) =_{F^1_{s^1_0}} 1$.

By definition, this implies $M(q_0, w_1) = [f]_{q_0^2} + g$ for some $f,g \in \mathcal{F}$, and if $w_2 = aw_3$, $M(q_0, w_1a) = [f']_{q_0^2} + M(q_0^2, a) + g'$ for some f',g' . Clearly $M_2(q_0^2, w_2) =_{F^2_{s^2_0}} 1$ and hence $M(q_0, w) =_{F_{s_0}} 1$ since $F^2_{s^2_0} - \{q_0^2\} \subseteq F_{s_0}$. Therefore $w \in L(\underset{\sim}{Q})$. Now assume $w \in L(\underset{\sim}{Q})$, i.e. $M(q_0, w) =_{F_{s_0}} 1$. This implies that $M(q_0, w) = [f]_{q_0^2} + g$ where $g =_{F^2_{s^2_0}} 1$. Therefore some prefix w_1 of $w = w_1w_2$ must be in $L(Q_1)$ such that $M_2(q_0^2, w_2) =_{F^2_{s^2_0}} 1$. This shows $w \in L(\underset{\sim}{Q}_1)L(\underset{\sim}{Q}_2)$. Thus, we verified the first requirement for linear closure under concatenation. The second one is again obvious.

Finally, we deal with the star. As in the case of complement, let

$$\underset{\sim}{Q}' = (A, Q_\tau, M', q_0, F'_\tau)$$

be a quasiautomaton. Define $\underset{\sim}{Q} = (A, Q_\tau, M, q_0, F_\tau)$ where F_τ is the same as F'_τ with the exception of F_{s_0} , $F_{s_0} = F'_{s_0} \cup \{q_0\}$, and M is as follows: $M(q_0, a) = \{M'(q_0, a)\}_{q_0}$ for all $a \in A$, and $M(q, a) = M'(q, a)$ for all $a \in A$, $q \neq q_0$. Again, this defines a quasiautomaton.

We claim: $L(\underset{\sim}{Q}) = (L(\underset{\sim}{Q}'))^*$. This follows in the same fashion as for concatenation . Furthermore, the number of states remains unchanged. Therefore quasiautomata are also linearly closed under star.

This concludes the proof of the theorem. □

<u>Example</u>: Let $\underset{\sim}{Q}_1$ and $\underset{\sim}{Q}_2$ be quasiautomata.

$\underset{\sim}{Q}_1 = (\{0,1\}, Q^1_{\tau^1}, M_1, X^2, F^1_{\tau^1})$ where τ^1 is the tree with one node, labelled 1, $Q^1_{s^1_0} = \{A,B\}$, $F^1_{s^1_0} = \{X^2, A,B\}$, and M_1 is given by

	0	1
X^2	A	B
A	$A+\overline{B}$	$\overline{A}+B$
B	A	$\overline{B}$

.

$\underset{\sim}{Q}_2 = (\{0,1\}, Q^2_{\tau^2}, M_2, X^3, F^2_{\tau^2})$ where τ^2 is the tree with one node, labelled 2, $Q^2_{s^2_0} = \{C,D\}$, $F^2_{s^2_0} = \phi$, and M_2 is given by

	0	1
X^3	C	$\overline{C}$
C	D	$\overline{D}$
D	$C+D$	$\overline{C+D}$

.

We construct a quasiautomaton $\underset{\sim}{Q}$ such that

$$L(\underset{\sim}{Q}) = L(\underset{\sim}{Q}_1)\cdot L(\underset{\sim}{Q}_2)$$

$\underset{\sim}{Q} = (\{0,1\}, Q_\tau, M, X^1, F_\tau)$ where τ is the tree with three nodes, the root labelled $(1, 2)$, its left son 1, and its right son 2.

$$Q_1 = \{A,B\}\ , \quad Q_2 = \{C,D\}\ , \quad Q_{1,2} = \{A,B,C,D\}\ .$$

The initial nodes of $Q_1, Q_2, Q_{1,2}$ are X^2, X^3, X^1 , respectively. $F = \{X^2, A, B\}$, $F_2 = \phi$, $F_{1,2} = \phi$. Finally, M is given by

	0	1
X^1	$[A]_{X^3}+C$	$[B]_{X^3}+\overline{C}$
X^2	A	B
A	$A+\overline{B}$	$\overline{A}+B$
B	A	$\overline{B}$
X^3	C	$\overline{C}$
C	D	$\overline{D}$
D	$C+D$	$\overline{C+D}$

By constructing the derived deterministic automata $\underset{\sim}{A}_{\underset{\sim}{Q}_1}$, $\underset{\sim}{A}_{\underset{\sim}{Q}_2}$, and $\underset{\sim}{A}_{\underset{\sim}{Q}}$ one can verify the result directly i.e. without referring to the theorem.

REGULAR EXPRESSIONS AND DETERMINISTIC AUTOMATA: A BOUND ON THE NUMBER OF STATES

In this section we apply the results of the previous ones to solve the following question: Given an (unrestricted) regular expression, is there a bound on the number of states a deterministic automaton must have which accepts the language denoted by the given expression?

We define the function s from the set of regular expressions to the set of natural numbers - s is sometimes called the "letter content" of an expression and gives a measure for the "size" of the expression:

(a) $s(\beta) = 1$ for $\beta \in A \cup \{\lambda, \phi\}$

(b) $s(\alpha \circ \beta) = s(\alpha) + s(\beta)$ where α , β are regular expressions and $\circ$ is a binary boolean operator;

similarly for concatenation,

$s(\alpha \cdot \beta) = s(\alpha) + s(\beta)$

(c) $s(\overline{\alpha}) = s(\alpha^*) = s(\alpha)$ for α a regular expression.

Our aim is to construct a quasiautomaton $\underset{\sim}{Q}_\alpha$ from a given regular expression α . This will be done by structural induction on α . It should be obvious that the induction step is precisely the construction given in the proof of the theorem in the last section. All that remains is to give a basis. This is rather trivial. Let τ be the tree with one node. Construct $\underset{\sim}{Q}_\phi, \underset{\sim}{Q}_\lambda, \underset{\sim}{Q}_a$ accepting ϕ , $\{\lambda\}$, $\{a\}$:

$\underset{\sim}{Q}_\phi = (A, Q_\tau, M_\phi, X, \phi)$, $Q_{s_0} = \{q\}$, $M_\phi(X, a) = q$ and

$M_\phi(q, a) = 0$ for all $a \in A$.

$$\underset{\sim}{Q}_\lambda = (A, Q_\tau, M_\lambda, X, F_\tau) \ , \quad Q_{s_0} = \{q\} \ , \quad M_\lambda(X, a) = q \quad \text{and}$$

$$M_\lambda(q, a) = 0 \quad \text{for all} \quad a \in A \ , \ F_{s_0} = \{X\} \ .$$

$$\underset{\sim}{Q}_a = (A, Q_\tau, M_a, X, F_\tau) \ , \quad Q_{s_0} = \{q\} \quad , \ M_a(X, a) = q \ ,$$

$$M_a(X, b) = 0 \quad \text{for all} \quad b \in A - \{a\} \ , \quad M(q, c) = 0$$

$$\text{for all} \quad c \in A \ , \quad F_{s_0} = \{q\} \quad .$$

(0 in the definition of the transition functions denotes the constant boolean function 0 .)

Therefore, given a regular expression α there exists a quasiautomaton $\underset{\sim}{Q}_\alpha$ with $2 \cdot s(\alpha)$ states such that $L(\underset{\sim}{Q}_\alpha) = L(\alpha)$.

Now recall the proof that $\mathcal{F}$ is finite. The crucial property is that a function of type $s = s_1 \cup s_2 \in S$ can be considered as a function of two variables x_1, x_2 where functions of type s_i may be substituted for x_i , $i = 1,2$. Since the above construction imposes certain restrictions on the quasiautomata one obtains, it is easily verified that there are $2^{(2^2)} \cdot N_{s_1} \cdot N_{s_2}$ functions of type s $(= s_1 \cup s_2)$ where N_{s_i} denotes the number of functions of type s_i , $i = 1,2$. Thus, it follows by induction that the derived deterministic automaton $\underset{\sim}{A}_{\underset{\sim}{Q}_\alpha}$ accepting $L(\alpha)$ has at most

$$[2^{(2^2)}]^{s(\alpha)-1} \cdot [2^{(2^1)}]^{s(\alpha)} + 1$$

states, or letting $s(\alpha) = n$ we get the bound

$$2^{4n-4} \cdot 2^{2n} + 1 = 2^{6n-4} + 1$$

(The "+1" comes from the fact that the only initial state ever appearing as state of $\underset{\sim}{A}_{\underset{\sim}{Q}_\alpha}$ is the initial state of $\underset{\sim}{Q}_\alpha$.)

Since the reduced automaton accepting $L(\alpha)$ cannot have fewer states than $\underset{\sim}{A}_{\underset{\sim}{Q}_\alpha}$ we can summarize:

<u>Theorem 3</u> Given an (unrestricted) regular expression α , the reduced automaton accepting α has at most

$$2^{6s(\alpha)-4} + 1$$

states. □

Acknowledgement

I am indebted to Professor J.A. Brzozowski for several discussions which helped me in the formulation of the results reported.