TWO PAPERS ON THE LINEARITY OF FINITE AUTOMATA[+]

Bernd Reusch[++]

TR 76-267

Department of Computer Science
Cornell University
Ithaca, New York 14853

[+]This research was partially supported by National Science Foundation under grant DCR 75-09433.

[++]On sabbatical leave from Universität Dortmund, Abteilung Informatik, 46 Dortmund 50, Postfach 500500, W. Germany.

REALIZATIONS OF FINITE AUTOMATA WITH LINEAR PARTS

Bernd Reusch

Universitat Dortmund

West Germany

46 Dortmund

REALIZATIONS OF FINITE AUTOMATA WITH LINEAR PARTS+

Bernd Reusch

1. Introduction

The question for linear realization of finite automata has been solved in the literature (see [5] for a list of references). It has been shown that simple methods from linear algebra are sufficient to treat the subject and to give algorithms not only for checking for linearity but also for producing realizations of minimal dimensionality if there exist any. Only a few authors attack the more general problem of realization with linear parts, i.e. [3,4]. In this paper we show how the methods of [5] can be used to solve this problem completely without extra cost after failing to produce a linear realization.

2. Basic Notions

In this section we want to introduce some basic ideas and our notion of them.

Def. 2.1: $A = (X,S,Y,\delta,\lambda)$ is a finite automaton if

1. X,S,Y are non-empty finite sets
2. δ and λ are mappings

   $\delta: X \times S \to S$

   $\lambda: X \times S \to Y$

δ and λ are extended to range on $X^* \times S$ in the usual way.

As a picturial notation for A we use $X \to \boxed{A} \to Y$ or

$$\begin{tikzcd} X & \\ \boxed{A} \arrow[u] \arrow[r] & Y \end{tikzcd}$$

or some other arrangement of incoming and outgoing arrows.
If $X = X_1 \to X_2$ and/or $Y = Y_1 \to Y_2$ we note this for instance by

$$\begin{tikzcd} & X_1 & \\ X_2 \arrow[r] & \boxed{A} \arrow[u] \arrow[r] \arrow[d] & Y_1 \\ & Y_2 & \end{tikzcd} .$$

If X or S are singletons, we simply draw $\boxed{A} \to Y$ or $X \to A$ resp.

We use the following notion of composition of automata.

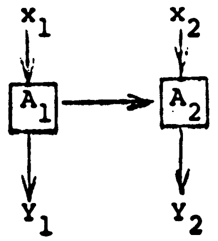
<u>Def. 2.2:</u> Let $A_1 = (X_1 \times S_2, S_1, \delta_1, \lambda_2)$ and $A_2 = (X_2 \times S_1, S_2, \delta_2, \lambda_2)$ be automata.

Then

$$\begin{tikzcd} X_1 & X_2 \\ \boxed{A_1} \arrow[u] \arrow[d] \arrow[r, shift left] & \boxed{A_2} \arrow[u] \arrow[d] \arrow[l, shift left] \\ Y_1 & Y_2 \end{tikzcd} = (X_1 \times X_2, S_1 \times S_2, Y_1 \times Y_2, \delta, \lambda)$$

where $\delta((x_1,x_2),(s_1,s_2)) = (\delta_1((x_1,s_2),s_1), \delta_2((x_2,s_1),s_2))$

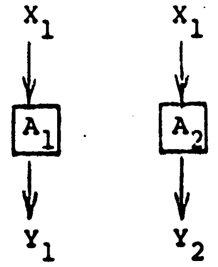
$\lambda((x_1,x_2),(s_1,s_2)) = (\lambda_1((x_1,s_2),s_1), \lambda_2((x_2,s_1),s_2))$

Special cases: a) 

if $\delta_1((x_1,s_2),s_1) = \tilde{\delta}_1(x_1,s_1)$

$\lambda_1((x_1,s_2),s_1) = \tilde{\lambda}_1(x_1,s_1)$

for some $\tilde{\delta}_1$ and $\tilde{\lambda}_1$.

b) 

if (as above for 1 and 2)

Special automaton: $X \to \boxed{N}$, $(X,X,\{y\},\delta,\lambda)$

$\delta(x,x') = x$

Our definitions simply use an arrow from A_1 to A_2, if A_2 obtains information about the state of A_1 to compute its own next state. The special automaton N is introduced to have a notion for "branching".

The next goal is to define "realization", "linear realization" and finally "realization with linear parts".

<u>Def. 2.3</u>: a) $A = (X,S,Y,\delta,\lambda)$ is a <u>homomorphic image</u> of $A' = (X',S',Y',\delta',\lambda')$ if there are surjections h_X, h_S, h_Y such that the following diagram commutes

$$\begin{tikzcd}
Y' \arrow[d, "h_Y"'] & X' \times S' \arrow[l, "\lambda'"'] \arrow[r, "\delta'"] \arrow[d, "h_X \times h_S"] & S' \arrow[d, "h_S"] \\
Y & X \times S \arrow[l, "\lambda"'] \arrow[r, "\delta"] & S
\end{tikzcd}$$

b) A is <u>isomorphic</u> to A', if A is a homomorphic image of A' and the mappings h_X, h_S, h_Y are bijections.

<u>Def. 2.4</u>: $A = (X,S,Y,\delta,\lambda)$ is a <u>subautomaton</u> of $A' = (X',S',Y',\delta',\lambda')$ if $X \subseteq X'$, $S \subseteq S'$ and $Y \subseteq Y'$ and the mappings δ and λ are restrictions of δ' and λ' resp. to $X \times S$.

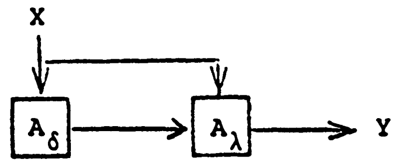
<u>Def. 2.5</u>: $A = (X,S,Y,\delta,\lambda)$ is <u>realized</u> by $A' = (X',S',Y',\delta',\lambda')$ iff A is isomorphic to a subautomaton of A'.

Recalling the definitions of "subautomaton" and "realized" we know that A is realized by A' iff there are <u>injections</u> h_X, h_S and h_Y such that the following diagram commutes.

$$\begin{tikzcd}
Y \arrow[d, "h_Y"'] & X \times S \arrow[l, "\lambda"'] \arrow[r, "\delta"] \arrow[d, "h_X \times h_S"] & S \arrow[d, "h_S"] \\
Y' & X' \times S' \arrow[l, "\lambda'"'] \arrow[r, "\delta'"] & S'
\end{tikzcd}$$

To use our definitions, we will state two simple lemmas without proof.

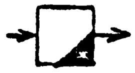
<u>Lemma 2.1</u>: $A = (X,S,Y,\delta,\lambda)$ is isomorphic to



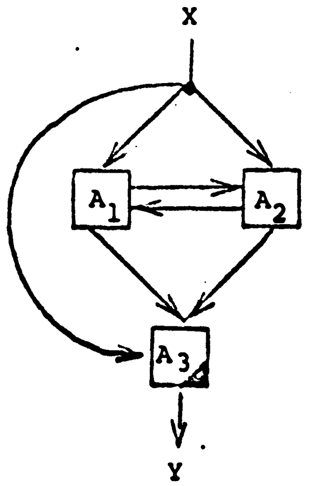
where

$A_\delta = (X,S,\{\bar{y}\},\delta,\bar{\lambda})$

$A_\lambda = (X{\times}S,\{\bar{s}\},Y,\bar{\delta},\lambda)$

A_δ is a one-output automaton and A_λ a one-state automaton. As a shorthand we will use  for one-state automata from now on.

<u>Lemma 2.2</u>: $A = (X,S,Y,\delta,\lambda)$ is realized by



$$
\begin{aligned}
\text{where}\quad A_1 &= (X\times S_2,\ S_1,\ \{\bar{y}\},\ \delta_1,\ \lambda_1\} \\
A_2 &= (X\times S_1,\ S_2,\ \{\tilde{y}\},\ \delta_2,\ \lambda_2\} \\
A_3 &= (X\times S_1\times S_2,\ \{\bar{s}\},\ Y,\ \delta_3,\ \lambda_3\}
\end{aligned}
$$

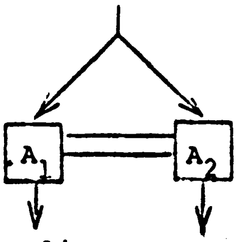
iff there are mappings
$h_1\colon S \to S_1$ and $h_2\colon S \to S_2$
such that $h_1\times h_2\colon S \to S_1 \times S_2$ is
an injection.

Automata may carry some special algebraic structure. The one we will use is linearity over a given finite field K.

<u>Def. 2.6</u>: $A = (X,S,Y,\delta,\lambda)$ is <u>linear</u> over the finite field K if X,S,Y are finite dimensional vector spaces over K and δ,λ are linear mappings.

<u>Def. 2.7</u>: $A = (X,S,Y,\delta,\lambda)$ is <u>linear realizable</u> over K if it is realized by a linear automaton over K.

<u>Def. 2.8</u>: $A = (X,S,Y,\delta,\lambda)$ is <u>realized with linear parts</u> if it is realized by an automaton of the form



where A_1 is a linear automaton.

As shorthand we will use $\to\boxed{}\to$ for linear automata.

In this paper we will study the possibility of realization with linear parts for a given automaton and a given field.

We want to reformulate our definition a little bit to make it more uniform.

<u>Lemma 2.3:</u> $A = (X,S,Y,\delta,\lambda)$ is realized with linear parts over K iff there are natural numbers ℓ,m,n and injections h_X, h_S, h_Y such that the following diagram commutes

$$\begin{tikzcd}
Y \arrow[d, "h_Y"'] & X \arrow[l, "\lambda"'] \arrow[d, "h_X"] & \times & S \arrow[r, "\delta"] \arrow[d, "h_S"] & S \arrow[d, "h_S"] \\
K^{\ell} & K^{m} \arrow[l, "\bar{\lambda}"'] & \times & K^{n} \arrow[r, "\bar{\delta}"] & K^{n}
\end{tikzcd}$$

and some of the mappings $i\text{-proj}\circ\bar{\delta}$, $i = 1,\dots,n$, and $j\text{-proj}\circ\bar{\lambda}$, $j=1,\dots,\ell$ are linear.

From this lemma we know that we have to search in the structures defined next, if we want to find realization with linear parts.

<u>Def. 2.9:</u> a) $\mathcal{P}_X := \{Q \mid Q\colon X \to K\}$

b) $\mathcal{P}_S := \{P \mid P\colon S \to K\}$

c) $\mathcal{P}_Y := \{R \mid R\colon Y \to K\}$

With the usual operations on mappings, namely $(P_1 \oplus P_2)(b) = P_1(b) + P_2(b)$ and $(k \cdot P)(b) = k \cdot P(b)$, the sets $\mathbb{P}_X$, $\mathbb{P}_S$ and $\mathbb{P}_Y$ turn out to be vector spaces. Moreover, the constant mappings form a subspace and we may talk about the factor spaces $\mathbb{F}_X$, $\mathbb{F}_S$ and $\mathbb{F}_Y$ respectively.

The spaces so defined are linked by operators, which are specific for the automaton under consideration.

Def. 2.10: Let $A = (X,S,Y,\delta,\lambda)$ be an automaton.

a) $\pi_x P(s) := P(\delta(x,s))$, for all $s \in S$, $x \in X^*$ and $P \in \mathbb{P}_S$.

b) $\tau_x R(s) := R(\lambda(x,s))$, for all $s \in S$, $x \in X^*$ and $R \in \mathbb{P}_Y$.

These operators turn out to be linear ([5]) and various subspaces related to them will be used to characterize automata which posses realizations with linear parts.

Def. 2.11: a) For all $P \in \mathbb{P}_S$ $(R \in \mathbb{P}_Y)$ we denote by $\bar{P} \in \mathbb{F}_S$ $(\bar{R} \in \mathbb{F}_Y)$ the element of $\mathbb{F}_S$ $(\mathbb{F}_Y)$ onto which P (R) is mapped under the canonical mapping $\mathbb{P}_S \to \mathbb{F}_S$ $(\mathbb{P}_Y \to \mathbb{F}_Y)$.

b) $\mathbb{P}_S^n := \{P \in \mathbb{P}_S \mid \overline{\pi_x P} = \overline{\pi_{x'} P}$, for all $x, x' \in X^*$ of equal length up to $n\}$, $n=1,2,3,\ldots$.

c) $\mathbb{P}_S^{=n} := \{P \in \mathbb{P}_S \mid \pi_x P = \pi_{x'} P$, for all $x,x' \in X^*$ of equal length up to $n\}$, $n=1,2,3,\ldots$.

d) $P_S^{\infty} := \{P \in P_S \mid \overline{\pi_x P} = \overline{\pi_{x'} P}$, for all $x, x' \in X^*$ of equal length$\}$.

e) $P_S^{=} := \{P \in P_S \mid \pi_x P = \pi_{x'} P$, for all $x, x' \in X^*$ of equal length$\}$.

f) $P_Y^{n} := \{R \in P_Y \mid \overline{\tau_x R} = \overline{\tau_{x'} R}$, for all $x, x' \in X^*$ of equal length up to $n\}$,
$n = 1,2,3,\ldots$

g) $P_Y^{=n} := \{R \in P_Y \mid \tau_x R = \tau_{x'} R$, for all $x, x' \in X^*$ of equal length up to $n\}$,
$n = 1,2,3,\ldots$

h) $P_Y^{\infty} := \{R \in P_Y \mid \overline{\tau_x R} = \overline{\tau_{x'} R}$, for all $x, x' \in X^*$ of equal length$\}$.

i) $P_Y^{=} := \{R \in P_Y \mid \tau_x R = \tau_{x'} R$, for all $x, x' \in X^*$ of equal length$\}$.

All so defined sets can easily be shown to be vector spaces containing the vector space of constant mappings. Hence we can speak of the resp. factor spaces $\mathbb{F}_S^n$, $\mathbb{F}_S^{=n}$, $\mathbb{F}_S^{\infty}$, $\mathbb{F}_S^{=}$, $\mathbb{F}_Y^n$, $\mathbb{F}_Y^{\infty}$ and $\mathbb{F}_Y^{=}$.

There are a number of relations between the vector spaces defined (see [5]).

<u>Lemma 2.4</u>: a) $\mathbb{P}_S \supseteq \mathbb{P}_S^1 \supseteq \mathbb{P}_S^2 \supseteq \ldots$

b) $\mathbb{P}_Y \supseteq \mathbb{P}_Y^1 \supseteq \mathbb{P}_Y^2 \supseteq \ldots$

c) $\mathbb{P}_S^\infty \subseteq \mathbb{P}_S^n$, for all n = 1,2,...

d) $\mathbb{P}_Y^\infty \subseteq \mathbb{P}_Y^n$, for all n = 1,2,...

e) $\mathbb{P}_S \supseteq \mathbb{P}_S^{=1} \supseteq \mathbb{P}_S^{=2} \supseteq \ldots$

f) $\mathbb{P}_Y \supseteq \mathbb{P}_Y^{=1} \supseteq \mathbb{P}_Y^{=2} \supseteq \ldots$

g) $\mathbb{P}_S^= \subseteq \mathbb{P}_S^{=n}$, for all n = 1,2,...

h) $\mathbb{P}_Y^= \subseteq \mathbb{P}_Y^{=n}$, for all n = 1,2,...

i) $\mathbb{P}_S^n \supseteq \pi_x \mathbb{P}_S^{n+1}$, for all $x \in X$ and n = 1,2,...

j) $\mathbb{P}_S^n \supseteq \tau_x \mathbb{P}_Y^{n+1}$, for all $x \in X$ and n = 1,2,...

<u>Lemma 2.5:</u> If $k = |S|$, i.e. k is the number of states, then

a) $\mathbb{P}_Y^k = \mathbb{P}_Y^\infty$

b) $\mathbb{P}_S^{k-1} = \mathbb{P}_S^\infty$

c) $\mathbb{P}_Y^{=k} = \mathbb{P}_Y^=$

d) $\mathbb{P}_S^{=k} = \mathbb{P}_S^=$

Methods for computing the various vector spaces are given in [5].

The last thing to do is relating mappings to equivalence relations in a very simple way.

<u>Def. 2.12</u>: For $M \in \{X,S,Y\}$ we define

a) If $P \in \mathbb{P}_M$, then $m_1 \equiv m_2\ (\alpha P)$ iff $P(m_1) = P(m_2)$.

b) $\beta\{P_1,\ldots,P_r\} = \prod_{i=1}^{r} \alpha P_i$

0 and 1 are the smallest and biggest equivalence relations respectively. We obviously have

<u>Lemma 2.6</u>: If $\mathbb{P}_1 \subseteq \mathbb{P}_2 \subseteq \mathbb{P}_M$, then

$\beta\mathbb{P}_1 \geq \beta\mathbb{P}_2$ and $\beta\mathbb{P}_M = 0$.

3. Linearity of Single Coordinates.

In this section we will recall the basic theorems from [5] on the linearity of single coordinates.

<u>Theorem 3.1</u>: A) There is a realization of (X,S,Y,δ,λ) such that $i\text{-proj}\circ\bar{\delta}$ is linear

iff 1. There are $\{Q_1,\ldots,Q_m\} \subseteq \mathbb{P}_X$, $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S$ and $\{R_1,\ldots,R_\ell\} \subseteq P_Y$ such that $\beta\{Q_1,\ldots,Q_m\} = 0$, $\beta\{P_1,\ldots,P_n\} = 0$, $\beta\{R_1,\ldots,R_\ell\} = 0$.

2. $\pi_x P_i = \bigoplus_{j=1}^{n} a_{ij} P_j \oplus h(i,x)$, for all $x \in X$ and some $h(i,x) \in K$.

3. $\emptyset\bar{P}_i = \bigoplus_{k=1}^{m} b_{ik}\bar{Q}_k$ (*)

B) There is a realization of (X,S,Y,δ,λ) such that j-proj$\circ\bar{\lambda}$ is linear

iff 1. as in A)

2. $\tau_x R_j = \bigoplus_{\ell}^{n} \dot{c}_{j\ell}\, P_\ell \oplus k(j,x)$ for all $x \in X$ and some $k(i,x) \in K$

3. $\psi\bar{R}_j = \bigoplus_{r=1}^{m} d_{jr}\bar{Q}_r$ (*)

<u>Proof</u>: As given in [5] by exploiting commutativity of the realizat: diagram.

Conditions 1) and 3) are superfluous, if we are interested in linear realizability for one coordinate. If the coding of X still is to be given, we may very well use $\emptyset P_i$ as a part of it. Hence we obtain:

<u>Theorem 3.2</u>: A) There is a realization of (X,S,Y,δ,λ) such that i-proj$\circ\bar{\delta}$ is linear iff there are $\{P_1,\dots,P_k\} \subseteq \mathbb{P}_S$, $P_i \in \mathbb{P}_S$ such that

$$\overline{\pi_x P_i} = \bigoplus_{j=1}^{k} a_{ij}\, \bar{P}_j \quad \text{for all } x \in X.$$

B) There is a realization of (X,S,Y,δ,λ) such that j-proj$\circ\bar{\lambda}$ is linear

---

*$\emptyset: \mathbb{F}_S^1 \to \mathbb{F}_X$ and $\psi: \mathbb{F}_Y^1 \to \mathbb{F}_X$ are not defined in this paper, since they are never used anywhere. For definition see [5].

iff there are $\{P_1,\ldots,P_k\} \subseteq \mathbb{P}_S$, $R_j \in \mathbb{P}_Y$

such that

$$\overline{\tau_x R_j} = \bigoplus_{j=1}^{k} c_{j\ell}\, \bar{P}_\ell \text{ for all } x \in X.$$

From these conditions we obtain $\overline{\pi_x P_i} = \overline{\pi_{x'} P_i}$ and $\overline{\tau_x R_j} = \overline{\tau_{x'} R_j}$ for all $x, x' \in X$. On the other hand we may include $\pi_x P_i$ and $\tau_x R_j$ in the coding of S and hence result in the following theorem.

<u>Theorem 3.3</u>: A) There is a realization of (X,S,Y,δ,λ) such that i-projo$\bar{\delta}$ is linear iff $P_i \in \mathbb{P}_S^1$ for some $P_i \in \mathbb{P}_S$.

B) There is a realization of (X,S,Y,δ,λ) such that j-projo$\bar{\lambda}$ is linear iff $R_j \in \mathbb{P}_Y^1$ for some $R_j \in \mathbb{P}_Y$.

In Theorem 3.1 the constants h(i,x) and k(j,x) represent the influence of $i_x(X)$ on the linear functions. If there is no influence, these constants are equal to 0. Hence by the same argument as above, we obtain the following result.

<u>Theorem 3.4</u>: A) There is a realization of (X,S,Y,δ,λ) such that i-projo$\bar{\delta}$ is linear and independent of $i_x(X)$ iff $P_i \in \mathbb{P}_S^{=1}$ for some $P_i \in \mathbb{P}_S$.

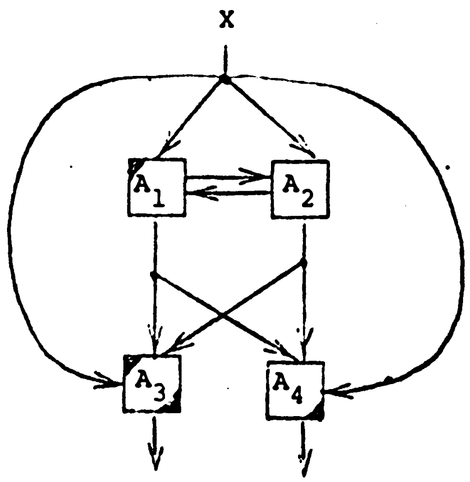
B) There is a realization of (X,S,Y,δ,λ) such that j-projo$\bar{\lambda}$ is linear and independent of $i_x(x)$ iff $R_j \in \mathbb{P}_Y^{=1}$ for some $R_j \in \mathbb{P}_Y$.

The theorems given here are used in the next section to solve the question for linear and partial linear realization of finite automata.

4. Decomposition with linear parts.

If for a given automaton A there is realizations with linear parts, then this realization has the following general form:

(*)



There are two kinds of questions to be asked about the general realization (*):

i) Is there a realization (*) such that one or some of the A_i's are trivial/nontrivial?

ii) Is there a realization (*) such that one or some of the arrows are missing?

Let us first consider i). There always exist realizations such that A_2 and/or A_4 are nontrivial and/or such that A_1 and/or A_3 are trivial. Hence all situations of interest are treated in the following theorem.

<u>Theorem 4.1</u>: For a given automaton $A = (X,S,Y,\delta,\lambda)$ there is a realization of the form (*) where:

a) A_1 is nontrivial iff $\beta\mathbb{P}_S^1 < 1$
b) A_3 is nontrivial iff $\beta\mathbb{P}_Y^1 < 1$
c) A_2 is trivial iff $\beta\mathbb{P}_S^\infty = 0$
d) A_4 is trivial iff $\beta\mathbb{P}_Y^1 = 0$
e) A_2 <u>and</u> A_4 are trivial iff $\beta\mathbb{P}_Y^\infty = 0$

<u>Proof</u>: a) and b) are in fact reformulations of Theorem 3.3 since $P_i \notin K$, $R_j \notin K$ and $P_i \in \mathbb{P}_S^1$, $R_j \in \mathbb{P}_Y^1$ iff $\beta\mathbb{P}_S^1 < 1$ and $\beta\mathbb{P}_Y^1 < 1$.

d) If A_4 is trivial, we have used $\{R_1,\ldots,R_\ell\} \subseteq \mathbb{P}_Y^1$ for coding Y. From $\beta\{R_1,\ldots,R_\ell\} = 0$ we conclude $\beta\mathbb{P}_Y^1 = 0$.
If $\beta P_Y^1 = 0$ take any subset $\{R_1,\ldots,R_\ell\} \subseteq P_Y^1$ satisfying $\beta\{R_1,\ldots,R_\ell\} = 0$ for coding.

c) From Theorem 3.1 we know that A_2 is trivial iff S is coded using $\{P_1,\ldots,P_n\}$ such that $\overline{\pi_x P_i} = \bigoplus_{j=1}^{n} a_{ij}\bar{P}_j$ for all $i = 1,\ldots,n$ and $x \in X$.
Let us assume $\overline{\pi_x P_i} = \bigoplus_{j=1}^{n} a_{ij}^k P_j$ for all $i = 1,\ldots,n$

and all $x \in X^*$ of length k. Then for all $\bar{x} = x\, x' \in X^*$ of length k+1 we obtain

$$\overline{\pi_{\bar{x}} P_i} = \overline{\pi_{x'} \pi_x P_i} = \bar{\pi}_{x'} \bigoplus_{j=1}^{n} a_{ij}^{k} \bar{P}_j =$$

$$= \bigoplus_{j=1}^{n} a_{ij}^{k} (\overline{\pi_{x'} P_j}) = \bigoplus_{j=1}^{n} a_{ij}^{k} (\bigoplus_{j=1}^{n} a_{ij} \bar{P}_j) =$$

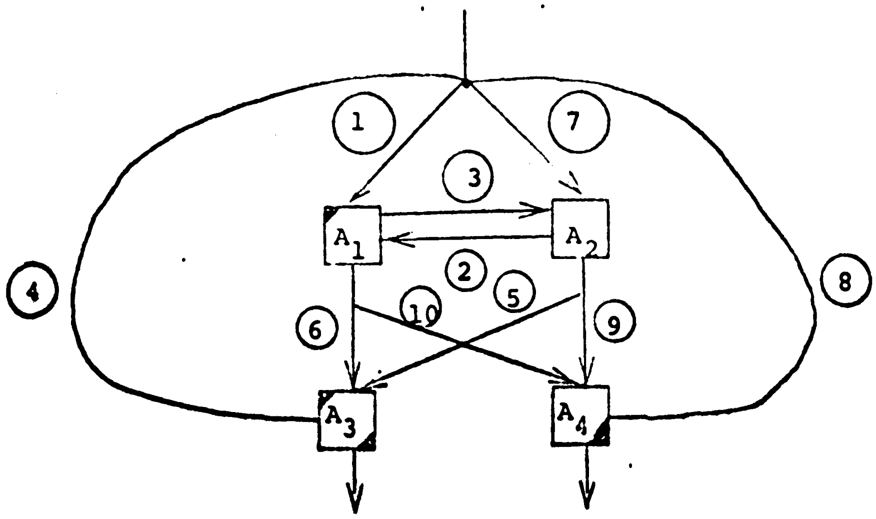
$$= \bigoplus_{j=1}^{n} a_{ij}^{k+1} \bar{P}_j$$

Hence we proved by induction that $P_i \in \mathcal{P}_S^k$ for all k and all $i = 1,\ldots,n$. Therefore $\{P_1,\ldots,P_n\} \subseteq \mathcal{P}_S^\infty$ and from $\beta\{P_1,\ldots,P_n\} = 0$ we conclude $\beta\mathcal{P}_S^\infty = 0$.

If on the other hand $\beta\mathcal{P}_S^\infty = 0$ then we may use $\{P_1,\ldots,P_n\} = \mathcal{P}_S^\infty$ for coding. In this case $\pi_x P_i = P_j \in \{P_1,\ldots,P_n\}$ for all i and by Theorem 3.2 we get a linear realization.

e) This is in fact Corollary 4.2 of [5] and proven there.

Let us now come back to question ii), whether there are realizations of the form (*) such that one or the other arrow is missing.



Missing of arrows (7), (8), (9) and (10) may be settled using methods described in [2]. Missing of arrow (6) is of no interest since it can always be omitted by repeating the computation of A_1 in A_2. Hence we are left with the arrows (1), (2), (3), (4) and (5) and therefore with 32 possible cases.

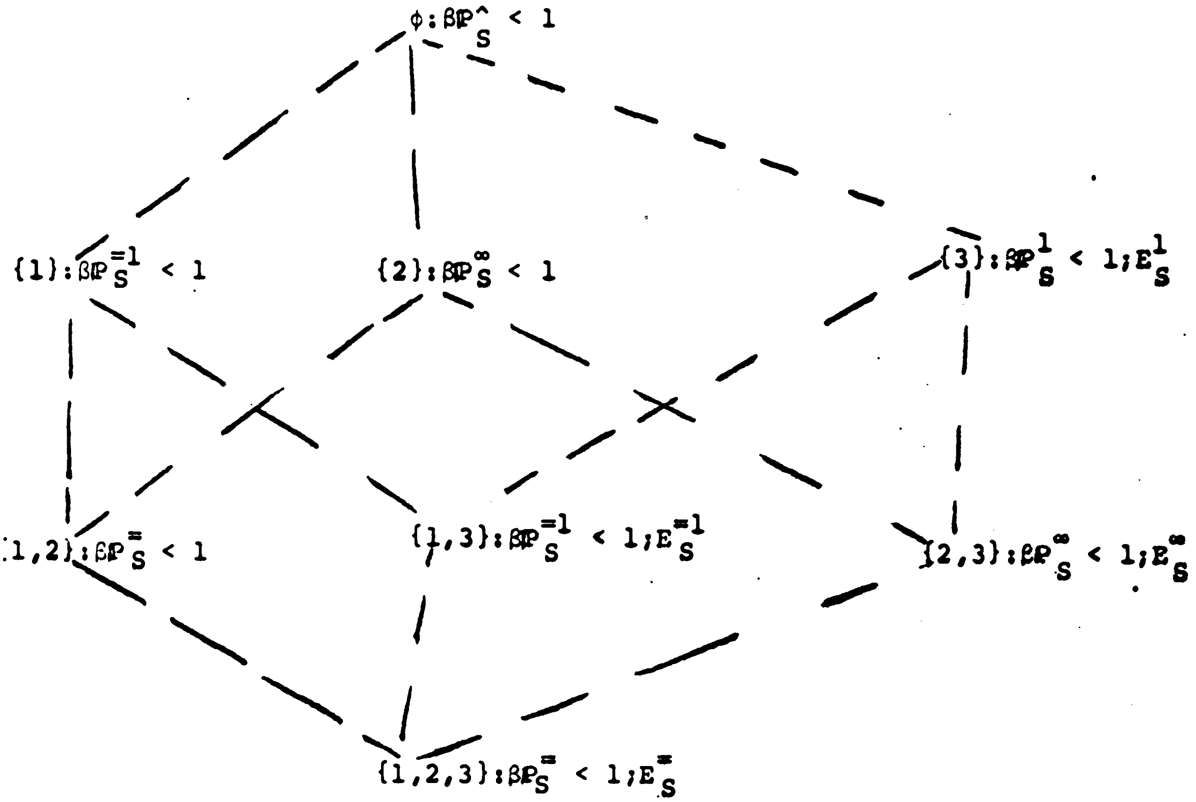
Let us first consider (1), (2) and (3), i.e. the state behavior of the given automaton. Theorem 4.2 will list the results for all 8 cases. The following shorthands will be used.

a) for instance $\{2,3\}$: $\beta P_S^\infty < 1$; E_S^∞

means There is a realization such that arrows (2) and (3) are missing iff $\beta P_S^\infty < 1$ and property E_S^∞ is satisfied.

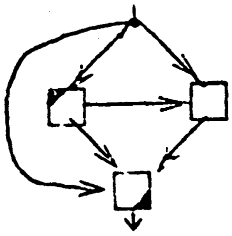
b) For all $\nu \in \{1, =1, \infty, =\}$, E^{ν}_S is the property: There exists a partition P on S with the substitution property (see [ ]) such that $\beta\mathbb{P}^{\nu}_S \cdot P = 0$.

Theorem 4.2:



We will not prove this theorem in detail but pick some of the more interesting cases and examine them.

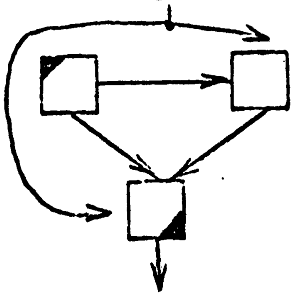
<u>Lemma 4.1</u>: There is a realization of (X,S,Y,δ,λ) of the form



iff $\beta\mathbb{P}_S^{\infty} < 1$

<u>Proof:</u> If $\beta\mathbb{P}_S^{\infty} < 1$ then $P_S^{\infty} \nsubseteq K$. Use $\{P_1,\ldots,P_n\} = \mathbb{P}_S^{\infty}$ for partial coding. Then $\pi_x P_i \in \{P_1,\ldots,P_n\} = \mathbb{P}_S^{\infty}$ for all $x \in X$ and $i = 1,\ldots,n$ and the argument goes as in the proof of Theorem 4.1,c). If there is a realization of this form, again the same argument as for Theorem 4.1,c) goes through except the property $\beta\mathbb{P}_S^{\infty} = 0$. But if the realization is nontrivial i.e. uses at least one $P_i \notin K$, then $\beta\mathbb{P}_S^{\infty} < 1$.

<u>Lemma 4.2</u>: There is a realization of (X,S,Y,δ,λ) of the form

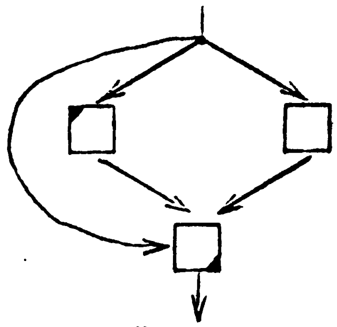


iff $\beta\mathbb{P}_S^{=} < 1$

<u>Proof:</u> If $\beta\mathbb{P}_S^{=} < 1$ then $\mathbb{P}_S^{=} \nsubseteq K$. Use $\{P_1,\ldots,P_n\} = \mathbb{P}_S^{=}$ for partial coding. Then $\pi_x P_i \in \{P_1,\ldots,P_n\} = \mathbb{P}_S^{=}$ for all $i = 1,\ldots,n$ and all $x \in X$. Hence by Theorem 3.4 we are through.

If there is a realization of the given form, then there are $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S^1$ such that $P_i = \bigoplus_{j=1}^{n} a_{ij}P_j$ for $i=1,\ldots,n$ by Theorem 3.4. Similar as for Theorem 4.1.c we prove by induction that $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S^{=}$. From $P_i \notin K$ for some i follows $\beta\{P_1,\ldots,P_n\} < 1$ and therefore $\beta\mathbb{P}_S^{=} < 1$.

<u>Lemma 4.3</u>: There is a realization of (X,S,Y,δ,λ) of the **form**



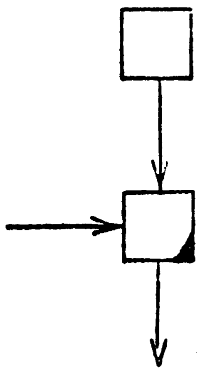
iff $\beta\mathbb{P}_S^{\infty} < 1$ and E_S^{∞}.

<u>Proof</u>: If $\beta\mathbb{P}_S^{\infty} < 1$ and E_S^{∞}, then we may use $\{P_1,\ldots,P_n\} = \mathbb{P}_S^{\infty}$ and $\{P_{n+1},\ldots,P_\ell\}$ for coding where $\beta\{P_{n+1},\ldots,P_\ell\} \leq P$. By previous remarks and methods of [H.S.] we are through.

If there is a realization of the given form, then by Lemma 4.1 we have $\beta\mathbb{P}_S^{\infty} < 1$. The rest follows from [2].

There is one interesting case which is not covered by Theorem 4.1 and Theorem 4.2, namely the case where the state behavior is independent of X. We state the result without reference to linearity, but from Lemma 4.2 we conclude the total linearity of autonomous automata.

<u>Lemma 4.4</u>: There is a realization of (X,S,Y,δ,λ) of the form



iff $\beta\mathbb{P}_S^{=} = 0$.

<u>Proof</u>: If $\beta\mathbb{P}_S^{=} = 0$, then $\{P_1,\ldots,P_n\} = \mathbb{P}_S^{=}$ gives the result. If there is a realization of this form, then there are $P_1,\ldots,P_n$ such that $\beta\{P_1,\ldots,P_n\} = 0$ and $\pi_x P_i = f_i(P_1,\ldots,P_n)$ for all $x \in X$ and $f_i: \mathbb{P}_S \times \ldots \times \mathbb{P}_S \to \mathbb{P}_S$, $i = 1,\ldots,n$. Hence we conclude $P_i \in \mathbb{P}_S^{=1}$ for $i = 1,\ldots,n$. Induction over the length of $x \in X^*$ shows $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S^{=}$ and hence $\beta\mathbb{P}_S^{=} = 0$. The induction step is a little different from that of Theorem 4.1c because of the possible non-linearity of some of the f_i's, but is left as an exercise.

So far we have dealt with the state-behavior of the given automaton. The next step will be to discuss the output-behavior and then combining the two aspects.

Theorem 4.3:

$$\begin{tikzcd}
 & \emptyset : \beta\mathbb{P}_Y^1 < 1 \arrow[dl, dash, dashed] \arrow[dr, dash, dashed] & \\
\{4\} : \beta\mathbb{P}_Y^{=1} < 1 \arrow[dr, dash, dashed] & & \{5\} : \beta\mathbb{P}_Y^2 < 1 \arrow[dl, dash, dashed] \\
 & \{4,5\} : \beta(\neg\mathbb{P}_Y^{=1} \cap \mathbb{P}_S^1) < 1 &
\end{tikzcd}$$

Proof: The case where no arrow is missing is already discussed in Theorem 4.1.

a) Missing of arrow (4) is dealt with in Theorem 3.4.

b) If there is a realization where arrow (5) is missing, then there is a partial coding of Y using $R_1,\ldots,R_\ell$ and a partial coding of S using $P_1,\ldots,P_n$ such that $\{R_1,\ldots,R_\ell\} \subseteq \mathbb{P}_Y^1$ and $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S^1$ and

$\tau_x \bar{R}_j = \bigoplus_{i=1}^{n} a_{ij} \bar{P}_i$ for all $j = 1,\ldots,\ell$ and all $x \in X$. But then clearly $\tau_x \bar{R}_j = \tau_{x'} \bar{R}_j$ for all $x, x' \in X^*$ of length 2 and if $R_j \nsubseteq K$ for some j, then $\beta \mathbb{P}_Y^2 < 1$.

c) If (4) and (5) are missing, then there is a partial coding of Y using $\{R_1,\ldots,R_\ell\} \subseteq \mathbb{P}_Y^{=1}$ and a partial coding of S using $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S^1$ such that τR_j is a linear combination of $\{P_1,\ldots,P_n\}$ for all $j=1,\ldots,\ell$. Hence $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S^1 \cap \tau\mathbb{P}_Y^{=1}$ and therefore $\beta(\tau\mathbb{P}_Y^{=1} \cap \mathbb{P}_S^1) < 1$.
If now $\beta(\tau\mathbb{P}_Y^{=1} \cap \mathbb{P}_S^1) < 1$, we use $\tau\mathbb{P}_Y^1 \cap \mathbb{P}_S^1$ for partial coding of S and all $R_j \in \mathbb{P}_Y^{=1}$ for partial coding of Y if $\tau R_j \in \mathbb{P}_S^1$.

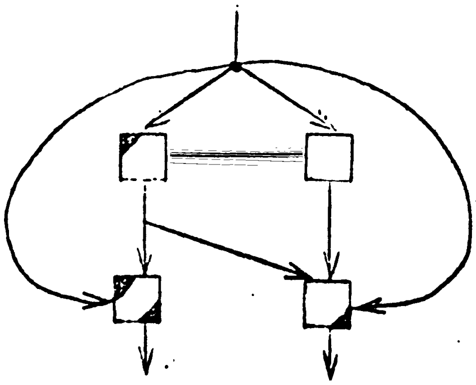
Our next step is to combine Theorem 4.2 and Theorem 4.3. Some of the combinations are trivial in the sense that the related conditions in both theorems are just put together without making any use of the structural relation between the various subspaces involved. This is true for the cases {1,4}; {2,4} and {1,2,4}. All other cases are treated in our next theorem.

Theorem 4.4:

$$
\begin{tikzcd}
 & & \phi:\beta P^{1}_{Y} < 1 \arrow[dll, dash, dashed] \arrow[dr, dash, dashed] & & \\
\{1,5\}:\beta(\tau P^{1}_{Y} \cap P^{=1}_{S}) < 1 \arrow[d, dash, dashed] \arrow[drr, dash, dashed] & & & \{2,5\}:\beta P^{\infty}_{Y} < 1 \arrow[dl, dash, dashed] \arrow[dr, dash, dashed] & \\
\{1,4,5\}:\beta P^{=2}_{Y} < 1 \arrow[drr, dash, dashed] & & \{1,2,5\}:\beta(\tau P^{1}_{Y} \cap P^{=}_{S}) < 1 \arrow[d, dash, dashed] & & \{2,4,5\}:\beta(\tau P^{=1}_{Y} \cap P_{S} \arrow[dll, dash, dashed] \\
 & & \{1,2,4,5\}:\beta P^{=}_{Y} < 1 & &
\end{tikzcd}
$$

Again we prove some of the more interesting cases in the following lemmas.

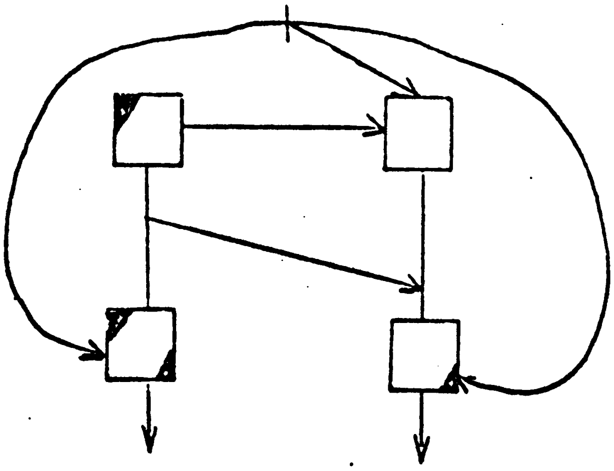
Lemma 4.5: There is a realization of $(X,S,\Psi,\delta,\lambda)$ of the form



iff $\beta P^{\infty}_{Y} < 1$.

<u>Proof</u>: If there is a realization of this form, then there are partial codings $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S^\infty$ of S and $\{R_1,\ldots,R_\ell\} \subseteq \mathbb{P}_Y^1$ of Y such that $\tau\{R_1,\ldots,R_\ell\}$ is linearly generated by $\{P_1,\ldots,P_n\}$. Hence $\tau\{R_1,\ldots,R_\ell\} \subseteq \mathbb{P}_S^\infty$ and finally $\{R_1,\ldots,R_\ell\} \subseteq \mathbb{P}_Y^\infty$. If $\beta\mathbb{P}_Y^\infty < 1$, use $\{R_1,\ldots,R_\ell\} = \mathbb{P}_Y^\infty$ for partial coding of Y.

<u>Lemma 4.6</u>: There is a realization of (X,S,Y,δ,λ) of the form



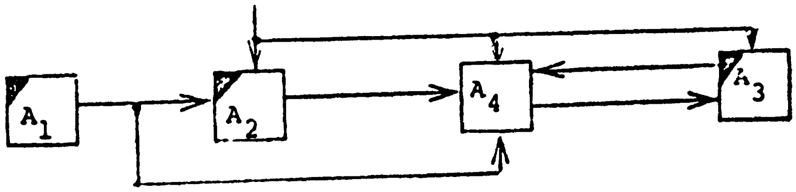
iff $\beta(\tau\mathbb{P}_Y^1 \cap \mathbb{P}_S^\infty) < 1$

<u>Proof</u>: If there is a realization of this form, then there are partial codings $\{P_1,\ldots,P_n\} \subseteq \mathcal{P}_S^{=}$ for S and $\{R_1,\ldots,R_\ell\} \subseteq \mathcal{P}_Y^1$, such that $\beta\{P_1,\ldots,P_n\}$ is linearly generated by $\{P_1,\ldots,P_n\}$. Hence $\tau\{P_1,\ldots,P_n\} \subseteq \mathcal{P}_S^{=}$ and $\beta(\tau\mathcal{P}_Y^1 \cap \mathcal{P}_S^{=}) < 1$. If $\beta(\tau\mathcal{P}_Y^1 \cap \mathcal{P}_S^{=}) < 1$, take $\{P_1,\ldots,P_n\} = (\tau\mathcal{P}_Y^1 \cap \mathcal{P}_S^{=})$ for partial coding of S and take R_j for partial coding of Y, if $\tau P_j \in \mathcal{P}_S^{=}$.

5. Canonical decomposition.

The techniques and results derived so far allow us to prove the existence of a canonical decomposition of a given automa automaton, which makes maximal use of possible linearity, or conversely singles out the "hard core" of non-linearity. For the sake of simplicity we concentrate on the state behavior of automata. Output can be treated similarly.

<u>Theorem 5.1</u>: For every automaton (X,S,Y,δ,λ) there is a realization of the form



such that the linear parts are "maximal", i.e. if $P_{i,1},\ldots,P_{i,n(i)}$ is used in coding S to produce A_i, i=1,2,3, and if there is another realization of the same form using $\tilde{P}_{i,1},\ldots,\tilde{P}_{i,m(i)}$, i=1,2,3, for this purpose, then

$\langle P_{i,1},\ldots,P_{i,m(i)}\rangle \subseteq \langle P_{i,1},\ldots,P_{i,n(i)}\rangle$, i=1,2,3.

__Proof:__ We use the following sets of indexed partitions.

$P_{1,1},\dots,P_{1,n(1)}$ such that $\{\bar{P}_{1,1},\dots,\bar{P}_{1,n(1)}\} = \mathbb{F}_S^=$

$P_{2,1},\dots,P_{2,n(2)}$ such that $\{\bar{P}_{2,1},\dots,\bar{P}_{2,n(2)}\} = \mathbb{F}_S^\infty \setminus \mathbb{F}_S^=$

$P_{3,1},\dots,P_{3,n(3)}$ such that $\{\bar{P}_{3,1},\dots,\bar{P}_{3,n(3)}\} = \mathbb{F}_S^1 \setminus \mathbb{F}_S^\infty$

$P_1,\dots,P_k$ such that $\langle\bar{P}_1,\dots,\bar{P}_k\rangle \cap \mathbb{F}_S^1 = 0$

and $\langle\{P_1,\dots,P_k\} \cup \mathbb{F}_S^1\rangle = \mathbb{F}_S$.

j-proj$\circ\bar{\delta}$ for this realization is not unique by commutativity of the realization diagram. We define

$$(j\text{-proj}\circ\bar{\delta})(i_S(s),\ i_X(x)) = (\pi\bar{P}_j)(s) + c_{j,x}$$

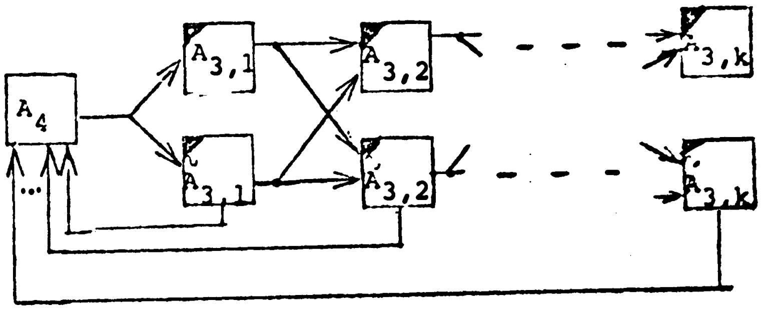
for all $P_j \in \mathbb{P}_S^2$ and arbitrary for $P_j \notin \mathbb{P}_S^2$. From $\pi\mathbb{P}_S^= \subseteq \mathbb{P}_S^=$, $\pi\mathbb{P}_S^\infty \subseteq \mathbb{P}_S^\infty$ and $\pi\mathbb{P}_S^2 \subseteq \mathbb{P}_S^1$. We conclude the form of our realization indicated in the theorem. From previous results we know that any realization of this form must use subsets of the spaces we used.

__Remark:__ We actually proved not only

$\langle\tilde{P}_{i,1},\dots,\tilde{P}_{i,m(i)}\rangle \subseteq \langle P_{i,1},\dots,P_{i,n(i)}\rangle$ but

$\{\tilde{P}_{i,1},\dots,\tilde{P}_{i,m(i)}\} \subset \{P_{i,1},\dots,P_{i,n(i)}\}$ which is a stronger result. However we did this by using a very uneconomical realization. The theorem holds, if we do not use $\mathbb{F}_S^1$ for coding, but a basis of $\mathbb{F}_S^1$ which contains a basis of $\mathbb{F}_S^\infty$ and one of $\mathbb{F}_S^{=}$. As a second remark we note that A_3 and its relation to A_4 in our realization in both forms can be refined further.

<u>Lemma 5.1</u>: In the realization of Theorem 5.1, the automaton A_3 can be replaced by



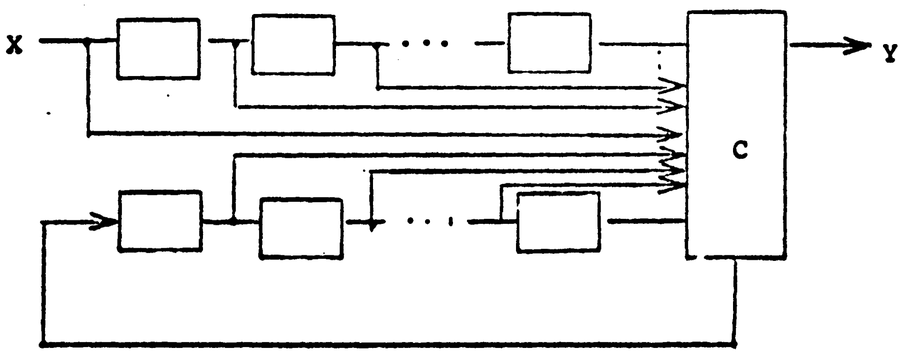
where $\mathbb{P}_S^k \neq \mathbb{P}_S^\infty$ and $\mathbb{P}_S^{k+1} = \mathbb{P}_S^\infty$

<u>Proof</u>: We know that $\pi\mathbb{P}_S^n \subseteq \mathbb{P}_S^{n-1}$ for all n. Hence we are through if we choose a basis of $\mathbb{F}_S^1$ which contains a basis for each $\mathbb{F}_S^n$, n=1,2,...,k.

Finally we want to point out that our results depend heavily on the definition chosen for "realization". In the case of linear realization this is justified by the fact that if there is a linear realization in a more general sense, then there is one in the restricted sense we adopted. However, this is not true for realizations with linear parts, as we will show in the following.

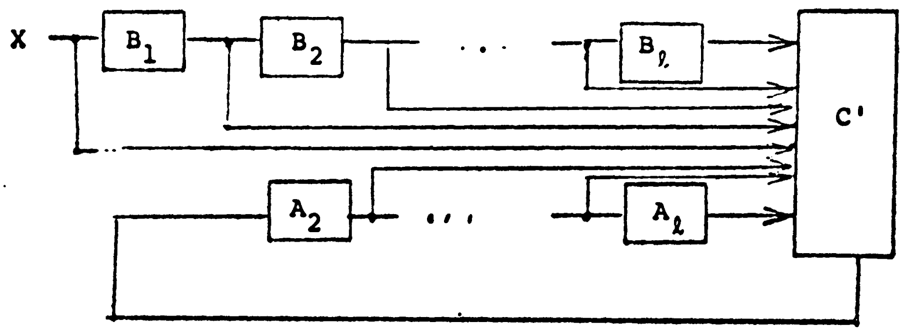
<u>Def. 5.1</u>: $A = (X,S,Y,\delta,\lambda)$ is realized <u>homomorphically</u> by $A' = (X',S',Y',\delta',\lambda')$, if A is a homomorphic image of a subautomaton of A'.

Friedman in [ ] has shown that any finite automaton can be realized homomorphically by a composition of the form



where C is a mapping into $Y \times K$ and the connections are interpreted a little differently as we do in this paper.

However, if we rearrange the picture we obtain for the state behavior:



where $A_i = (K, K, \{\bar{y}\}, \delta_A, \lambda_A)$, $\delta_A(k_2, k_1) = k_2$

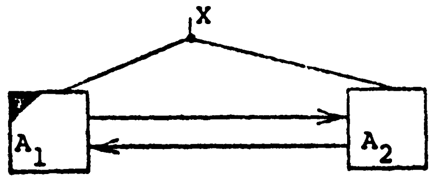
and $B_i = (X, X, \{\tilde{y}\}, \delta_B, \lambda_B)$, $\delta_B(x_2, x_1) = x_2$ and

$C' = (K^{\ell-1} \times X^\ell, K, \{\hat{y}\}, \delta_{C'}, \lambda_{C'})$,

$C'((k_2, \ldots, k_{\ell-1}, x_1, \ldots, x_\ell, x), k_1) = C(k_1, \ldots, k_\ell, x_1, \ldots, x_\ell, x)$.

All parts except possibly C' are linearly realizable. Hence we have the following result.

<u>Theorem 5.2</u>: For every automaton (X,S,Y,δ,λ) there is a homomorphic realization of the form



where A_2 has only $|X|$ states, i.e. in this realization all but one coordinates are linear.

Universitat Dortmund

West Germany

46 Dortmund

LINEAR REALIZATION OF FINITE AUTOMATA

Bernd Reusch

LINEAR REALIZATION OF FINITE AUTOMATA

The theory of linear automata has attracted people at least since the publications of Huffman [17,18] and special properties of linear automata are given in many papers. Most of them are summarized in books [8,10,22]. The problem whether an automaton can be represented by a linear one if given in some other form was introduced quite early. Elspas [6] was one of the first to attack the problem by using special properties of the state graphs of linear automata. Many others followed that line, like Kandzia [19], Wang [30], Yau and Wang [31], Srinivasan [27], Gill in many papers which are summarized in [8] and quite a number of others. Several other techniques are used by Cohn and Even [1], Mariono [20], Hartmanis [13], Gill [9] and others, some of which are summarized by Davis in [3]. All of the mentioned papers put restrictions on the automata or the underlying field and most of them use some search technique.

Our paper belongs to a series of publications which have their roots in a paper of Davis and Brzozowski [4]. They for the first time used partitions to find linear realizations of permutation automata over $GF(2) = \{0,1\}$. Starting from their ideas the author developed the concept of ordered or indexed partitions and in [21] gave an algorithm for linear realization of arbitrary finite automata over arbitrary $GF(p)$. This algorithm included some searching and in [22] the algorithm was improved. In the meantime

Abstract:

This paper presents a method for linear realization of automata when such exists. The basic tools employed are indexed partitions (as a way of writing certain mappings) and predecessor operations. The indexed partitions are studied in some detail and their connection to codes is explained. The method itself is simple and straightforward and contains no search steps. Only standard techniques of linear algebra, such as solving linear equations, are used. Examples are given to show the advantage of our method compared with previously known ones.

1. Statement of the problem.

In order to make our paper more or less self-contained, we give some basic definitions and facts in this section.

Def. 1.1: $M = (X,S,Y,\delta,\lambda)$ is a finite automaton if

1. X,S,Y are finite nonempty sets
2. δ and λ are mappings:

$$Y \xleftarrow{\lambda} X \times S \xrightarrow{\delta} S$$

The mappings δ and λ are extended to range over $S \times X^+$ in a natural way, where X^+ is the set of all nonempty finite strings (or words) of symbols from X.

Def. 1.2:
1. $\delta^+(s,x) := \delta(s,x)$ for all $s \in S$ and $x \in X$.
   $\delta^+(s,\bar{x}x) := \delta(\delta^+(s,\bar{x}),x)$ for all $s \in S$, $\bar{x} \in X^+$ and $x \in X$.
2. $\lambda^+(s,x) := \lambda(s,x)$ for all $s \in S$ and $x \in X$.
   $\lambda^+(s,\bar{x}x) := \lambda(\delta^+(s,\bar{x}),x)$ for all $s \in S$, $\bar{x} \in X^+$ and $x \in X$.

Since no confusion will occur we identify δ with δ^+ and λ with λ^+.

We define homomorphisms and isomorphisms for automata in the very same way as for other algebraic structures.

Def. 1.3: $M' = (X,S',Y,\delta',\lambda')$ is a homomorphic image of $M = (X,S,Y,\delta,\lambda)$ if there are bijections h_X and h_Y and a surjection h_S such that the following diagram commutes:

Walter [29] noticed that indexed partitions were nothing but a way of writing certain mappings and used this knowledge to simplify notation and give an algorithm based on [22]. He even did the job for arbitrary rings with unit at no extra cost. Walter's remark was used in [24] to extend the necessary and sufficient conditions for linearity over a field in [23] to necessary and sufficient conditions for linearity over rings (even infinite). The last paper in that line known to the author is [5], where Eichner reformulates more or less the results of [24] for fields and uses the ideas of [22,23,24] to solve the question for the <u>possible</u> underlying fields. Stucky [28] also worked on this problem and got partial results using a different approach.

This paper may be viewed a step in the line of [21,14]. We give a complete theory and a simple algorithm for linear realization of finite automata over a given finite field. In Section 1 we state our problem carefully, since there is some confusion about definitions, as can be learned from [14].

To define realization of automata by others, we have to use the concept of subautomaton.

Def. 1.5: $M' = (X',S',Y',\delta',\lambda')$ is a subautomaton of $M = (X,S,Y,\delta,\lambda)$ if $X' \subseteq X$, $S' \subseteq S$, $Y' \subseteq Y$ and δ' and λ' are restrictions of δ and λ' to these sets.

Def. 1.6: 1) M' is a realization of M, if M' has a subautomaton M" which is equivalent to M.

2) M' is a homomorphic realization of M, if M' has a subautomaton M" such that M is a homomorphic image of M".

3) M' is an isomorphic realization of M, if M' has a subautomaton M" which is isomorphic to M.

Clearly 3) implies 2) and 2) implies 1). In special cases the other direction also holds.

Def. 1.7: M is minimal (or reduced) if no two different states of M are equivalent.

Lemma 1.2: If M is minimal and M' a realization of M, then M' is a homomorphic realization of M.

For a special class of automata, namely the linear automata defined below, we can restrict ourselves to isomorphic realizations.

$$\begin{array}{ccccccc}
Y & \xleftarrow{\lambda} & S & \times & X & \xrightarrow{\delta} & S \\
\downarrow h_Y & & \downarrow h_S & & \downarrow h_X & & \downarrow h_S \\
Y' & \xleftarrow{\lambda'} & X' & \times & S' & \xrightarrow{\delta'} & S'
\end{array}$$

The triple (h_X, h_S, h_Y) is called a homomorphism of M onto M'.

Isomorphisms are homomorphisms where h_S is bijective.

Another important concept is the equivalence of automata. Roughly automata are equivalent if there is no "experiment" to distinguish them. More precisely:

<u>Def. 1.4</u>: Let $M = (X,S,Y,\delta,\lambda)$ and $M' = (X',S',Y',\delta',\lambda')$ be finite automata (not necessarily distinct).

1. $s \in S$ and $s' \in S'$ are equivalent $(s \sim s')$ if there are bijections h_X and h_Y from X to X' and Y to Y' respectively. Such that $h_Y(\lambda(s,x)) = \lambda'(s',h_X(x))$ for all $x \in X^+$
2. M and M' are equivalent if for every $s \in S$ there is a $s' \in S'$ such that $s \sim s'$ and for every $s' \in S'$ there is a $s \in S$ such that $s' \sim s$.

Homomorphisms and equivalence are related by the following lemma.

<u>Lemma 1.1</u>: $M = (X,S,Y,\delta,\lambda)$ and $M' = (X',S',Y',\delta',\lambda')$ are equivalent iff they have a common homomorphic image.

is an isomorphic realization of $M = (X,S,Y,\delta,\lambda)$ if there are injections i_X, i_S and i_Y of X into X, S into S' and Y into Y' resp. such that the following diagram commutes:

$$\begin{tikzcd}
Y \arrow[d, "i_Y"'] & S \arrow[l, "\lambda"'] \arrow[d, "i_S"] & \times & X \arrow[r, "\delta"] \arrow[d, "i_X"] & S \arrow[d, "i_S"] \\
Y' & S' \arrow[l, "\lambda'"'] & \times & X' \arrow[r, "\delta'"] & S'
\end{tikzcd}$$

In the linear case we can restrict ourselves to $X = K^m$, $S = K^n$ and $Y = K^\ell$. Hence we obtain the following formulation of linear realizability.

The automaton $M = (X,S,Y,\delta,\lambda)$ has an isomorphic linear realization over K iff there are natural numbers ℓ,m,n and injections of X into K^m, S into K^n and Y into K such that the following diagram commutes

$$\begin{tikzcd}
Y \arrow[d, "i_Y"'] & S \arrow[l, "\lambda"'] \arrow[d, "i_S"] & \times & X \arrow[r, "\delta"] \arrow[d, "i_X"] & S \arrow[d, "i_S"] \\
K^l & K^n \arrow[l, "\bar{\lambda}"'] & \times & K^m \arrow[r, "\bar{\delta}"] & K^n
\end{tikzcd}$$

and the mappings $\bar{\delta}$ and $\bar{\lambda}$ are linear.

The linearity or non-linearity of $\bar{\delta}$ and $\bar{\lambda}$ depends primarily on the chosen i_Y, i_S and i_X. If we look for linear realizations for a given automaton over a given field we hence must look for an algorithm to produce i_Y , i_S and i_X properly.

<u>Def. 1.8</u>: $M = (X,S,Y,\delta,\lambda)$ is a linear automaton over the finite field K if

1. X,S and Y are finite dimensional vector spaces over K.
2. There are linear mappings

   $A: S \to S$

   $B: X \to S$

   $C: S \to Y$

   $D: X \to Y$

   such that $\delta(s,x) = As + Bx$, $\lambda(s,x) = Cs + Dx$ for all $s \in S$ and $x \in X$.

M' is a linear realization of M if M' is a linear automaton and a realization of M. For linear realizations the following can be shown:

<u>Theorem 1.1</u>: Let M be a minimal automaton. If there is a linear realization of M, then there is also an isomorphic linear realization of M.

Since there is an algorithm to find the minimal equivalent automaton for every given automaton, the question of linear realization of automata is reduced to the question of isomorphic linear realization (see comment at the end of Section 4.).

To obtain more insight into this question, let us first reformulate the definition of isomorphic realization. $M' = (X',S',Y',\delta',\lambda')$

We call the T^k's the blocks of T. There may be empty blocks. Given a mapping T, we define an indexed partition $T = (T^k)_{k \varepsilon T}$ by saying $m \varepsilon M$ is in T^k iff $T(m) = k$, and given an indexed partition $T = (T^k)_{k \varepsilon K}$ we define a mapping T by saying $T(m) = k$ iff $m \varepsilon T^k$. We do not distinguish between mappings and indexed partitions in the sequel and use the definition which seems to be more convenient.
In definition 2.1 a set $\{T_1, \ldots, T_n\}$ of mappings is obtained from the mapping $\imath$. The converse construction will now be indicated.

<u>Def. 2.3:</u> Let T_j be a mapping of M into K for $j = 1, \ldots, n$, then

$$\imath(m) := (T_1(m), \ldots, T_n(m)) \quad \text{for all } m \varepsilon M.$$

Definitions 2.1 and 2.3 establish a one-one correspondence between ordered n-tuples of mappings T_j of M into K and mappings $\imath$ of M into K^n. Hence the following set will interest us.

<u>Def. 2.4:</u> $\mathbb{P}_M := \{T \mid T: M \longrightarrow K\}$

Our goal is to characterize those ordered n-tuples over $\mathbb{P}_M$, which correspond to (or for short "are") codes, i.e. elements of C_M.
To achieve this goal we use two operators α and β which map $\mathbb{P}_M$ and the powerset of $\mathbb{P}_M$ to the set of equivalence relations on M.

<u>Def. 2.5:</u> a) For all $T \varepsilon \mathbb{P}_M$ and $a,b \varepsilon M$ we define

$$a \equiv b(\alpha T) :\Longleftrightarrow T(a) = T(b)$$

2. Indexed partitions and codes.

Our discussion in the last section indicated the importance of the set

$\mathbb{C}_M = \{ \imath : M \longrightarrow K^n \mid \imath \text{ injective}, n = 1,2,\ldots, \}$,

being the set of codes which may be used in realizations, where M is X,S or Y and K a finite field. $\mathbb{C}_M$ has a very complicated structure, mainly because of the variable n and we would like to reduce the discussion to a more simple set. This can be done by breaking $\imath$ into pieces T_j, j = 1,...,n, where T_j is a mapping of M into K only.

<u>Def. 2.1</u>: Let $\imath$ be a mapping of M into K^n, then

$T_j := \text{j-proj} \circ \imath$,

where j-proj $\circ\, \imath$ means "first apply $\imath$ and then j-proj to the result" and $(\text{j-proj})(k_1,\ldots,k_n) = k_j$.

The mappings T_j are in a one-one correspondence to what we like to call indexed partitions, since this correspondence is analogous to the one between equivalence relations and ordinary partitions.

<u>Def. 2.2</u>: $T = (T^k)_{k \in K}$ is an indexed partition of M over K iff

1. $T^k \subseteq M$, for all $k \in K$
2. $T^k \cap T^l = \emptyset$, if $k \neq l$
3. $\bigcup_{k \in K} T^k = M$

Proof: a) $\beta(U \cup V) = \prod_{T \in U \cup V} \alpha T = (\prod_{T \in U} \alpha T)(\prod_{T \in V} \alpha T) = (\beta U)(\beta V)$.

c) If $a \equiv b(\beta V)$, then $a \equiv b(\alpha T)$ for all $T \in V$. Since $U \subseteq V$ we obtain $a \equiv b(\alpha T)$ for all $T \in U$ and hence $a \equiv b(\beta U)$. Thus $\beta U \geq \beta V$

d) $U \subseteq U \cap V$ and $V \subseteq U \cap V$ holds and hence by c) we get $\beta U \leq (U \cap V)$ and $\beta V \leq (U \cap V)$. From this we conclude $\beta U + \beta V \leq \beta(U \cap V)$.

So far we have not used the structure of the field K and our discussion is valid for arbitrary sets K. The codes used for linear realizations cannot be characterized without knowledge of K. For this reason we put some algebraic structure on P_M using the structure of K.

Def. 2.6: a) $(T_1 \oplus T_2)(a) := T_1(a) + T_2(a)$,
for all T_1, $T_2 \in P_M$ and all $a \in M$.
b) $(k \cdot T)(a) = k \cdot T(a)$,
for all $T \in P_M$, $k \in K$ and $a \in M$.

This is the usual construction for operations on mappings. In our case it puts a well known structure on P_M.

Theorem 2.2: P_M is a vector space over K.

Proof: by verifying the axioms.

b) For all $U \subseteq P_M$ we define

$$\beta U := \prod_{T \in U} \alpha T$$

α is a kind of "forget operator" since its meaning for indexed partitions is: forget the indices of blocks and remove all the empty blocks. The result is an ordinary partition. In defining β we used the usual definition of products of partitions or equivalence relations. The finest partition (or identity relation) is denoted by O (see [22]).

<u>Theorem 2.1</u>: $U = \{T_1,\dots,T_n\}$ is a code iff $\beta U = 0$

<u>Proof</u>: Let $\mathcal{r}$ be the mapping related to U according to definition 2.3.

Asume $a \equiv b(\beta U)$ and U a code. Then $a \equiv b(\alpha T_i)$ and hence $T_i(a) = T_i(b)$ for $i = 1,\dots,n$. We conclude $\mathcal{r}(a) = \mathcal{r}(b)$ and therefore a=b or $\mathcal{r}$ is not injective. Thus $\beta U = 0$.

Now asume $\beta U = 0$ and $\mathcal{r}(a) = \mathcal{r}(b)$. Then $T_i(a) = T_i(b)$ and hence $a \equiv b(\alpha T_i)$ for $i = 1,\dots,n$. Therefore $a \equiv b(\beta U)$ and since $\beta U = 0$ we obtain a=b. Thus $\mathcal{r}$ is injective.

The following properties of β might be of interest in view of this theorem.

<u>Lemma 2.1</u>: a) $\beta(U \cup V) = (\beta U)(\beta V)$

b) $\beta(U \cap V) \geq \beta U + \beta V$

c) $U \subseteq V \Longrightarrow \beta U \geq \beta V$

Since obviously $\tilde{k}_1 \oplus \tilde{k}_2 = \widetilde{k_1+k_2}$ and $k_i\tilde{k}_2 = \widetilde{k_ik_2}$, we identify $\tilde{k}$ and k. With this convention our lemma states that K itself is a subspace of P_M. We will be interested in the factor space of P_M modulo K.

<u>Def. 2.8</u>: $F_M := P_M/K$

We write $\overline{T_1} = \overline{T_2}$ if we want to express that T_1 and T_2 are mapped onto the same element of the factor space. T_1 and T_2 are called "equal up to shift" in that case, since:

<u>Lemma 2.4</u>: Let $T_1 = (T_1^j)_{j\varepsilon K}$, $T_2 = (T_2^i)_{i\varepsilon K}$.
$\overline{T_1} = \overline{T_2}$ iff there exists a $k\varepsilon K$ such that $T_1^j = T_2^{j+k}$ for all $j\varepsilon K$.

<u>Proof</u>: Asume $\overline{T_1} = \overline{T_2}$. Then there is a $k\varepsilon K$ such that $T_2 = T_1 \oplus \tilde{k}$. But then $T_1(a) = j$ iff $T_2(a) = (T_1\oplus\tilde{k})(a) = T_1(a)+k = j+k$. Hence $a\varepsilon T_1^j$ iff $a\varepsilon T_2^{j+k}$. Asume there exists a $k\varepsilon K$ such that $T_1^j = T_2^{j+k}$ for all $j\varepsilon K$. This means $a\varepsilon T_1^j$ iff $a\varepsilon T_2^{j+k}$ and hence $T_1(a) = j$ iff $T_2(a) = j+k$. From this we conclude $T_2 = T_1\oplus\tilde{k}$ and therefore $\overline{T_1} = \overline{T_2}$.

<u>Lemma 2.5</u>: a) $\alpha T = \alpha(k\cdot T) = \alpha(T\oplus l)$,
for all $T\varepsilon P_M$, $k,l\varepsilon K$ if $k \neq 0$.

b) $\beta\{T_1,\ldots,T_n\} = \beta\langle T_1,\ldots,T_n\rangle$, where $\langle T_1,\ldots,T_n\rangle$ is the subspace generated by $\{T_1,\ldots,T_n\}$.

For actual computing, it is of interest how "$\oplus$" and multiplication are performed on indexed partitions.

<u>Lemma 2.2:</u> Let $T_1 = (T_1^j)_{j\varepsilon K}$ and $T_2 = (T_2^j)_{j\varepsilon K}$ be indexed partitions and $k\varepsilon K$.

a) $(T_1 \oplus T_2) = (P^m)_{m\varepsilon K}$ iff

$$P^m = \bigcup_{i+j=m} (T_1^i \cap T_2^j)$$

b) $k \cdot T_1 = (P^m)_{m\varepsilon K}$ iff $P^m = T_1^j$ such that $k \cdot j = m$.

<u>Proof:</u> a) $(T_1 \oplus T_2)(a) = m$ iff $T_1(a) + T_2(a) = m$. Hence $a\varepsilon P^m$ iff $a\varepsilon T_1^i$ <u>and</u> $a\varepsilon T_2^j$ such that $i+j=m$. The latter is true iff $a\varepsilon T_1^i \cap T_2^j$ such that $i+j=m$.

b) $(k \cdot T_1)(a) = m$ iff $k \cdot T_1(a) = m$. Hence $a\varepsilon P^m$ iff $a\varepsilon T_1^j$ such that $k \cdot j = m$.

The constant mappings play an important role in our further discussion.

<u>Def. 2.7:</u> $\tilde{k}(a) = k$ for all $k\varepsilon K$ and $a\varepsilon M$

The constant mappings are easily realized as indexed partitions. The block indexed by k contains the whole set M and all other blocks are empty. From this and lemma 2.2 we get immediately the following:

<u>Lemma 2.3:</u> If $V_M := \{\tilde{k}/k\varepsilon K\}$, then V_M is a subspace of $\mathcal{P}_M$.

<u>Def. 2.9:</u> $T_a(b) = \begin{cases} 1, \text{ if } a=b \\ 0, \text{ otherwise} \end{cases}$, for all $a,b\varepsilon M$

Viewed as indexed partition, T_a has a very simple structure: the block indexed by 1 consists of a, all other elements of M are placed in the block indexed by 0. All other blocks are empty.

<u>Lemma 2.6</u>: $T = \bigoplus_{a\varepsilon M} T(a)T_a$, for all $T\varepsilon P_M$

<u>Proof</u>: $(\bigoplus_{a\varepsilon M} T(a)T_a)(b) = \bigoplus_{a\varepsilon M} T(a)T_a(b)$ for all $b\varepsilon M$.

Since $T_a(b) = 0$ if $a \neq b$ and $T_b(b) = 1$, we obtain

$(\bigoplus_{a\varepsilon M} T(a)T_a)(b) = T(b)T_b(b) = T(b)$

Hence $\bigoplus_{a\varepsilon M} T(a)T_a = T$ as was to be shown.

This lemma shows that P_M is generated by $\{T_a|a\varepsilon M\}$. Asume $T_b = \bigoplus_{a\varepsilon M} k_aT_a$ for some $b\varepsilon M$ and $k_a\varepsilon K$. Then $1 = T_b(b) =$ $= \bigoplus_{a\varepsilon M} k_aT_a(b) = k_b$ and for $c \neq b$ we obtain $0 = T_b(c) = \bigoplus_{a\varepsilon M} k_aT_a(c) = k_c$
Hence no T_b is a linear combination of $\{T_a|a\varepsilon M, a \neq b\}$. This means:

<u>Theorem 2.3</u>: $\{T_a|a\varepsilon M\}$ is a basis of P_M.

<u>Corollary 2.1</u>: P_M has dimension $|M|$, where $|M|$ denotes the cardinality of M.

<u>Proof</u>: a) $a \equiv b(\alpha(kT))$ iff $(kT)(a) = (kT)(b)$

iff $k \cdot T(a) = k \cdot T(b)$ iff $T(a) = T(b)$

iff $a \equiv b(\alpha T)$.

Hence $\alpha T = \alpha(kT)$.

$\alpha T = \alpha(T+1)$ is proved similarly.

b) $\beta\{T_1,\dots,T_n\} \geq \beta\langle T_1,\dots,T_n\rangle$ follows from lemma 2.1.

To prove $\beta\{T_1,\dots,T_n\} \leq \beta\langle T_1,\dots,T_n\rangle$ it is sufficient to show $\alpha T \geq \beta\{T_1,\dots,T_n\}$ for all $T \varepsilon \langle T_1,\dots,T_n\rangle$.

If $T \varepsilon \langle T_1,\dots,T_n\rangle$, then $T = \bigoplus_{i=1}^{n} k_i T_i$ for some $k_i \varepsilon K$, $i = 1,\dots,n$.

Let $a \equiv b(\beta\{T_1,\dots,T_n\})$. From this we know $T_i(a) = T_i(b)$ for all $i = 1,\dots,n$ and hence $(\bigoplus_{i=1}^{n} k_i T_i)(a) = (\bigoplus_{i=1}^{n} k_i T_i)(b)$.

This means $T(a) = T(b)$ and therefore $a \equiv b(\alpha T)$. Thus we have proven $\alpha T \geq \beta\{T_1,\dots,T_n\}$.

Because of part a) of this lemma, we are able to extend α and β to classes $\bar{T}$ of indexed partitions and we will not use new symbols. It will turn out that in order to obtain linear realizations we have to look for the kernels of certain linear mappings of $\mathbb{P}_Y$ and $\mathbb{P}_S$. Hence we have to write linear mappings in some way and the usual way to do that is to represent $\mathbb{P}_Y$ and $\mathbb{P}_S$ as vector spaces of n-tuples over K and the linear mappings as matrices. For this purpose we select a basis of $\mathbb{P}_M$ which is easy to handle.

3. Linearity of single coordinates.

In this section we want to use indexed partitions and the structure on the sets of indexed partitions over X,S and Y to characterize linearity of single coordinates in given realizations. We fix the notation P, P_1, P_i etc. to be elements of $\mathbb{P}_S$; Q, Q_1, Q_i etc. to be elements of $\mathbb{P}_X$; and R, R_1, R_i etc. to be elements of $\mathbb{P}_Y$.

The vector space structures of $\mathbb{P}_X$, $\mathbb{P}_S$ and $\mathbb{P}_Y$ are linked in a way which is specific for the automaton under consideration. To show this we introduce the following mappings.

Def. 3.1: 1. $\delta_x(s) := \delta(s,x), \lambda_x(s) := \lambda(s,x)$ for all $x \varepsilon X$ and $s \varepsilon S$.

2. $\pi_x P := P \circ \delta_x$, $\tau_x R := R \circ \lambda_x$ for all $P \varepsilon \mathbb{P}_S$, $R \varepsilon \mathbb{P}_Y$ and $x \varepsilon X$

We call π_x and τ_x the "predecessor operators with respect to x" according to properties given in the next lemma. Predecessor operators for indexed partitions have a very simple meaning: Replace every element in every block by the set of its predecessors with respect to x. Empty sets may appear.

Lemma 3.1: a) If $P \varepsilon \mathbb{P}_S$, $P = (P^i)_{i \varepsilon K}$, then

$\pi_x P = (\pi_x P^i)_{i \varepsilon K}$, where

$\pi_x P^i := \{s | \delta(s,x) \varepsilon P^i\}$

__Corollary 2.2:__ F_M has dimension $|M|-1$.

__Corollary 2.3:__ $\{\overline{T_a} | a\epsilon M,\ a\neq b\}$ is a basis of F_M for every $b\epsilon M$.

We will use the knowlegde about this basis of F_M in section 5.

for all $s \varepsilon S$. Therefore $\pi_{x_1 x_2} P = (\pi_{x_2} \circ \pi_{x_1}) P$.

b) is proven similarly to a).

c) $\pi_x(k_1P_1 \oplus k_2P_2)(s) = ((k_1P_1 \oplus k_2P_2) \circ \delta_x)(s) =$

$= (k_1P_1 \oplus k_2P_2)(\delta(s,x)) = k_1P_1(\delta(s,x)) + k_2P_2(\delta(s,x)) =$

$= k_1(P_1 \circ \delta_x)(s) + k_2(P_2 \circ \delta_x)(s) = k_1\pi_xP_1(s) + k_2\pi_xP_2(s) =$

$= (k_1\pi_xP_1 \oplus k_2\pi_xP_2)(s)$ for all $s \varepsilon S$ and $x \varepsilon X$

The statement for τ_x is proven by replacing δ by λ

It follows easily from lemma 3.1 that the constant mappings are not changed if we apply π_x or τ_x to them. Hence π_x and τ_x induce linear mappings $\overline{\pi_x}$ and $\overline{\tau_x}$ on the factor spaces F_S and F_Y.

$$\begin{tikzcd} P_Y \arrow[r, "\tau_x"] & P_S \arrow[loop right, "\pi_x"] \end{tikzcd}$$

$$\begin{tikzcd} F_Y \arrow[r, "\overline{\tau_x}"] & F_S \arrow[loop right, "\overline{\pi_x}"] \end{tikzcd}$$

The structures on P_S and P_Y are now linked by sets of linear mappings which were derived from properties of the given automaton. The connection to P_X we have in mind cannot be defined on the whole sets P_S and P_Y but on subsets only which depended on the predecessor operators and hence again on the structure of the automaton itself.

<u>Def. 3.2:</u> a) $P_S^1 := \{P \varepsilon P_S \mid \overline{\pi_x P} = \overline{\pi_{x'} P}$, for all $x,x' \varepsilon X\}$

b) $P_Y^1 := \{R \varepsilon P_Y \mid \overline{\tau_x R} = \overline{\tau_{x'} R}$, for all $x,x' \varepsilon X\}$

b) If $R \varepsilon P_Y$, $R = (R^j)_{j \varepsilon K}$, then

$\tau_x R = (\tau_x R^j)_{j \varepsilon K}$, where

$\tau_x R^j := \{s | \lambda(s,x) \varepsilon R^j\}$

Proof: We prove part a), the other statement follows similarly. From the definitions we know

$\pi_x P(s) = (P \circ \delta_x)(s) = P(\delta(s,x))$. This gives $(\pi_x P)(s) = i$ iff $P(\delta(s,x)) = i$, which is equivalent to $\delta(s,x) \varepsilon P^i$, which by definition means that $s \varepsilon \pi_x P^i$.

The predecessor operators will be of great importance in our discussion of linear realizability. For this reason we want to represent them in a convenient fashion. This will be easy because of the following lemma.

Lemma 3.2: a) $\pi_{x_1 x_2} P = (\pi_{x_2} \circ \pi_{x_1}) P$

b) $\tau_{x_1 x_2} R = (\pi_{x_2} \circ \tau_{x_1}) R$

c) τ_x and π_x are linear mappings for all $x \varepsilon X^*$

Proof: a) $\pi_{x_1 x_2} P = P \circ \delta_{x_1 x_2}$. Hence

$\pi_{x_1 x_2} P(s) = (P \circ \delta_{x_1 x_2})(s) = P(\delta(s, x_1 x_2)) =$

$= P(\delta(\delta(s,x_1),x_2)) = P(\delta_{x_2}(\delta_{x_1}(s))) = (P \circ \delta_{x_2} \circ \delta_{x_1})(s) =$

$= ((\pi_{x_2} \circ \pi_{x_1}) P)(s)$

Lemma 3.4: a) $\overline{\emptyset_U R} = \overline{\emptyset_{U'} P}$ for all $P \in P_S^1$ and all possible choices of U and U'.

b) $\overline{\Psi_V R} = \overline{\Psi_{V'} R}$ for all $R \in P_Y^1$ and all possible choices of V and V'.

Proof: We prove part a).

There is a $k \in K$, such that $U = U' \oplus k$. Hence for all $x \in X$ we obtain $(\emptyset_U P)(x) = i$ iff $\pi_X P = U \oplus i = U' \oplus (k+i)$ hence iff $(\emptyset_{U'} P)(x) = k+i$. Thus $\emptyset_U P = \emptyset_{U'} P \oplus k$ and therefore $\overline{\emptyset_U P} = \overline{\emptyset_{U'} P}$.

This property together with the remark $\emptyset_U(P \oplus k) = \emptyset_U P \oplus k$ gives us unique mappings $\emptyset$ and Ψ:

$$F_S^1 \xrightarrow{\ \emptyset\ } F_X \xleftarrow{\ \Psi\ } F_Y^1 .$$

Lemma 3.5: $\emptyset$ and Ψ are linear mappings.

Proof: We have to show $\emptyset(k_1\overline{P_1} \oplus k_2\overline{P_2}) = k_1(\emptyset\overline{P_1}) \oplus k_2(\emptyset\overline{P_2})$, which is done if we find suitable U, U_1 and U_2 such that
$\emptyset_U(k_1P_1 \oplus k_2P_2) = k_1(\emptyset_{U_1}P_1) \oplus k_2(\emptyset_{U_2}P_2)$.
Let us take U_1 and U_2 such that $\emptyset_{U_1}P_1$ and $\emptyset_{U_2}P_2$ are defined. In this case $\emptyset_U(k_1P_1 \oplus k_2P_2)$ is defined also, if $U = k_1U_1 \oplus k_2U_2$. We obtain the following chain of equivalent statements.

Since the π_x's and τ_x's are linear mappings which leave the constants unchanged, we have immediately the following.

<u>Lemma 3.3</u>: $\mathbb{P}_S^1$ and $\mathbb{P}_Y^1$ are subspaces of $\mathbb{P}_S$ and $\mathbb{P}_Y$ resp. containing K.

Hence we may define

$$\mathbb{F}_S^1 := \mathbb{P}_S^1 / K \quad \text{and} \quad \mathbb{F}_Y^1 := \mathbb{P}_Y^1 / K$$

We will define linear mappings from $\mathbb{F}_S^1$ and $\mathbb{F}_Y^1$ into $\mathbb{F}_X$. This is possible according to the next definition and lemma.

<u>Def. 3.3</u>: a) For all $U \epsilon \mathbb{P}_S$ and $P \epsilon \mathbb{P}_S^1$ such that $\bar{U} = \overline{\pi_X P}$ we define $\emptyset_U P$ by: $(\emptyset_U P)(x) = i$ iff $\pi_x P = U \oplus i$, for all $x \epsilon X$ and $i \epsilon K$.

b) For all $V \epsilon \mathbb{P}_Y$ and $R \epsilon \mathbb{P}_Y^1$ such that $\bar{V} = \overline{\tau_X R}$ we define $\Psi_V R$ by: $(\Psi_V R)(x) = j$ iff $\tau_x R = V \oplus j$ for all $x \epsilon X$ and $j \epsilon K$.

We have defined mappings from $\mathbb{P}_S^1$ and $\mathbb{P}_Y^1$ to $\mathbb{P}_X$.

$$\mathbb{P}_S^1 \xrightarrow{\emptyset_U} \mathbb{P}_X \xleftarrow{\Psi_V} \mathbb{P}_Y^1$$

The next result shows that the choice of U and V is of limited influence only. We prove that for different U and U' the images $\emptyset_U P$ and $\emptyset_{U'} P$ are equal to shift and similarly for $\Psi_V R$ and $\Psi_{V'} R$.

<u>Theorem 3.1</u>: A) There is a realization of (X,S,Y,δ,λ) such that $i\text{-proj} \circ \bar{\delta}$ is linear

iff 1. there are $\{Q_1,\ldots,Q_m\} \subseteq \mathbb{P}_X$, $\{P_1,\ldots,P_n\} \subseteq \mathbb{P}_S$ and $\{R_1,\ldots,R_l\} \subseteq \mathbb{P}_Y$ such that $\beta\{Q_1,\ldots,Q_m\} = 0$, $\beta\{P_1,\ldots,P_n\} = 0$ and $\beta\{R_1,\ldots,R_l\} = 0$

2. $P_i \in \mathbb{P}_S^1$

3. $\overline{\pi_x P_i} \in \langle \overline{P_1},\ldots,\overline{P_n} \rangle$ for some $x \in X$ and hence for all $x \in X$ by 2.

4. $\emptyset\, \overline{P_i} \in \langle \overline{Q_1},\ldots,\overline{Q_m} \rangle$

B) There is a realization of (X,S,Y,δ,λ) such that $j\text{-proj} \circ \bar{\lambda}$ is linear

iff 1. as in A)

2. $R_j \in \mathbb{P}_Y^1$

3. $\overline{\tau_x R_j} \in \langle \overline{P_1},\ldots,\overline{P_n} \rangle$ for some $x \in X$ and hence for all $x \in X$ by 2.

4. $\Psi \overline{R_j} \in \langle \overline{Q_1},\ldots,\overline{Q_m} \rangle$

<u>Proof</u>: We prove A)

Let a realization be given such that $i\text{-proj} \circ \bar{\delta}$ is linear. Part 1. obviously holds. By commutativity of the realization-diagram we obtain the following for all $s \in S$ and $x \in X$:

$(\pi_x P_i)(s) = (P_i \circ \delta_x)(s) = (i\text{-proj} \circ \dot{h}_S \circ \delta_x)(s) =$

$= (i\text{-proj} \circ \dot{h}_S \circ \delta)(s,x) = (i\text{-proj} \circ \bar{\delta})(\dot{h}_S(s), \dot{h}_X(x)) =$

$$(1)\quad (k_1(\emptyset_{U_1}P_1)\oplus k_2(\emptyset_{U_2}P_2))(x) = i$$

$$(2)\quad k_1(\emptyset_{U_1}P_1)(x) + k_2(\emptyset_{U_2}P_2)(x) = i$$

$$(3)\quad (\emptyset_{U_1}P_1)(x) = i_1,\ (\emptyset_{U_2}P_2)(x) = i_2,\ k_1i_1 + k_2i_2 = i$$

$$(4)\quad \pi_x P_1 = U_1\oplus i_1,\ \pi_x P_2 = U_2\oplus i_2,\ k_1i_1 + k_2i_2 = i$$

$$(5)\quad k_1\pi_x P_1\oplus k_2\pi_x P_2 = U\oplus i$$

$$(6)\quad \pi_x(k_1P_1\oplus k_2P_2) = U\oplus i$$

$$(7)\quad (\emptyset_U(k_1P_1\oplus k_2P_2))(x) = i$$

The equivalence of (1) and (7) implies

$\emptyset_U(k_1P_1\oplus k_2P_2) = k_1(\emptyset_{U_1}P_1)\oplus k_2(\emptyset_{U_2}P_2)$ and hence

$\emptyset(k_1\overline{P_1}\oplus k_2\overline{P_2}) = k_1(\emptyset\overline{P_1})\oplus k_2(\emptyset\overline{P_2})$

We are now able to state the main theorem of this section. Let us first recall the "realization-diagram". The automaton (X,S,Y,δ,λ) is realized over $\dot{K}$ iff the following diagram commutes:

$$\begin{tikzcd}
Y \arrow[d, "i_Y"'] & S \times X \arrow[l, "\lambda"'] \arrow[r, "\delta"] \arrow[d, "i_S \times i_X"] & S \arrow[d, "i_S"] \\
K^l & K^n \times K^m \arrow[l, "\bar{\lambda}"] \arrow[r, "\bar{\delta}"'] & K^n
\end{tikzcd}$$

The mappings i_X, i_S and i_Y are injections (or codes) and R_j = j-proj $\circ\, i_Y$, P_i = i-proj $\circ\, i_S$, Q_r = r-proj $\circ\, i_X$.

$P_1,\dots,P_n$ and let $\imath_X: X \longrightarrow K^{m+1}$ be given by $Q_1,\dots,Q_m$ for the first m coordinates and $Q_{m+1}(x) = k_x$ for all $x \in X$. Then we can show in a way similar to the first part of the proof that

$$(i\text{-proj} \circ \bar{\delta})(\imath_S(s), \imath_X(x)) =$$

$$= \sum_{j=1}^{n} a_{ij}(j\text{-proj} \circ \imath_S)(s) + \sum_{p=1}^{m+1} b_{ip}(p\text{-proj} \circ \imath_X)(x)$$

for all commutative completions $\bar{\delta}$, where $b_{i(m+1)} = 1$. For elements of K^{n+m+1} not of the form $(\imath_S(s), \imath_X(x))$ we use the equation above as definition of $i\text{-proj} \circ \bar{\delta}$.

We will use theorem 3.1 to formulate an algorithm for linear realization of automata in case the code for Y is already given. Our next theorem gives a more simple necessary and sufficient condition for linearity of single coordinates. It will be used to develop a general algorithm for linear realization of automata.

<u>Theorem 3.2</u>: A) There is a realization of (X,S,Y,δ,λ) such that $i\text{-proj} \circ \bar{\delta}$ is linear

iff there are $\{P_1,\dots,P_k\} \subseteq P_S$, $P_i \in P_S$ such that

$$\overline{\pi_x P_i} = \bigoplus_{j=1}^{k} a_{ij} \overline{P_j} \text{ for all } x \in X.$$

B) There is a realization of (X,S,Y,δ,λ) such that $j\text{-proj} \circ \bar{\lambda}$ is linear

iff there are $\{P_1,\dots,P_k\} \subseteq P_S$, $R_j \in P_Y$ such that

$$\overline{\tau_x R_j} = \bigoplus_{r=1}^{k} c_{rj} \overline{P_r} \text{ for all } x \in X.$$

$$=\sum_{j=1}^{n} a_{ij}\ (j\text{-proj} \circ \imath_S)(s) + \sum_{p=1}^{m} b_{ip}\ (p\text{-proj} \circ \imath_X)(x) =$$

$$=\sum_{j=1}^{n} a_{ij}P_j(s) + \sum_{p=1}^{m} b_{ip}Q_p(x) =$$

$$=(\bigoplus_{j=1}^{n} a_{ij}P_j)(s) + (\bigoplus_{p=1}^{m} b_{ip}Q_p)(x).$$

More explicitly, we obtained $\pi_x P_i = \bigoplus_{j=1}^{n} a_{ij}P_j \oplus h_{ix}$, where h_{ix} is a constant depending on i and x but not on s, namely $h_{ix} = (\bigoplus_{p=1}^{m} b_{ip}Q_p)(x)$. But then $\overline{\pi_x P_i} = \overline{\pi_{x'} P_i} =$ $= \bigoplus_{j=1}^{n} a_{ij}\dot{\overline{P_j}}$, which proves conditions 2. and 3.

To prove 4., set $U = \bigoplus_{j=1}^{n} a_{ij}P_j$ and look for $\emptyset_U P_i$. This gives: $(\emptyset_U P_i)(x) = k$ iff $k = h_{ix}$, which means by the discussion above $\emptyset_U P_i = \bigoplus_{p=1}^{m} b_{ip}Q_p$ and hence $\emptyset\overline{P_i} = \bigoplus_{p=1}^{m} b_{ip}\overline{Q_p}$.

This completes the first part of the proof. To prove the converse, let $\{Q_1,\dots,Q_m\}$, $\{P_1,\dots,P_n\}$ and $\{R_1,\dots,R_l\}$ be given such that conditions 1. to 4. hold. We have to define $\bar{\delta}$ such that the realization diagram commutes and $i\text{-proj} \circ \bar{\delta}$ is linear. By asumption we know

$$\pi_x P_i(s) = (\bigoplus_{j=1}^{n} a_{ij}P_j)(s) + (\bigoplus_{p=1}^{m} b_{ip}Q_p)(x) + k_x, \text{ where}$$

k_x is a constant depending on x. Let $\imath_S$ be given by

4. Linear realizations.

A given automaton A is linearly realizable over K, iff there is a realization induced by the sets $\{Q_1,\dots,Q_m\}$, $(P_1,\dots,P_n\}$ and $\{R_1,\dots,R_l\}$, such that the conditions of theorem 3.1 hold for all P_i, $i = 1,\dots,n$ and R_j, $j = 1,\dots,l$.
From this observation we obtain immediately the following result.

<u>Theorem 4.1</u>: The automaton (X,S,Y,δ,λ) is linearly realizable over K

iff 1. there are $\{Q_1,\dots,Q_m\} \subseteq \mathbb{P}_X$, $\{P_1,\dots,P_n\} \subseteq \mathbb{P}_S$ and $\{R_1,\dots,R_l\} \subseteq \mathbb{P}_Y$ such that $\beta\{Q_1,\dots,Q_m\} = 0$ $\beta\{P_1,\dots,P_n) = 0$ and $\beta\{R_1,\dots,R_l\} = 0$.

2. $\langle P_1,\dots,P_n\rangle \subseteq \mathbb{P}_S^1$
$\langle R_1,\dots,R_l\rangle \subseteq \mathbb{P}_Y^1$

3.
$$\begin{tikzcd} \langle \bar{R}_1,\dots,\bar{R}_l\rangle \arrow[rr, "\overline{\tau_x}"] \arrow[dr, "\gamma"'] & & \langle \bar{P}_1,\dots,\bar{P}_n\rangle \arrow[dl, "\phi"] \arrow[loop right, "\overline{\pi_x}"] \\ & \langle \bar{Q}_1,\dots,\bar{Q}_m\rangle & \end{tikzcd}$$

for all $x \in X$.

This theorem implies a simple algorithm for linear realization, if the outputcode is already given by $\{R_1,\dots,R_l\}$.

Proof: We prove A)

If there is a realization such that i-proj $\circ\ \bar{\delta}$ is linear, then the condition follows from 2. and 3. of theorem 3.1

To show the converse we have to find proper sets $\{Q_1,\dots,Q_m\}$ $\{P_1,\dots,P_n\}$ and $\{R_1,\dots,R_l\}$ satisfying conditions. 1. to 4. of theorem 3.1.

a) We take any set $\{Q_2,\dots,Q_m\}$ holding $\beta\{Q_2,\dots,Q_m\} = 0$ and Q_1 such that $\overline{Q_1} = \emptyset\overline{P_i}$.

This gives 4.

b) We take $\{P_1,\dots,P_n\}$ such that $\{P_i,P_1,\dots,P_k\} \subseteq \{P_1,\dots,P_n\}$ and $\beta\{P_1,\dots,P_n\} = 0$ holds. Clearly

$$\pi_x\overline{P_i} = \bigoplus_{j=1}^{n} a'_{ij}\overline{P_j}\ ,\ \text{where } a'_{ij} = a_{ij} \text{ if } 1\le j\le k \text{ and}$$

$a_{ij} = 0$ else, and hence 2. and 3. are satisfied.

c) We take any set $\{R_1,\dots,R_l\}$ such that $\beta(R_1,\dots,R_l\} = 0$

This finally gives 1.

Hence theorem 3.2 follows from theorem 3.1

Theorem 3.1 and theorem 3.2 are the starting points for the next section.

will give a shorter formulation of a necessary and sufficient condition for the linearity of automata.

<u>Def. 4.1</u>: $P_Y^\infty := \{R \in P_Y \mid \overline{\tau_x R} = \overline{\tau_{x'} R}$, for all $x, x' \in X$ of equal length$\}$

We know that π_x and τ_x are linear mappings for all $x \in X$. This is also true for all $x \in X$, since π_x and τ_x then compositions of linear mappings. Hence the following lemma holds.

<u>Lemma 4.1</u>: P_Y^∞ is a subspace of P_Y.

P_Y^∞ is now used to formulate a short necessary and sufficient condition for the linearity of automata.

<u>Theorem 4.2</u>: The reduced automaton $A = (X,S,Y,\delta,\lambda)$ is linearly realizable over K iff $\beta P_Y^\infty = 0$.

<u>Proof:</u> Let $A = (X,S,Y,\delta,\lambda)$ be linearly realizable. In this case conditions 1., 2. and 3. of theorem 4.1 hold. We want to prove $\langle R_1,\dots,R_l\rangle \subseteq P_Y^\infty$ from which we are able to conclude $\beta P_Y^\infty = 0$ since $\beta\{R_1,\dots,R_l\} = 0$.
We know $\overline{\pi_x R_j} = \overline{\pi_{x'} R_j}$ for all $j = 1,\dots,l$ and $x, x' \in X$. Assume $\overline{\pi_x R_j} = \overline{\pi_{x'} R_j}$ for all $j = 1,\dots,l$ and all $x, x' \in X$ of length k and let x_1 and x_2 be arbitrary elements of X. We compute $\pi_{xx_1} R_j = \pi_{x_1}(\pi_x P_j)$ and $\pi_{x'x_2} R_j = \pi_{x_2}(\pi_{x'} R_j)$. Since

<u>Algorithm.</u>

1. Is $\{R_1,\dots,R_l\} \subseteq P_Y^1$?

   No : go to 13

2. Set $\tilde{L}_1 := \{\tau_x R_j \mid j = 1,\dots,l,$ some fixed $x\varepsilon X\}$
3. Select L_1 as a basis of $\langle\tilde{L}_1\rangle$
4. $i := 1$
5. Is $L_i \subseteq P_S^1$?

   No : go to 13

6. Set $\tilde{L}_{i+1} := \{\pi_x P \mid P\varepsilon L_i,$ some fixed $x\varepsilon X\}$
7. Select L_{i+1} as a basis of

   $\langle\langle L_1\cup\dots\cup L_i\cup\tilde{L}_{i+1}\rangle \setminus \langle L_1\cup\dots\cup L_i\rangle\rangle$

8. Is $L_{i+1} = \emptyset$?

   No : Set $i := i+1$, go to 5

9. $k := i$
10. Set $M := \{\emptyset_U P \mid P\varepsilon\{L_1\cup\dots\cup L_k\}\} \cup \{\Psi_V R_j \mid j = 1,\dots,l\}$
11. Select N as a basis of M.
12. $\{N\},\{L_1\cup\dots\cup L_k\},\{R_1,\dots,R_l\}$ induces a linear realization.
13. There is no linear realization.

If the outputcode is not given and Y is a small set, one may <u>try</u> all minimal sets $\{R_1,\dots,R_l\}$ holding $\beta\{R_1,\dots,R_l\} = 0$.
For bigger sets Y we have to look for better methods. Analyzing theorem 4.1, one realizes that there is no real condition on $\langle Q_1,\dots,Q_m\rangle$. This code could be <u>constructed</u>, if $\{P_1,\dots,P_n\}$ and $\{R_1,\dots,R_l\}$ are given. In addition, the iteration of τ_x resp. π_x, i.e. using τ_x and π_x for $x\varepsilon X$ instead of those for $x\varepsilon X$ only,

$(\Psi_V R_j)(x) = (\Psi_V R_j)(x')$ for some V and $j = 1,\dots,l$ and $(\emptyset_U P_i)(x) = (\emptyset_U P_i)(x')$ for some U and $i = 1,\dots,n$. Using the definition of Ψ_V and $\emptyset_U$ we obtain $\tau_x R_j = \tau_{x'} R_j$ and $\pi_x P_i = \pi_{x'} P_i$ for $j = 1,\dots,l$ and $i = 1,\dots,n$, which finally implies $R_j(\lambda(s,x)) = R_j(\lambda(s,x'))$ and $P_i(\delta(s,x)) = P_i(\delta(s,x'))$ for all s and i,j as above. From $\beta\{R_1,\dots,R_l\} = 0$ and $\beta\{P_1,\dots,P_n\} = 0$ we conclude $\lambda(s,x) = \lambda(s,x')$ and $\delta(s,x) = \delta(s,x')$ for all $s\epsilon S$ which contradicts $x \neq x'$ since the automaton was reduced. Hence $\beta\{Q_1,\dots,Q_m\} = 0$.

This proof easily leads to an algorithm for linear realization if P_Y^∞ is given.

We actually proved a little more than stated in theorem 4.2, namely.

<u>Corollary 4.1:</u> Let $A = (X,S,Y,\delta,\lambda)$ be a reduced automaton. $\{R_1,\dots,R_l\}$ may be used for a linear realization of A iff $\beta\{R_1,\dots,R_l\} = 0$ and $\{R_1,\dots,R_l\} \subseteq P_Y^\infty$.

We want to give two more formulations of our result which may be useful in applications.

<u>Corollary 4.2:</u> (shift - condition) :
Let $A = (S,X,Y,\delta,\lambda)$ be a reduced automaton.
A is linearly realizable iff for all $y,y'\epsilon Y$,

$\overline{\pi_x R_j} = \overline{\pi_{x'} R_j}$ by asumption and $\overline{\pi_x R_j} \in \langle \bar{P}_1, \ldots, \bar{P}_n \rangle$ by theorem 4.1, we conclude

$$\overline{\pi_{xx_1} R_j} = \overline{\pi_{x_1}(\pi_x R_j)} = \overline{\pi_{x_2}(\pi_{x'} R_j)} = \overline{\pi_{x'x_2} R_j}.$$

Thus we proved by induction that $\overline{\pi_x R_j} = \overline{\pi_{x'} R_j}$ for all $j = 1, \ldots, l$ and $x, x' \in X$ of equal length and hence $\langle R_1, \ldots, R_l \rangle \subseteq P_Y^\infty$.

To show the converse, we have to find $\{R_1, \ldots, R_l\}$, $\{P_1, \ldots, P_n\}$ and $\{Q_1, \ldots, Q_m\}$ such that conditions 1., 2. and 3. of theorem 4.1 hold.

a) We take any subset $\{R_1, \ldots, R_l\}$ of P_Y^∞ such that $\beta\{R_1, \ldots, R_l\} = 0$. Next we choose $\{P_1, \ldots, P_n\}$ such that $\langle P_1, \ldots, P_n \rangle = \langle \{\tau_x R_j \mid x \in X \ , \ j = 1, \ldots, l\} \rangle$ and $\{Q_1, \ldots, Q_m\}$ such that
$\langle \bar{Q}_1, \ldots, \bar{Q}_m \rangle = \langle \forall \bar{R}_1, \ldots, \forall \bar{R}_l \ , \ \emptyset \bar{P}_1, \ldots, \emptyset \bar{P}_n \rangle$.

b) Conditions 2. and 3. of theorem 4.1 obviously hold.

c) We have to prove $\beta\{P_1, \ldots, P_n\} = 0$ and $\beta\{Q_1, \ldots, Q_m\} = 0$. Asume $\beta\{P_1, \ldots, P_n\} \neq 0$ and hence $\beta\{\tau_x R_j \mid x \in X \ , \ j = 1, \ldots, l\} \neq 0$. This implies the existence of $s, s' \in S$, such that $s \neq s'$ and $s \equiv s'(\alpha(\tau_x R_j))$ and hence $R_j(\lambda(s,x)) = R_j(\lambda(s',x))$ for all $x \in X$ and $j = 1, \ldots, l$. We conclude $\lambda(s,x) = \lambda(s',x)$ since $\beta\{R_1, \ldots, R_l\} = 0$. This means the equivalence of s and s' which is a contradiction to the minimality of the given automaton. Hence $\beta\{P_1, \ldots, P_n\} = 0$. Now assume $\beta\{Q_1, \ldots, Q_m\} \neq 0$. This implies the existence of $x, x' \in X$ such that $x \neq x'$ and

<u>5. Computation of $\mathbb{P}_Y^\infty$.</u>

According to the discussion in the last section the computation of $\mathbb{P}_Y^\infty$ is all that is left. The definition of $\mathbb{P}_Y^\infty$ as it stands does not help too much, since we would have to check some property for all natural numbers. The following sets will give more insight

<u>Def.5.1</u>: a) $\mathbb{P}_Y^n := \{R \in \mathbb{P}_Y \mid \overline{\tau_x R} = \overline{\tau_{x'} R}$ for all $x, x' \in X^+$ of equal length up to $n\}$.

b) $\mathbb{P}_S^n := \{P \in \mathbb{P}_S \mid \overline{\pi_x P} = \overline{\pi_{x'} P}$ for all $x, x' \in X^+$ of equal length up to $n\}$.

c) $\mathbb{P}_S^\infty := \{P \in \mathbb{P}_S \mid \overline{\pi_x P} = \overline{\pi_{x'} P}$ for all $x, x' \in X^+$ of equal length$\}$.

As trivial consequences of these definitions we note:

<u>Lemma 5.1</u>: a) $\mathbb{P}_S^n \supseteq \mathbb{P}_S^{n+1}$, b) $\mathbb{P}_Y^n \supseteq \mathbb{P}_Y^{n+1}$, c) $\mathbb{P}_S^n \supseteq \tau_x \mathbb{P}_Y^{n+1}$,

d) $\mathbb{P}_S^n \supseteq \pi_x P_S^{n+1}$, e) $\mathbb{P}_Y^\infty = \bigcap_{i=1}^{\infty} \mathbb{P}_Y^i$, f) $\mathbb{P}_S^\infty = \bigcap_{i=1}^{\infty} \mathbb{P}_S^i$.

In fact we have two descending chains of subspaces of P_Y and P_S respectively.

$$\mathbb{P}_Y \supseteq \mathbb{P}_Y^1 \supseteq \mathbb{P}_Y^2 \supseteq \dots \supseteq \mathbb{P}_Y^n \supseteq \dots \supseteq \mathbb{P}_Y^\infty$$

$$\mathbb{P}_S \supseteq \mathbb{P}_S^1 \supseteq \mathbb{P}_S \supseteq \dots \supseteq \mathbb{P}_S^n \supseteq \dots \supseteq \mathbb{P}_S^\infty$$

The next two results show that we are able to compute $\mathbb{P}_Y^\infty$ as well as $\mathbb{P}_S^\infty$ in a finite number of steps and can easily decide whether we reached the result.

$y \neq y'$, there is an $R \in P_Y$ such that $R(y) \neq R(y')$ and $\overline{\tau_x R} = \overline{\tau_{x'} R}$ for all $x, x' \in X$ of equal length.

<u>Corollary 4.3</u>: Let $A = (X,S,Y,\delta,\lambda)$ be a reduced automaton. The code $i_Y : Y \longrightarrow K^n$ may be used for a linear realization of A iff $i_Y(\lambda(s,x)) = i_Y(\lambda(s,x')) + c(x,x')$ for all $s \in S$ and $x, x' \in X$ of equal length, where $c(x,x')$ is a vector depending on x and x' but not on s.

Another remark is the following. As could be seen from the proof of theorem 4.2, two states s and s' are equivalent iff $s \equiv s'(\beta\{\tau_x R_j \mid x \in X\ ,\ j = 1,\ldots,l\})$. Hence if we compute P_Y^∞ of a nonreduced automaton and then apply the indicated algorithm for linear realization, we obtain a realization of the reduced form of the given automaton.
An algorithm for computing P_Y^∞ will be derived in the next section.

For actual computation of $\mathbb{P}_S^n$ and $\mathbb{P}_Y^n$ we may use the following lemma which follows from lemma 5.2 by induction (see [26]).

<u>Lemma 5.3</u>: a) $P\varepsilon\ \mathbb{P}_S^{n+1}$ iff $\pi_x^m P\varepsilon\ \mathbb{P}_S^1$ for some arbitrary $x\varepsilon X$ and all $0 \leq m \leq n$.

b) $R\varepsilon\ \mathbb{P}_Y^{n+1}$ iff $R\varepsilon\ \mathbb{P}_Y^1$ and $\pi_x^m \tau_x R\varepsilon\ \mathbb{P}_S^1$ for some arbitrary $x\varepsilon X$ and all $0 \leq m \leq n-1$.

Since the predecessor operators are linear, we obtain immediately from lemma 5.2 and theorem 5.2 the following characterization of $\mathbb{P}_Y^\infty$ and $\mathbb{P}_S^\infty$

<u>Lemma 5.4</u>: a) $P\varepsilon\ \mathbb{P}_S$ iff

$(\pi_x - \pi_{x'})\pi_{x_0}^n P = k_n$ for all $x,\ x'\varepsilon X$, some k_n some arbitrary $x_0\varepsilon X$ and $0 \leq n \leq |S|-1$.

b) $R\varepsilon\ \mathbb{P}_Y^\infty$ iff

$(\tau_x - \tau_{x'})R = k_0$ and $(\pi_x - \pi_{x'})\pi_{x_0}^n \tau_{x_0} R = k_n$ for all $x,\ x'\varepsilon X$, some $k_0, k_n\varepsilon K$, some $x_0\varepsilon X$ and $0 \leq n \leq |S| - 2$.

The advantage of this formulation lies in the fact, that iteration of the predecessor operators need only be carried out for a <u>fixed</u> but arbitrary $x_0\varepsilon X$ and not for all $x\varepsilon X^+$. If we want to get rid of the constants from K used above, we simply go to the factor spaces and obtain:

<u>Lemma 5.5</u>: a) $P\varepsilon\ \mathbb{P}_S^\infty$ iff

$(\overline{\pi}_x - \overline{\pi}_{x'})\overline{\pi}_{x_0}^n \overline{P} = 0$ for all $x,\ x'\varepsilon X$, some arbitrary $x_0\varepsilon X$ and $0 \leq n \leq |S|-1$.

Lemma 5.2: a) $P\varepsilon\, \mathbb{P}_S^{n+1}$ iff $P\varepsilon\, \mathbb{P}_S^n$ and $\pi_x P\varepsilon\, \mathbb{P}_S^n$ for some $x\varepsilon X$ (and hence all $x\varepsilon X$), $n = 1,2,\ldots$.

b) $R\varepsilon\, \mathbb{P}_Y^{n+1}$ iff $R\varepsilon\, \mathbb{P}_Y^n$ and $\tau_x R\varepsilon\, \mathbb{P}_S^n$ for some $x\varepsilon X$ (and hence all $x\varepsilon X$), $n = 1,2,\ldots$.

Proof: by straightforward induction on n (see [26]).

Theorem 5.1: If $\mathbb{P}_S^n = \mathbb{P}_S^{n+1}$, then $\mathbb{P}_S^n = \mathbb{P}_S^\infty$ and $\mathbb{P}_Y^{n+1} = \mathbb{P}_Y^\infty$

Proof: Using an induction argument it is sufficient to show $\mathbb{P}_S^{n+2} \supseteq \mathbb{P}_S^n$ and $\mathbb{P}_Y^{n+2} \supseteq \mathbb{P}_Y^{n+1}$.

If $P\varepsilon\, \mathbb{P}_S^n = \mathbb{P}_S^{n+1}$, then by lemma 5.2 we have $\Pi_x P\varepsilon\, \mathbb{P}_S^n$ also. Using lemma 5.2 again we obtain $P\varepsilon\, \mathbb{P}_S^{n+2}$.

If $R\varepsilon\, \mathbb{P}_Y^{n+1}$, then from lemma 5.2 we have $R\varepsilon\, \mathbb{P}_Y^n$ and $\tau_x R\varepsilon\, \mathbb{P}_S^n = \mathbb{P}_S^{n+1}$. Again by lemma 5.2 this results in $R\varepsilon\, \mathbb{P}_Y^{n+2}$.

Since S is a finite set, $\mathbb{P}_S^n = \mathbb{P}_S^{n+1}$ happens for some n. It is interesting and important to note that there is a bound on n, which is independent of the field K.

Theorem 5.2: If $k = |S|$, then $\mathbb{P}_S^{k-1} = \mathbb{P}_S^\infty$ and $\mathbb{P}_Y^k = \mathbb{P}_Y^\infty$.

Proof: Assume $\mathbb{P}_S^{k-1} \neq \mathbb{P}_S^k$. Then $\mathbb{P}_S^n \neq \mathbb{P}_S^{n+1}$ for all $n = 1,2,\ldots,$ $k-1$ and $\mathbb{P}_S \neq \mathbb{P}_S^1$. But the dimension of $\mathbb{P}_S$ equals k and hence the dimension of $\mathbb{P}_S^k$ has to be 0. This is a contradiction since K is contained in $\mathbb{P}_S^\infty$.

matrices.

<u>Lemma 5.6</u>: a) $\Pi_x P_s(b) = \begin{cases} 1 \text{ , if } \delta(b,x) = s \\ 0 \text{ , otherwise} \end{cases}$

for all $x \varepsilon X^+$ and $s,b \varepsilon S$.

b) $\tau_x R_y(b) = \begin{cases} 1 \text{ , if } \lambda(b,x) = y \\ 0 \text{ , otherwise} \end{cases}$

for all $x \varepsilon X^+$, $y \varepsilon Y$ and $b \varepsilon S$.

Proof: $\Pi_x P_s(b) = (P_s \circ \delta_x)(b) = P_s(\delta(b,x))$. By definition of P_s (see Section 2) the latter equals 1 if $s = \delta(b,x)$ and 0 otherwise. Part b) is proven the same way.

This result means that in representation of $\tau_x R_y$ and $\pi_x P_s$ as linear combinations of the basis vectors the only occurring coefficients are 0 and 1. Hence:

<u>Lemma 5.7</u>: a) $\pi_x P_s = \bigoplus_{b \varepsilon \hat{S}} P_b$, where $\hat{S} = \{b | \delta(b,x) = s\}$,
for all $x \varepsilon X^+$ and $s \varepsilon S$.

b) $\tau_x R_y = \bigoplus_{b \varepsilon \hat{S}} P_b$, where $\hat{S} = \{b | \lambda(b,x) = y\}$,
for all $x \varepsilon X^+$ and $y \varepsilon Y$.

It is important to remark that the <u>formal</u> representation of π_x and τ_x are independent of the specific field K, since 0 and 1 are elements of every field. Thus the set of formal equations in lemma 5.4 is independent of K and hence an <u>invariant of the automaton</u>! (1)

---

(1) This observation was first made by Eichner in [5].

b) $R\varepsilon\, \mathbb{P}_Y^{\infty}$ iff

$(\bar{\tau}_x-\bar{\tau}_x)R = 0$ and $(\bar{\pi}_x-\bar{\pi}_x)\bar{\pi}^n_{x_0}\tau_{x_0}\bar{R} = 0$

for all x, x'εX some arbitrary $x_0 \varepsilon X$ and

$0 \leq n \leq |S|-2$.

To reach our final version of the characterization of $\mathbb{P}_Y^{\infty}$ and $\mathbb{P}_S^{\infty}$ let the predecessor operators be given by matrices, which we denote by π_x and τ_x again, and $X = \{x_1,\ldots,x_r\}$. Consider the following matrices:

$$M_S := \begin{pmatrix} \pi_{x_1} - \pi_{x_2} \\ \pi_{x_2} - \pi_{x_3} \\ \vdots \\ \pi_{x_{r-1}} - \pi_{x_r} \end{pmatrix} \qquad M_Y := \begin{pmatrix} \bar{\tau}_{x_1} - \bar{\tau}_{x_2} \\ \bar{\tau}_{x_2} - \bar{\tau}_{x_3} \\ \vdots \\ \bar{\tau}_{x_{r-1}} - \bar{\tau}_{x_r} \end{pmatrix}$$

<u>Theorem 5.3</u>: a) $P\varepsilon\, \mathbb{P}_S$ iff $M_S\bar{\pi}^n_{x_0}\ \bar{P} = 0$ for some arbitrary $x_0\varepsilon X$ and $0 \leq n \leq |S|-1$.

b) $R\varepsilon\, \mathbb{P}_Y^{\infty}$ iff $M_Y\ \bar{R} = 0$ and $M_S\ \bar{\pi}^n_{x_0}\ \bar{\tau}_{x_0}\ \bar{R} = 0$ for some arbitrary $x_0\varepsilon X$ and $0 \leq n \leq |S|-2$.

Proof: Immediate consequence of lemma 5.5.

Theorem 5.3 provides us with a simple algorithm, if $\bar{\pi}_x$ and $\bar{\tau}_x$ are given as matrices for all xεX. To achieve this we use the basis of $\mathbb{P}_S$ and $\mathbb{P}_Y$ as given in Section 2 and compute the predecessors of the basis vectors. These constitute the columns of the desired

6. Examples.

In this section we want to explain the previously introduced construction and algorithms by some examples.

A) First consider the following automaton.

$M = (X,S,Y,\delta,\lambda)$ where $X = \{1,2\}$, $S = \{a,b,c,d\}$, $Y = \{0,1\}$ and δ and λ given by the tables:

	1	2	1	2
a	b	c	1	1
b	d	a	0	0
c	c	b	1	1
d	a	d	0	0

Let us try to realize this automaton over GF(2). The indexed partitions we have to look for have two blocks and we write the block with index 0 in the first place.

We start with $R = (0\ ,\ 1)$ and have to compute the predecessors with respect to 1 and 2 .

$\tau_1 R = (bd,ac) = \tau_2 R$.

The predecessors are equal up to shift and we compute the predecessors of $(bd,ac) = P_1$.

$\pi_1 P_1 = (ab,cd)$, $\pi_2 P_1 = (cd,ab)$.

There are again equal up to shift and we proceed with $(ab,cd) = P_2$.

$\pi_1 P_2 = (ad,bc)$, $\pi_2 P_2 = (bc,ad)$.

The computed predecessors are linear combinations of P_1 and P_2 , namely

$(bd,ac) \oplus (ab,cd) = (bc,ad) = \pi_2 P_2$.

Hence the automaton is linear and we have to compute the indexed partitions on X .

To use theorem 5.3 for computing $\mathbb{P}_S^{\infty}$ and $\mathbb{P}_Y^{\infty}$ we have to represent $\bar{\pi}_x$ and $\bar{\tau}_x$ by matrices. The following simple result will help to create $\bar{\pi}_x$ from π_x and $\bar{\tau}_x$ from τ_x.

<u>Lemma 5.8</u>: Let $T = \bigoplus_{m \in U} T_m$ for some $U \subseteq M$ and $m_0 \in M$. Then

a) $\bar{T} = \bigoplus_{m \in U} \bar{T}_m$, if $m_0 \notin U$.

b) $\bar{T} = \bigoplus_{m \in \bar{U}} (-\bar{T}_m)$, if $m_0 \in U$, where $\bar{U}$ is the complement of U in M.

Proof: Part a) is obvious. To prove part b), consider the following.

$$\bar{T} = (\bigoplus_{m \in U-\{m_0\}} \bar{T}_m) \oplus \bar{T}_{m_0} = (\bigoplus_{m \in U-\{m_0\}} \bar{T}_m) \oplus \overline{(1 \oplus \bigoplus_{m \neq m_0} (-\bar{T}_m))} =$$

$$= (\bigoplus_{m \in U-\{m_0\}} \bar{T}_m) \oplus \bigoplus_{m \neq m_0} (-\bar{T}_m) = \bigoplus_{m \in \bar{U}} (-\bar{T}_m).$$

We used $T_{m_0} = 1 \oplus \bigoplus_{m \neq m_0} (-T_m)$ which has to be proven. For any b M we obtain: $(1 \oplus \bigoplus_{m \neq m_0} (-T_m))(b) = 1 - \sum_{m \neq m_0} T_m(b)$.

But $T_m(b) = 1$ iff $m = b$. Hence $(1 \oplus \bigoplus_{m \neq m_0} (-T_m))(b) = 1$ iff $b = m_0$ i.e. iff $T_{m_0}(b) = 1$.

This makes the representation of $\bar{\pi}_x$ and $\bar{\tau}_x$ independent of the specific field K and hence the set of equations in theorem 5.3 is an invariant of the automaton also.

Some examples of actual computations will be given in the next section.

where a "1" is in position (row s_1 , column s_2) if $\delta(s_1, x) = s_2$.

To go from π_x to $\bar{\pi}_x$, whe have to delete one row and column. Let us take row and colum d .

$$\bar{\pi}_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} , \qquad \bar{\pi}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

The first colum of π_1 had to be changed, since the entry in the deleted row was 1 .

Next we have to compute $\bar{\pi}_1 - \bar{\pi}_2$ etc.

$$\bar{\pi}_1 - \bar{\pi}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} .$$

$$(\bar{\pi}_1 - \bar{\pi}_2)\bar{\pi}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} .$$

No new rows appear. Hence we know $P_S^1 = P_S^2 = P_S^\infty$.

From $\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = 0$ we conclude

$$F_S^\infty = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} , \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} , \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} , \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$$ and hence

$$P_S^\infty = \{(ab,cd,\phi),(\phi,abcd),(ab,cd),(cd,ab),(ac,bd),(bd,ac),(ad,bc),(bc,ad)\} .$$

We know that every code given by a subset of P_S^∞ results in a linear realization.

If we set $U_1 = (ab,cd)$, then $\Phi_{U_1} P_1 = (1,2)$. Similarly if

$U_2 = (ad,bc)$, then $\Phi_{U_2} P_2 = (1,2)$.

Hence we obtain the following codings:

$$S: \begin{array}{l} \\ a \longrightarrow \\ b \longrightarrow \\ c \longrightarrow \\ d \longrightarrow \end{array} \begin{array}{cc} s_1 & s_2 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{array} \qquad X: \begin{array}{l} \quad\quad x \\ 1 \longrightarrow 0 \\ 2 \longrightarrow 1 \end{array} \qquad Y: \begin{array}{l} \quad\quad y \\ 0 \longrightarrow 0 \\ 1 \longrightarrow 1 \end{array}$$

This results in the following realization:

$$s_1' = s_2 \oplus x$$
$$s_2' = s_2 \oplus s_2 \oplus x \oplus 1$$
$$y = s_1$$

In order to get a "real" linear realization, we have to add one more internal variable, such that the code of S becomes

$$a \longrightarrow 1\ 0\ 1$$
$$b \longrightarrow 0\ 0\ 1$$
$$c \longrightarrow 1\ 1\ 1$$
$$d \longrightarrow 0\ 1\ 1$$

and $s_1' = s_2 \oplus x$

$$s_2' = s_1 \oplus s_2 \oplus s_3 \oplus x$$
$$s_3' = s_3$$

B) Now let us use the methods of Section 5 to compute $\wp_S^\infty$ for the automaton in A). We have to write π_1 and π_2 as matrices. Simply:

π_1 :

	a	b	c	d
a	0	1	0	0
b	0	0	0	1
c	0	0	1	0
d	1	0	0	0

π_2 :

	a	b	c	d
a	0	0	1	0
b	1	0	0	0
c	0	1	0	0
d	0	0	0	1

$$\begin{pmatrix}1&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&1&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&0&1&0\\0&0&0&1&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&1\end{pmatrix}\begin{array}{c}P_1P_2P_3\\\begin{pmatrix}0&0&0\\1&1&0\\0&1&1\\1&0&1\\1&0&0\\0&1&0\\1&1&1\\0&0&1\end{pmatrix}\end{array}=\begin{array}{c}\begin{array}{ccc}P_1+P_2&P_1&P_1+P_2+P_3\end{array}\\\begin{pmatrix}0&0&0\\0&1&0\\1&0&0\\1&1&0\\1&1&1\\1&0&1\\0&1&1\\0&0&1\end{pmatrix}\end{array}$$

Hence new coordinates are not needed. The last thing to be determined is the influence of the input. For this reason we have to multiply by τ_1 and π_1 .

$$\begin{pmatrix}0&0&0&0&0&1&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&0&0&1&0\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&1\\0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0\end{pmatrix}\begin{array}{c}R_1R_2R_3\\\begin{pmatrix}0&0&0\\0&0&1\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&0\\1&1&1\end{pmatrix}\end{array}=\begin{array}{c}\bar{P}_1P_2\bar{P}_3\\\begin{pmatrix}1&0&1\\0&1&1\\1&1&0\\0&0&0\\0&0&1\\1&1&1\\0&1&0\\0&0&0\end{pmatrix}\end{array}$$

$$\begin{pmatrix}0&0&0&1&0&0&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\0&0&1&0&0&0&0&0\\0&0&0&0&0&1&0&0\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0\end{pmatrix}\begin{array}{c}P_1P_2P_3\\\begin{pmatrix}0&0&0\\1&1&0\\0&1&1\\1&0&1\\1&0&0\\0&1&0\\1&1&1\\0&0&1\end{pmatrix}\end{array}=\begin{array}{c}\begin{array}{ccc}\overline{P_1+P_2}&P_1&\overline{P_1+P_2+P_3}\end{array}\\\begin{pmatrix}1&0&1\\1&1&1\\0&0&1\\0&1&1\\0&1&0\\0&0&0\\1&1&0\\1&0&0\end{pmatrix}\end{array}$$

Finally the following linear realization is obtained :

$y_1 = s_1 + x$ $\qquad$ $s_1' = s_1 + s_2 + x$

$y_2 = s_2$ $\qquad$ $s_2' = s_1$

$y_3 = s_3 + x$ $\qquad$ $s_3' = s_1 + s_2 + s_3 + x$.

C) Our next example is taken from Eichner [5].

$M = (X,S,Y,\delta,\lambda)$, where $X = \{1,2\}$, $S = \{1,2,3\}$, $Y = \{1,2,3\}$ and δ and λ are given by :

	1 2	1 2
1	1 2	1 3
2	1 2	2 1
3	3 2	1 3

We want to compute P_Y^∞ .

$$\tau_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} , \quad \tau_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\pi_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} , \quad \pi_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

We delete the last rows and colums .

$$\bar{\tau}_1 = \begin{pmatrix} 0 & 0 \\ -1 & 1 \end{pmatrix} , \quad \bar{\tau}_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} , \quad \bar{\pi}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} , \quad \bar{\pi}_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Now :

$$\bar{\tau}_1 - \bar{\tau}_2 = \begin{pmatrix} 0 & 0 \\ -2 & 1 \end{pmatrix} , \quad \bar{\pi}_1 - \bar{\pi}_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} .$$

Since $\bar{\pi}_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, we can conclude $P_S^1 = P_S^2 = P_S^\infty$.

Hence $\begin{pmatrix} A \\ B \end{pmatrix} \in F_Y^\infty$ iff

$$\begin{pmatrix} 0 & 0 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = 0 , \quad \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = 0 .$$

From this we get $\begin{pmatrix} A \\ B \end{pmatrix} \in F_Y^\infty$ iff $B = 2A$.

This statement is independent of the field under consideration!

If we want GF(2) , then $B = 0$ and there do not exist enough solutions to obtain a code. For any other finite prime field, P_Y^∞ is big enough.

Take GF(3) for example. Then

$$F_Y^\infty = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} , \begin{pmatrix} 1 \\ 2 \end{pmatrix} , \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right\} \quad \text{which means}$$

$$P_Y^\infty = ((123,\phi,\phi) , (\phi,123,\phi) , (\phi,\phi,123) , (3,1,2) , (1,2,3) , (2,3,1) , (3,2,1) , (2,1,3) , (1,3,2)) .$$

D) Let us now consider the following example.

	0	1	0	1
a	a	d	A	F
b	f	g	G	D
c	e	h	D	G
d	b	c	F	A
e	g	f	E	B
f	d	a	C	H
g	c	b	H	C
h	h	e	B	E

We desire a realization over GF(2) ,

$$\tau_0 = \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&1&0&0&0&0&0&0 \end{pmatrix} \qquad \tau_1 = \begin{pmatrix} 0&0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \end{pmatrix}$$

$$\pi_0 = \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \end{pmatrix} \qquad \pi_1 = \begin{pmatrix} 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \end{pmatrix}$$

We delete the first rows and colums to obtain :

This gives the following equations :

$$\bar{R}(D) + \bar{R}(F) + \bar{R}(G) = 0 \qquad \bar{R}(C) + \bar{R}(E) + \bar{R}(G) = 0$$
$$\bar{R}(B) + \bar{R}(E) + \bar{R}(F) = 0 \qquad \bar{R}(B) + \bar{R}(C) + \bar{R}(D) = 0$$
$$\bar{R}(C) + \bar{R}(F) + \bar{R}(H) = 0 \qquad \bar{R}(B) + \bar{R}(G) + \bar{R}(H) = 0$$

It is easy to see that there are enough solutions. For instance

$\bar{R}_1 = (0\ 0\ 0\ 1\ 1\ 1\ 1)$, $\bar{R}_2 = (0\ 1\ 1\ 0\ 0\ 1\ 1)$, $\bar{R}_3 = (1\ 0\ 1\ 0\ 1\ 0\ 1)$

is a set of solutions which gives the following code of Y :

$$\begin{array}{lccc} & R_1 & R_2 & R_3 \\ A \longrightarrow & 0 & 0 & 0 \\ B \longrightarrow & 0 & 0 & 1 \\ C \longrightarrow & 0 & 1 & 0 \\ D \longrightarrow & 0 & 1 & 1 \\ E \longrightarrow & 1 & 0 & 0 \\ F \longrightarrow & 1 & 0 & 1 \\ G \longrightarrow & 1 & 1 & 0 \\ H \longrightarrow & 1 & 1 & 1 \end{array}$$

In order to obtain a code of S , we have to multiply by τ_0 .

$$\begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&1&0&0&0&0&0&0 \end{pmatrix} \overset{R_1R_2R_3}{\begin{pmatrix} 0&0&0 \\ 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{pmatrix}} = \overset{P_1P_2P_3}{\begin{pmatrix} 0&0&0 \\ 1&1&0 \\ 0&1&1 \\ 1&0&1 \\ 1&0&0 \\ 0&1&0 \\ 1&1&1 \\ 0&0&1 \end{pmatrix}}$$

P_1, P_2, P_3 already constitutes a code of S . We have to multiply by π_0 to check whether we need more coordinates for linear realization.

Concluding Remarks.

We settled the long standing problem of realization of finite automata over a given arbitrary finite field. Section 1 provided the statement of the problem and Section 2 introduced indexed partitions which are used to formulate results in the remaining sections. The results on single coordinates in Section 3 were used for the main result in Section 4, where a necessary and sufficient condition for linearity is derived in terms of indexed partitions and predecessor operations. The crucial set P_Y is shown to be the set of solutions of a system of linear equations, which can be derived without explicit knowledge of the specific field K.

Careful examination of the concepts and proofs in this paper shows that all what was used from K is the existence and properties of 0 and 1 as well as the properties of "+" and the existence of "•". Thus the whole theory goes through if we use an abelian group A and a set B of mappings of A into A containing the identity instead of the field K, for instance a ring with unit. This remark agrees with results in [25], where certain properties of linear automata are generalized to this case.

The methods used in this paper may also be used to solve the problem of realizations with linear parts. This will be shown in a forthcoming paper.

$$\bar{\tau}_0 = \begin{pmatrix} 0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0 \end{pmatrix} \qquad \bar{\tau}_1 = \begin{pmatrix} 0&0&1&0&1&0&0 \\ 0&0&0&0&1&1&0 \\ 0&0&0&0&1&0&0 \\ 1&0&0&0&1&0&0 \\ 0&0&0&0&1&0&1 \\ 0&1&0&0&1&0&0 \\ 0&0&0&1&1&0&0 \end{pmatrix}$$

$$\bar{\pi}_0 = \begin{pmatrix} 0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0 \\ 1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1 \end{pmatrix} \qquad \bar{\pi}_1 = \begin{pmatrix} 0&0&1&0&0&1&0 \\ 0&0&1&0&0&0&1 \\ 0&1&1&0&0&0&0 \\ 0&0&1&0&1&0&0 \\ 0&0&1&0&0&0&0 \\ 1&0&1&0&0&0&0 \\ 0&0&1&1&0&0&0 \end{pmatrix}$$

$$\bar{\tau}_0 - \bar{\tau}_1 = \begin{pmatrix} 0&0&1&0&1&1&0 \\ 0&0&1&0&1&1&0 \\ 0&0&0&0&0&0&0 \\ 1&0&0&1&1&0&0 \\ 0&1&0&0&1&0&1 \\ 0&1&0&0&1&0&1 \\ 1&0&0&1&1&0&0 \end{pmatrix}, \qquad \bar{\pi}_0 - \bar{\pi}_1 = \begin{pmatrix} 0&0&1&0&1&1&0 \\ 0&0&1&1&0&0&1 \\ 1&1&1&0&0&0&0 \\ 0&0&1&0&1&1&0 \\ 0&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0 \\ 0&0&1&1&0&0&1 \end{pmatrix}$$

Only the linear independant rows of $\bar{\pi}_0 - \bar{\pi}_1$ are used for further computation. The same holds for the next steps.

$$\begin{pmatrix} 0&0&1&0&1&1&0 \\ 0&0&1&1&0&0&1 \\ 1&1&1&0&0&0&0 \end{pmatrix} \bar{\pi}_0 = \begin{pmatrix} 1&1&1&0&0&0&0 \\ 1&0&0&0&0&1&1 \\ 1&0&0&1&1&0&0 \end{pmatrix}$$

$$\begin{pmatrix} 1&0&0&0&0&1&1 \\ 1&0&0&1&1&0&0 \end{pmatrix} \bar{\pi}_0 = \begin{pmatrix} 0&1&0&0&1&0&1 \\ 0&0&1&0&1&1&0 \end{pmatrix}$$

$$\begin{pmatrix} 0&1&0&0&1&0&1 \end{pmatrix} \bar{\pi}_0 = \begin{pmatrix} 0&0&1&1&0&0&1 \end{pmatrix}$$

No new conditions are added. Hence $\bar{R} = (\bar{R}(B),\dots,\bar{R}(H)) \in \mathbb{F}_Y^{\infty}$ iff

$$(\bar{\tau}_0 - \bar{\tau}_1)\,\bar{R} = 0 \quad \text{and} \quad \begin{pmatrix} 0&0&1&0&1&1&0 \\ 0&0&1&1&0&0&1 \\ 1&1&1&0&0&0&0 \\ 1&0&0&0&0&1&1 \\ 1&0&0&1&1&0&0 \\ 0&1&0&0&1&0&1 \end{pmatrix} \bar{\tau}_0\,\bar{R} = 0$$

[9] Gill, A.:
Synthesis of Linear Sequential Circuits from Input-Output-Relations.
SIAM Journal on Applied Mathematics, 16, pp 216-227, 1968

[10] Harrison, M.A.:
Lectures on Linear Sequential Machines.
Academic Press, 1969

[11] Hartmanis, J., Davis, W.A.:
Homomorphic Images of Linear Sequential Machines.
Journal of Computer and System Sciences 1, pp 155-165, 1965

[12] Hartmanis, J.:
Linear Multivalued Sequential Coding Networks.
IEEE Transactions on Circuit Theory, CT - 6, pp 69-74, 1959

[13] Hartmanis, J.:
Two Tests for the Linearity of Sequential Machines.
IEEE Transactions on Computers, EC - 14, pp 781-786, 1965

[14] Herman, G.T.:
When is a Sequential Machine the Realization of Another? Mathematical System Theory 5, pp 115-127, 1971

[15] Herman, G.T.:
Every Finite Sequential Machine is linearly Realizable.
Journal of Computer and System Sciences 5, pp 489-510

References

[1] Cohn, M., Even, S.:
Identification and Minimization of Linear Machines.
IEEE Transactions on Computers, EC - 14, pp 367-376, 1965

[2] Deussen, P.:
Lineare Darstellungen von Automaten
Habilitationsschrift, Munchen, 1969

[3] Davis, W.A.:
The Linearity of Sequential Machines: A Critical Review. Ninth Annual Symposium on Switching and Automata Theory, pp 427-430, 1968

[4] Davis, W.A., Brzozowski, J.A.:
On the Linearity of Sequential Machines.
IEEE Transactions on Computers, EC - 15, pp 21-29, 1966

[5] Eichner, L.:
Lineare Realisierbarkeit endlicher Automaten uber endlichen Korpern. Dissertation, Freiburg 1971

[6] Elspas, B.:
The Theory of Autonomonus Linear Sequential Networks.
IEEE Transactions on Circuit Theory, CT - 6, pp 45-60, 1959

[7] Friedland, B.:
Linear Modular Sequential Circuits
IEEE Transactions on Circuit Theory, CT - 6, pp 61-68, 1959

[8] Gill, A.:
Linear Sequential Machines.
McGraw-Hill, New York, 1966

[23] Reusch, B.:
On Linear Realization of Automata over Arbitrary Fields. GMD, ITAS - Seminarberichte, Nr. 23, 1970

[24] Reusch, B.:
Zur Realisierung von Automaten über Algebren. Berichte der Gesellschaft für Mathematik und Datenverarbeitung, Bd. 38, Bonn, 1971

[25] Reusch, B.:
Theorems on Additive Automata. in : Z. Kohavi, A. Paz (ed.) Theory of Machines and Computations, Academic Press, 1971

[26] Reusch, B.:
Linear Realization of Finite Automata. University of Iowa, Iowa City, Tech. Rep. 62, 1972

[27] Srinivasan, C.V.:
State Diagrams of Linear Sequential Machines. Journal of the Franklin Institute, 273, pp 383-418, 1962

[28] Stucky, W.:
Linear realisierbare endliche Automaten Dissertation, Saarbrücken, 1969

[29] Walter, H.:
Einige Bemerkungen zum Problem der linearen Realisierung von endlichen Automaten. Unpublished, Saarbrüken, 1969

[30] Wang, K.C.:
Synthesis of Linear Sequential Machines with Unspecified Outputs. IEEE Transactions on Computers, C - 18, pp 145-153, 1969

[16] Hotz, G.:
On the Mathematical Theory of Linear Sequential Networks. in : H. Aiken, W.F. Main (ed.) Switching Theory in Space Technology, pp 11-19 Stanford University Press, 1963

[17] Huffman, D.A.:
A Linear Circuit Viewpoint of Error - Correcting Codes. IEEE Transactions on Information Theory, IT - 8, pp 20-28, 1956

[18] Huffman, D.A.:
The Synthesis of Linear Sequential Coding Networks. in : C. Cherry (ed.) Information Theory, pp 77-95 Academic Press, New York, 1956.

[19] Kandzia, P.:
Das Zyklenverhalten linearer Gruppenautomaten. 3. Colloquium über Automatentheorie, Hannover, 1965

[20] Marino, P.J.:
Identification and Synthesis of Linear Sequential Machines. Bell System Technical Journal 47, pp 343-384, 1968

[21] Reusch, B.:
Über vollständige and teilweise lineare Realisierung von Automaten. Dissertation, Bonn, 1968 Reprinted as : Berichte der Gesellschaft für Mathematik und Datenverarbeitung, Bd. 17, Bonn, 1969

[22] Reusch, B.:
Lineare Automaten
BI Hochschulskripten 708, Bibliographisches Institut, 1969

[31] Yau, S.S., Wang, K.C.:
Linearity of Sequential Machines.
IEEE Transactions on Computers, EC - 15, pp 337-354, 1966