UNIVERSITY OF NORTH CAROLINA
Department of Statistics
Chapel Hill, N. C.

Mathematical Sciences Directorate
Air Force Office of Scientific Research
Washington 25, D. C.

AFOSR Report No. 893

A REMARK ON FINITE TRANSDUCERS

by

M. P. Schützenberger

University of North Carolina

June, 1961
Contract No. AF 49(638)-213

Institute of Statistics
Mimeograph Series No. 289

A remark on finite transducers

M. P. Schützenberger

I. Introduction.

In this note we consider a very restricted class of transducers, i.e. of automata which transform finite input words into finite output words (CF. 6). The simplest case is the transformation consisting in the replacement of every input letter x by an output word $\eta(x)$ which is eventually the empty word e_Y. Algebraically, since the set F_X (F_Y) of all finite input (output) words is the free monoid[1] generated by the input alphabet $X = \{x\}$ (the output alphabet $Y = \{y\}$), this transformation is simply an homomorphism $\eta: F_X \to F_Y$.

If η is such that $\eta(f) = \eta(f')$ only if $f = f'$ it is called an encoding (with unique decipherability) and, then, η is an isomorphism.

Next in simplicity are the transformations realized by a conventional (one way, one tape[8]) automaton supplemented by a printing device. Upon reading x on the input tape and, accordingly, going from the state s to the state $s'=sx$, a word $\eta(s;x)$ function of s and x only is printed on the output tape which is moved the corresponding length. Trivially, any mapping from F_X to F_Y can be performed by a transformation of this type if no restriction is imposed on the number of states. We shall always assume here that $S = \{s\}$ is a finite set. This imposes drastic restrictions on η and, in particular, it introduces a difference between the right transformations (where reading and printing are done from left to right) and the left transformations (where both operations are done in the opposite direction). For example no (finite) right automaton can perform the task of reproducing the input word when it ends with a given letter and of printing nothing when it does not.

Consequently the composite operation which consist of transforming first the input word by a right automaton and, then, of transforming again the output word by a left automaton cannot as a rule be carried out in a single pass; we shall call it a transduction and we shall describe some of its elementary properties:

1. The transductions form a set closed by finite composition and also by inversion when this last operation has a meaning.

2. The transductions transform regular events[5] on the input words into regular events on the output words and any regular event can be obtained in this manner.

These two properties indicate that there is no difference between the languages which can be accepted by finite automata and the languages which can be produced by any <u>bounded</u> number of finite automata; here, the boundedness condition cannot be omitted as it is easily shown by Chomsky's counterexamples (Cf. 2).

For notational reasons it is more convenient to define a transduction η with sets of states S, S' as the transformation from an input word $f=x_1x_2\ldots x_n$ and a pair of states $s_1 \in S$, $s'_1 \in S'$ to an output word that is obtained by replacing every letter x_i by a fixed output word $\eta(s_i;x_i;s'_{n-i+1})$ where the states are given inductively by the equations $s_{j+1}=s_jx_j$ and $s'_{n-j+2} = x_j\, s'_{n-j+1}$. With this definition, right (left) transductions correspond to the special case where $\eta(s_i;x;s')$ does not depend effectively upon its left (right) argument and where, consequently, S' (S) can be taken as reduced to a single state and, finally, omitted.

The finite closure property 1. shows that this new construct is equivalent to the composition of a right and of a left transduction; encodings correspond to the case where S and S' reduce to a single state and, then, the property 1 shows that the deciphering can always be performed by a transduction.

Example. Let $X = x_1, x_2$ and $Y = y_1, y_2, y_3$. Every input word has a unique factorization $f = x_i^{n_1} x_i^{n_2} \ldots.$ into runs $x_j^{n_k}$ consisting of the same letter x_j repeated n_k times and we suppose that we want to perform the transformation η which lets invariant the runs of even length and replaces every run of odd length by y_3.

Thus, for example,

$$/ \quad \eta\, x_1^3 x_2^2 x_1 x_2^3 x_1^4 = y_3 x_2^2 y_3^2 x_1^4 \quad .$$

This can be realized if for any factorization $f = f'xf''$ we follow the two instructions:

Print out x if it belongs to a run of even length;

Print out y_3 or nothing when x belongs to a run of odd length according to x is or is not the last letter of this run.

In order to carry them out it is enough to know that f' and f" are respectively ending and beginning by runs of length $n'>0$ $n''>0$ in the letters x' and x'' because:

x belongs to a run of even length if $x'=x=x''$ and n' and n'' have different parity or if $x' = x \neq x''$ and n' is odd or if $x' \neq x = x''$ and n'' is odd;

x is the last letter of a run of odd length if $x \neq x''$ and n' is even or if $x' \neq x \neq x''$.

Consequently, all that is needed is the parity of n' and n" and the last and first letter respectively of f' and f". As we shall see below these informations can be supplied by twofinite state automata, one having read f' from left to right and the other one having read f" in the opposite direction.

Let us now consider how this transformation could be achieved in <u>two</u> passes. The first one is performed by a right transduction with states s_i $(0> i < 4)$, initial state s_o and transitions:

$s_o x_1 = s_2 x_1 = s_3 x_1 = s_4 x_1 = s_1,$

$s_1 x_1 = s_2,$

$s_o x_2 = s_1 x_2 = s_2 x_2 = s_4 x_2 = s_3,$

$s_3 x_2 = s_4$.

Thus for any input word f' the last state reached, s_j, has index of the same parity as the last run of f' and $j \leq 2$ if and only if the last letter of f' is x_1.

The machine has an output alphabet $Z = z_i$ $(1 \leq i \leq 4)$ with the printing rule $\eta'(s_i;x_j) = z_{i''}$ when $s_i x_j = s_{i''}$.

For example, $\eta'(x_1^3\ x_2^2\ x_1\ x_2^3\ x_1^4\ ) = z_1 z_2 z_1 z_3 z_4 z_1 z_3 z_4 z_3 z_1 z_2 z_1 z_2 = \eta' f$.

The second pass is performed by a left transducer with states s'_i $(0 \leq i \leq 4)$, initial state s'_o and transitions:

$z_1 s'_i = s'_1$ if $i \neq 2$ and $z_1 s'_2 = s'_2$; $z_2 s'_i = s'_2$ if $i \neq 1$ and $z_2 s'_1 = s'_1$;

$z_3 s'_i = s'_3$ if $i \neq 4$ and $z_3 s'_4 = s'_4$; $z_4 s'_i = s'_4$ if $i \neq 3$ and $z_4 s'_3 = s'_3$.

The printing rule is given by

$\eta''(z_i;s'_j) = x_1 x_1$ when $i = 2$ and $j \neq 1$; $= x_2 x_2$ when $i = 4$ and $j \neq 3$; $= y_3$ when $i = 1$ and $j = 0,2,4$ or $i = 3$ and $j = 0,1,2$; $= e_Y$ (nothing) in all other cases.

For example, $\eta''(\eta' f) = e_Y e_Y y_3 e_Y x_2 x_2 y_3 e_Y e_Y y_3 e_Y x_1 x_1 e_Y x_1 x_1$, that is $\eta(s_o;f;s'_o)$.

II. <u>Formal definition and Nerode's Theorem</u>.

A transduction η is given by the following structures:

1. A finite input alphabet $X = x$ and an output alphabet $Y = y$;

2. Two finite sets of states $S = S$ and $S' = s'$;

3. Two mappings $(S,X) \to S$ and $(X,S') \to S'$ written respectively sx and xs';

4. A mapping $\eta: (S,X,S') \to F_Y$ written $\eta(s;x;s')$.

These mappings are extended in a natural fashion to any $f \in F_X$ by the following inductive rules:

$se_X = s$ and $e_X s' = s'$, $\eta(s;e_X;s') = e_Y$ for any $(s,s') \in (S,S')$;

for any $f \in F_X$, $x \in X$, $(s,s') \in (S,S')$: $s(fx) = (sf)x$, $(fx)s' = f(xs')$,

$\eta(s;fx;s') = \eta(s;f;xs')\eta(sf;x;s')$.

It is easily checked that these rules are equivalent to the ones given in the introduction. By induction the last rules gives the following identity which could be taken as a definition and which displays η as a two-sided coset mapping $F_X \to F_Y$:

for any $f_1, f_2, f_3 \in F_X$

$\eta(s;f_1f_2f_3;s') = \eta(s;f_1;f_2f_3s')\ \eta(sf_1;f_2;f_3s')\ \eta(sf_1f_2;f_3;s')$.

In a more concrete manner η can be realized by finite matrices whose entries belong to the union of F_Y and of a zero 0.

Indeed for any $x \in X$ let μx be a square matrix whose rows and columns are indexed by the pairs $(s_i, s'_{i'}) \in (S,S')$ and whose entries are $\mu x((s_i,s'_{i'}),(s_j,s'_{j'})) = \eta(s_i;x;s'_{j'})$ if $s_i x = s_j$ and $s'_{i'} = x\, s'_{j'}$

$= 0$, otherwise.

Then if $f = x_1 x_2 \ldots x_n$ the corresponding output word $\eta(s;f;s')$ is equal to the entry $\mu f((s,fs'),(sf,s'))$ of $\mu f = \mu x_1\, \mu x_2 \ldots \mu x_n$.

Proof. For any $f \in F_X$ and $x \in X$ we have

$\mu fx((s_i,s'_{i'}),(s_j,s'_{j'})) = \sum_{k,} (\mu f((s_i,s'_{i'}),(s_k,s'_{k'})))(\mu x((s_k,s'_{k'}),(s_j,s'_{j'})))$

where the summation is over all the pairs $(s_k,s'_{k'}) \in (S,S')$. The only non zero term in the sum is the one corresponding to the pair defined by the equations $s_k = s_i\, f$ and $s'_{i'} = x\, s'_{j'}$; we have then $s_j = s_k x$ and

$s'_{i'} = fs'_{k'}$, that is $s_j = s_i fx$ and $s'_{i'} = fxs'_{j'}$. Thus the entry under consideration is equal to $\eta(s_i;f;xs'_{j'})\ \eta(s_i f;x;s'_{j'})$, that is, to $\eta(s_i;fx;s'_{j'})$ and the result follows by induction.

Example. Let

$X = a,b$; $Y = c,d$; $S = s_1,s_2$; $S' = t_1,t_2$;

$s_1 a = s_2 a = s_2 b = s_1$; $s_1 b = s_2$; $bt_1 = bt_2 = at_2 = t_1$; $at_1 = t_2$.

$\eta(s_i;x;t_j) = cc$ if $x=a$ and $i=j$; $= d$ if $x=a$ and $i \neq j$ or if $x=b$ and $1=i \neq j$;
$=c$ if $x=b$ and $i=j=2$; $= e_Y$ in all other cases.

Then for instance $\eta(s_1;bbab;t_1) = ccc$ according to the following self explanatory scheme

$$\begin{matrix} s_1 & b(e_Y) & s_2 & b(c) & s_1 & a(cc) & s_1 & b(e_Y) & s_2 \\ t_1 & & t_1 & & t_2 & & t_1 & & t_1 \end{matrix}$$

Also we have

$$\mu a = \begin{matrix} 0 & d & 0 & 0 \\ cc & 0 & 0 & 0 \\ 0 & cc & 0 & 0 \\ d & 0 & 0 & 0 \end{matrix} \qquad \mu b = \begin{matrix} 0 & 0 & e_Y & d \\ 0_0 & 0 & 0 & 0 \\ 0_Y & c & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \qquad \mu bbab = \begin{matrix} 0 & 0 & ccc & cccd \\ 0 & 0 & 0 & 0 \\ 0 & 0 & dd & ddd \\ 0 & 0 & 0 & 0 \end{matrix}$$

and $\eta(s_1;bbab;t_1)$ is equal to $\mu bbab\ ((s_1,t_1),(s_2,t_1))$.

As an immediate consequence of the definitions we derive the following weak form of Nerode's ultimate periodicity theorem

There exist finite integers m and n which are such that for any $f,f',f'' \in F_X$, $(s,s') \in (S,S')$, $p,r \geq 0$, and $r \leq n$ one has

$\eta(s;f'f^{2m+pn+r}f'';s') = g'g^p g''$ where $g,g',g'' \in F_Y$ do not depend on p.

Proof. Since S and S' are finite we can find integers m and n such that for all $(s,s') \in (S,S')$, $f \in F_X$, $p \geq 0$, $0 \leq r \leq n$ we have :

$sf^{m+pn+r} = sf^{m+r}$, $f^{m+pn+r}s' = f^{m+r}s'$. Thus $\eta(s;f'f^{2m+pn+r}f'';s') =$

$\eta(s;f'f^{m+r};f^{pn+m}f''s')\ \eta(sf'f^{m+r};f^{pn};f^{m+r}f''s')\ \eta(sf'f^{m+pn+r};f^m f'';s')$.

Because of our choice of m and n the second factor is equal to p times the word $g = \eta(sf'f^{m+r};f^{n};f^{m}f''s')$ and the result is proved.

III. Finite closure properties.

III.1. To any two tranductions η: $(S;F_X;S') \to F_Y$ and ξ: $(T;F_Y;T') \to F_Z$ there corresponds a transduction $\pi : (R;F_X;R:) \to F_Z$ which is such that for any $f \in F_X$, $(s,s') \in (S,S')$, $(t,t') \in (T,T')$ one has identically $\xi(t;\eta(s;f;'');t') = \pi(r;f;r')$ where the states $r \in R$ and $r' \in R'$ are functions of s and t and of s' and t' respectively.

Proof. We define an equivalence relation σ on F_X by the following rules: $\sigma f = \sigma f'$ (to be read : the σ-class of f is the same as that of f') if and only if

1. for all $s \in S$, $sf = sf'$;

2. for all $(s,s') \in (S,S')$ and $t \in T$, $t\eta(s;f;s') = t\eta(s;f';s')$.

The relation σ has at most $S^{S} \times T^{T \times S \times S'}$ (S : the number of states in S) distinct classes. Furthermore it is right regular (i.e. $\sigma f = \sigma f'$ implies $\sigma ff'' = \sigma f'f''$ for all f") since when $\sigma f = \sigma f'$ we have

1. $sff'' = sf'f''$ for any $s \in S$(because $sf=sf'$);

2. for any $(s,s') \in (S,S')$ and $t \in T$, $t\eta(s;ff'';s') = t\eta(s;f;f''s')\eta(sf;f'';s') = t\eta(s;f';f''s')\eta(sf';f'';s') = t\eta(s;f'f'';s')$. (because

We now define R as the set of all triples $r = (s,t,\sigma f)$ and the mapping $(R,X) \to R$ by $(s,t,\sigma f)x = (s,t, \sigma fx)$.

In a perfectly symmetric manner we construct a left regular equivalence σ', a set of states $R' = r' (s',t^{8},\sigma'f)$ and a mapping $(X,R') \to R'$.

Finally we put $\pi((s,t,\sigma f);x;(s';t';\sigma'f')) =$

$\xi(t\eta(s;f;xf's');\eta(sf;s;f's');\eta(sfx;f';s')t')$.

This definition is free from ambiguity because the three expressions

$\eta(\;;\;;\;)$ entering in it depend only upon the classes σf and $\sigma' f'$; this is a direct consequence of the definition of σ and σ' and it concludes the proof since it is enough now to check by developing the expressions that if $f=f'xf''$ we have $\xi(t;\eta(sf$

$\xi(t;\eta(s;f;s');t') = \pi(r;f;r')$ where $r = (s,t,\sigma e_X)$ and $r' = (s',t',\sigma' e_X)$.

Before verifying the second closure properties we recall the following facts:

1. Let R_Z denote the family of the subsets $F' \subset F_Z$ that are regular events in the sense of S. C. Kleene[5]. The given of an $F' \in R_Z$ is equivalent (Cf. 10) to that of an homomorphism $\gamma : F_Z \rightarrow P$ where P is a finite monoid together with the subset P' of P associated to F' by the relations $\gamma F' = P'$; $F' = \gamma^{-1} P'$ (= $f : \gamma f \in P'$). The equivalence on F_Z defined by $\gamma f = \gamma f'$ is at the same time left and right regular and it has only finitely many classes.

2. According to D. Huffman's theory[4] the transformation η can be said to be <u>information lossless</u> on the subset F' if the equations $\eta(s;f;s') = \eta(s;f';s')$, $sf=sf'$; $fs' = f's'$, $f,f' \in F'$ imply $f=f'$.

III. 2. If η is information lossless on the subset $F' \in R_X$ there exists a transduction ξ ($= \eta^{-1}$) which is such that for any $f \in F'$, $(s,s') \in (S,S')$ we have $\xi(t;\eta(s;f;s');t') = f$ where the states $t \in T$ and $t' \in T'$ are functions of s and fs' and of s' and sf respectively.

<u>Proof</u>. Let H be the set of all the words $\eta(s;x;s')$ with $x \in X$, and K (K') the set of all proper right (left) factors of the words of H (i.e. $k \in K$ if and only if $kf \in H$ for some $f \neq e_Y$).

If σ is a right regular equivalence on F_X with σ classes we say that $g \in F_Y$ admits a factorization of type $(\sigma f, s_i, s'_j, \;, s_j, s'_i, \;, k)$

(where σf is any σ-class, $s_i, s_j \in S$, $s'_{i'}, s'_{j'} \in S'$, $k \in k$) if there exist $f \in F_X$ and $g' \in F_Y$ such that the following relations are satisfied: $\sigma f = \sigma f'; g = g'k$; $g' = \eta(s_i; f'; s'_{i'})$; $s_i f' = s_j$; $f's'_{i'} = s'_{j'}$:
Clearly, if h is the maximal length of an element of H, there exists at most $\sigma \times (S \times S')^2 \times h$ different types of factorization. Thus if we write $\lambda g_1 = \lambda g_2$ when the elements $g_1, g_2 \in F_Y$ admit exactly the same set of types of factorization, the relation λ has only finitely many classes when the same is true of σ and, by construction, λ is right regular.
In perfectly symmetric manner we associate a left regular equivalence λ' on F_Y to any left regular σ' on F_X.
We now come to the construction of ξ . As indicated above we construct the relations λ and λ' on F_Y associated with the (left and right regular) relation γ on F_X used for the definition of F' and we define T as the set of all triples $(s, s', \lambda g)$ with $(s, s') \in (S, S')$ and λg a λ-class; the mapping $(T, Y)\, T$ is given by $(s, s', \lambda g)y = (s, s', \lambda gy)$. The set of states T' and the mapping $(Y, T') \to T'$ are defined in symmetric manner with the help of the relation λ'.

For each triple $(s, s', x \in X)$ such that $\eta(s; x; s') \neq e_Y$ we select arbitrarily one factorization kyk' of $\eta(s; x; s')$ and we define ξ by the following rules:
$\xi((s_1, s'_4, \lambda g\lambda); y; (s_4, s'_1, \lambda' g')) \Rightarrow x$ if there exists $\bar{g}, \bar{g}' \in F_Y$; $k \in K$; $k' \in K'$; $f, f' \in F_X$; $s_2, s_3 \in S$; s'_2 , $s'_3 \in S'$ satisfying the following relations:
$\lambda g = \lambda \bar{g} k$; $\lambda' g' = \lambda' k' \bar{g}'$;
$\bar{g} = \eta(s_1; f; s'_3)$; $s_1 f = s_2$; $fs'_3 = s'_4$; $s_2 x = s_3$;
$\bar{g}' = \eta(s_3; f'; s'_1)$; $s_3 f = s_4$; $f's'_1 = s'_2$; $xs'_2 = s'_3$;
kyk' is the selected factorization of $\eta(s_2; x; s'_3)$;
fxf' belongs to F'.

In all other cases the value of $\xi(\quad)$ above is e_Y .

The possibility of solving all except the last of the above equations for given $y \in Y$, $\lambda g, \lambda' g'$, s_1, s_4, s_1', s_4' is a direct consequence of the definition of λ and λ' . Taken together these equations imply that there exists at least one triple $f, x, f' \in F_X$ for which one has :

$\eta(s_1; fxf'; s_1') = \bar{g}kyk'\bar{g}' = g''$ with a selected factorization;

$s_1 fxf' = s_4$ and $fxf's_{11}' = s_4'$;

$fxf' \in F'$.

Thus if the word $g'' = \bar{g}kyk'\bar{g}'$ has been obtained from a word f'' in F' by a transduction with the indicated initial and final states, it follows from III.1 that $\xi(t; g''; t')$ will be identical to f'' and, because of the hypothesis that η is information lossless on F' this proves a posteriori that the above equations have a unique solution.

Remark. Because of the assumption that S' is finite it is always possible to realize in a single pass any arbitrary transduction if one is allowed to use a bounded number of output tapes and if one has the possibility of erasing on them.

Since the general case is rather cumbersome it may be enough to restrict ourself to the detailed examination of the procedure needed for the deciphering of an encoding.

Thus, let us assume now that S and S' are reduced to a single element and that consequently η is an isomorphism $F_X \rightarrow F_Y$. The sets H and K have the same meaning as in the last previous section and $P = \eta F_X$ is the submonoid of F_Y generated by H.

To any $g \in F_Y$ we associate the set λg of those $k \in K$ which are such that $g = pk$ for some $p \in P$; λg contains at most h elements and, consequently, the equivalence relation on F_Y defined by $\lambda g = \lambda g'$ has only finitely many classes; since, furthermore, it is right regular we can construct a conventional automaton whose states are identified with the various possible λg's and whose transitions are given by $(\lambda g)y = \lambda(gy)$. We still observe that for any $g \in F_Y$ either λg is empty (and in this case g cannot be a left factor of a word in P) or, if $k \in \lambda g$ there exists a uniquely determined element $f = \xi p \in F_X$ such that $g = (\eta f)k = pk$. Let us now consider a word $g'' \in P$ and any factorization $g'' = gyg'$ of it; let us assume also that we have been able to record on λg tapes the words ξp_i corresponding to the λg elements $k_i \in \lambda g$. The automaton is in state λg and upon reading the letter y it will go to the state $\lambda(gy)$. For each $k_i \in \lambda g$ four cases are possible and we list below the printing instructions to be followed in each of them:

1. $k_i y$ does not belong to H nor to K (i.e. $k_i y$ cannot be a left factor of a word of P); then the machine erases the corresponding word ξp_i .
2. $k_i y$ belongs to H and not to K; the machine writes on the corresponding tape the letter $x \in X$ such that $\eta x = k_i y$; thus, on this tape we now have $(\xi p_i)x$.
3. $k_i y$ belongs to K and not to H; the machine does nothing on the corresponding tape.
4. $k_i y$ belongs to H and to K; the machine does as in 2. above but also it takes a new tape and it reproduces on it the word ξp_i . This new tape corresponds to the element $k_i y \in \lambda(gy)$.

At the end of the reading of g'', $\lambda g''$ contains e_Y because, by hypothesis, $g''\pi \in P$ and the corresponding tape carries the word $\xi g''$ such that $\eta(\xi g'') = g''$.

It is clear that, at any given stage of the procedure only h tapes at most are needed since we can use the tapes made free by the operation 1. above.

The proof of the validity of the algorithm is left to the reader and in figures 1. and 2. we give a complete account of the constructinn of the state diagram and of the deciphering of the word $a^4ba^4b^2a^2$ for the following example:

$X = x_i$ ($1 \leq i \leq 5$) ; $Y = a,b$;

$\eta x_1 = aa$; $\eta x_2 = baa$; $\eta x_3 = bb$; $\eta x_4 = ba$; $\eta x_5 = bb$; $K = e_Y,a,b,ba,bb$.
(This is an encoding because it is a left prefix code in the three words: u=a;v=ba;w=bb

We find $\xi(a^4b\ a^4b^2a^4 = x_1^2x_2x_1x_3x_1$.

IV. Relationships with regular events.

As we shall deal here with fixed initial states, we write for any subset F' (G') of F_X (F_Y) :

$\eta F' = g\epsilon F_Y : g=\eta(s_1;f;s_1') \ f\epsilon F'$; $\eta^{-1}G' = f\epsilon F_X : \eta(s_1;f;s_1')=\epsilon\ G'$.

IV.1. The subset G' pf ηF_X belongs to R_Y if and only if $\eta^{-1}G'$ belongs to R_X .

Proof. By definition there corresponds to every $F' \epsilon R_X$ a right regular equivalence γ with finitely many classes such that F' is a union of γ-classes; in the proof of III.2. we have seen how to construct λ associated to γ and such that $\eta F'$ is a union of λ classes; since λ is right regular and has only finitely many classes, this proves the forward implication.

In particular, since F_X belongs to R_X, this shows that the total output ηF_X is a regular event.

Let now G' be a subset of ηF_X that belongs to R_Y; G' is defined by a certain right regular relation λ with finitely many classes and we construct the relation σ on F_X by the following conditions:

$\sigma f = \sigma f'$ if and only if

1. $s_1 f = s f f'_{11}$; 2. for any $s' \epsilon S'$, $\lambda\eta(s_1;f;s') = \lambda\eta(s_1;f';s')$.

σ is right regular because, if $\sigma f = \sigma f'$ we have $s_1 ff'' = s_1 f'f''$ and

$\lambda\eta(s_1;ff'';s') = \lambda\eta(s_1;f;f''s')\ \eta(s_1 f;f'';s') = \lambda\eta(s_1;f';f''s')\ \eta(s_1 f';f'';s')$

$= \lambda\eta(s_1;f'f'';s')$

where the second and third equality result from the right regularity of λ an where the second equality is a consequence of $\sigma f = \sigma f'$. Also, σ has at most $S \times \lambda^{S'}$ classes and $\eta^{-1}G'$ is a union of σ-classes. This concludes the proof.

IV.2. Provided that X contains two letters or more, there corresponds to each $G'' \epsilon R_Y$ a right transduction η such that $G'' = \eta F_X$.

<u>Proof</u>. Because of our hypothesis on X it is enough to prove the same statement for an arbitrarily large (finite) input alphabet and, then, to perform a preliminary encoding.

The result is trivial if G" is finite and, by Kleene's theory, it is enough to show that if G and G' are the total output respectively of the right transductions η and η' (with the disjoint input alphabets X,X') such that their total output is respectively G G', GG' and G* in Kleene's notation. The construction given below is the simplest to describe.

Lest S and S' be the set of states of the right transducers η and η'; we can assume that S and S' are disjoint and we define S" as the union of S, S' and of two new states s_1^* and s_o^* for which we have:

1. $s_1^* x'' = s_1 x''$ or $= s_1' x''$ and $\eta_1(s_1^*;x'') = \eta(s_1,x'')$ or $\eta'(s_1',x'')$ according to $x'' \epsilon X$ or X'.

2. $s_o^* x'' = s_o^*$ for all x" and $\eta_1(s_o^*;x'') = e_Y$ for all s",x" such that $s''x'' = s_o^*$.

3. $s''x''$ and $\eta_i(s'';x'')$ are the same as in the original transducers when $s''\epsilon S$ and $x''\epsilon X$ or when $s''\epsilon S'$ and $x''\epsilon X'$.

4. for η_1 : $s''x'' = s_o^*$ when $s''\epsilon S$ and $x''\epsilon X'$ or when $s''\epsilon S'$ and $x''\epsilon X$.

for η_2 : $s''x'' = s_1x''$ and $\eta_2(s'';x'') = \eta_2(s_1^*;x'')$ when $s''\epsilon S'$ and $x''\epsilon X'$

and $s''x'' = s_o^*$ when $s''\epsilon S'$ and $x''\epsilon X$.

for η_3 , we take G=G' (and S identical to S') and we define

$s''x'' = s_1^*$ and $\eta_3(s'';x'') = \eta_3(s_1^*;x'')$ when $s''\epsilon S$ and $x''\epsilon X'$ or when $s''\epsilon S'$ and $s''\epsilon X$.

The verification is left to the reader.

References

1. C. Chevalley, Fundamental concepts of Algebra

2. N. Chomsky, 1959. Information and Control. 2. p. 137

3. E. N. Gilbert and E. F. Moore, 1959. Bell System Technical Journal, 38, p. 933.

4. D. Huffman, 1959. Technical Report 349, R.L.E., M.I.T. Cambridge

5. S. C. Kleene, 1956. Automata Studies, Princeton, 11.

6. E. F. Moore, 1956. Automata Studies, Princeton 11.

7. A. Nerode, 1958, Proc. American Mathematical Society, 9. p.541

8. M. Rabin and D. Scott, 1959. I.B.M. Research Journal. 3. p. 114

9. M. P. Schutzenberger, Sept. 1956. I.R.E. Trans. T.2., p.47

10. J. C. Shepherdson, 1959. I.B.M. Research Journal. 3. p. 198