The full decomposition of sequential machines with the state and output behaviour realization

Full decomposition of sequential machines with the output behaviour realization

tu/e
Research Report
ISSN 0167-9708
Coden: TEUEDE

Eindhoven
University of Technology
Netherlands

Faculty of Electrical Engineering

The Full Decomposition of Sequential Machines with the State and Output Behaviour Realization

by
L. Jóźwiak

EUT Report 88-E-188
ISBN 90-6144-188-9

January 1988

Eindhoven University of Technology Research Reports
EINDHOVEN UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
Eindhoven The Netherlands

ISSN 0167- 9708 Coden: TEUEDE

THE FULL DECOMPOSITION OF SEQUENTIAL MACHINES WITH THE STATE AND OUTPUT BEHAVIOUR REALIZATION

by

L. Jóźwiak

EUT Report 88-E-188
ISBN 90-6144-188-9

Eindhoven
January 1988

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Jóźwiak, L.

The full decomposition of sequential machines with the state
and output behaviour realization / by L. Jóźwiak. - Eindhoven:
University of Technology, Faculty of Electrical Engineering. -
Fig. - (EUT report, ISSN 0167-9708; 88-E-188)
Met lit. opg., reg.
ISBN 90-6144-188-9
SISO 664 UDC 681.325.65:519.6 NUGI 832
Trefw.: automatentheorie.

THE FULL DECOMPOSITION OF SEQUENTIAL MACHINES WITH THE STATE AND OUTPUT BEHAVIOUR REALIZATION

Lech Jóźwiak
Group Digital Systems, Faculty of Electrical Engineering,
Eindhoven Universwity of Technology (The Netherlands)

Abstract-The design of large logic systems leads to the practical problem how to decompose a complex system into a number of simpler subsystems. The decomposition theory of sequential machines tries to find answers to this problem for sequential machines. For many years, the "simpler " machine was defined as a machine with fewer states and, therefore, state-decompositions of sequential machines were considered. Together with the progress in LSI technology and the introduction of array logic into the design of sequential circuits a real need arose for decompositions not only on states of sequential machines but on inputs and outputs too, i.e. for full-decompositions.

In this report, a general and unified classification of full-decompositions is presented, formal definitions of different sorts of full-decompositions for Mealy and Moore machines are introduced and theorems about the existence of full-decompositions with the state and output behaviour realization are formulated and proved. The presented theorems have a straightforward practical interpretation. Based on them, a set of algorithms has been developed and a system of programs has been made for computing the different sorts of decompositions.

Index Terms-Automata theory, decomposition, logic system design, sequential machines.

Acknowledgements-The author is greatly indebted to prof.ir.A.Heetman and prof.ir.M.P.J.Stevens for making it possible to perform this work.

CONTENTS

1. Introduction.

The design of large logic systems leads to the following practical problem:

How to decompose a complex system into a number of simpler subsystems in order to obtain:

- the clearer organization of the system and of the design, implementation and verification process,

- the possibility of optimization of the separate subsystems, whereas it can be impossible directly to optimize the whole system,

- the possibility of implementation of the system by existing building blocks.

The decomposition theory of sequential machines tries to find answers to the following question: *how to decompose a given sequential machine M into a number of "smaller" (and therefore easier to develop and implement) component sequential machines $M_1, M_2, \ldots, M_n$ which, in combination, realize the behaviour of a given machine M.*

Research in the above mentioned field was started in early sixties [8][9][10][19][20]. For many years, the "smaller" machine was defined as a machine with fewer states than the given machine; therefore state-decompositions of sequential machines were considered. Definitions of decompositions on states were introduced, constructive theorems about the existence of state decompositions were presented and some practical algorithms for state decompositions were developed [4][12][16][17][18][19] [20].

Together with the progress in LSI technology and the introduction of array logic (PAL, PGA, PLA, PLS) into the design of sequential circuits, a real need arose for decompositions not only on states of sequntial machines but on inputs and outputs too, i.e. for full-decompositions.

An approach to the full-decomposition of sequential machines has been presented in [14] and [15]. Among other things, the definitions and theorems concerning parallel and two types of serial full-decompositions for Mealy machines were introduced.

In this work a general and unified classification of full-decompositions will be presented, formal definitions of different sorts of full-decompositions for Mealy and Moore machines will be introduced and theorems about the existence of full-decompositions with the state and output behaviour realization will be formulated and proved leading immediately to some practical algorithms. The theorems concerning the types of full-decomposition defined in [14] were formulated and proved here with weaker assumptions than those in [14] and , therefore, they are more general. They include cases which are important from the practical point of view and were not covered by the theorems presented in [14]. The notions of output-dependent trinity, state dependent trinity semitrinity and induced semitrinity used in presented theorems have a straightforward practical interpretation which is an important advantage.

2.Algebraic models of sequential machines and a full-decomposition.

DEFINITION 2.1 A *sequential machine* M is an algebraic system defined as follows:

$$M = (I, S, O, \delta, \lambda) ,$$

where:

I - finite nonempty set of inputs,

S - finite nonempty set of internal states,

O - finite set of outputs,

δ - next state function, $\delta: SxI \longrightarrow S$,

λ - output function, $\lambda: SxI \longrightarrow O$ (a *Mealy machine*),

or $\lambda: S \longrightarrow O$ (a *Moore machine*).

If the output set O and the output function λ are not defined, the sequential machine $M = (I, S, \delta)$ is called a ***state machine.***

The machine functions δ and λ can be considered as sets of functions created for each input:

$\delta = \{\delta_x | \ \delta_x: S \longrightarrow S \text{ and } x \in I\}$

and

$\lambda = \{\lambda_x | \ \lambda_x: S \longrightarrow O \text{ and } x \in I\}$,

where $\delta_x: S \longrightarrow S$ and $\lambda_x: S \longrightarrow O$ are defined by:

$\forall x \in I \ \forall x \in S \quad \delta_x(s) = \delta(s,x),$

$\lambda_x(s) = \lambda(s,x).$

The δ_x and λ_x are called, respectively, the next-state function and the output function with respect to the input x.

In the next sections for $\delta_x(s)$ and $\lambda_x(s)$ we will use the notations $s\delta_x$ and $s\lambda_x$.

For $x \in I$ and $Q \subseteq S$, we will define the two partial functions:

$\overline{\delta}_x: 2^S \longrightarrow 2^S$ and $\overline{\lambda}_x: 2^S \longrightarrow 2^O$,

where:

$\forall x \in I \ \forall Q \subseteq S \quad Q\overline{\delta}_x = \{s\delta_x | \ s \in Q\}, \ Q\overline{\lambda}_x = \{s\lambda_x | \ s \in Q\}.$

For $X \subseteq I$ and $Q \subseteq S$, we will define also the following two partial functions:

$\overline{\delta}_X: 2^S \longrightarrow 2^S$ and $\overline{\lambda}_X: 2^S \longrightarrow 2^O$,

where:

$Q\overline{\delta}_X = \{s\overline{\delta}_x | \ s \in Q \wedge x \in X\},$

$Q\overline{\lambda}_X = \{s\overline{\lambda}_x | \ s \in Q \wedge x \in X\}.$

In this work, we take into account only simple decompositions (i.e. decompositions with two component machines) and, therefore, the term "decomposition" is used further in the meaning of "simple decomposition".

Let $M = \{I, S, O, \delta, \lambda\}$ be the machine we want to decompose and $M_1 = \{I_1, S_1, O_1, \delta_1, \lambda_1\}$ and $M_2 = \{I_2, S_2, O_2, \delta_2, \lambda_2\}$ are two partial machines.

In a full-decomposition, we are interested in finding such partial machines M_1 and M_2 that each of them has fewer states and/or outputs than machine M and/or each of them can calculate its next states and outputs using only the part of information about the input of machine M and, in combination, they form machine M' imitating M from the input-output point of view.

In a state-decomposition, we were interested in finding machines M_1 and M_2 with only fewer internal states. Inputs and outputs were not decomposed.

Before we consider different sorts of full-decomposition, we recall from [12] the definition of realization.

<u>DEFINITION</u> <u>2.2</u> Machine $M' = (I',S',O',\delta',\lambda')$ ***realizes*** *(is* ***realization*** *of)* machine $M = (I,S,O,\delta,\lambda)$ *if and only if* the following relations exist:

ψ: $I \longrightarrow I'$ (a function),

ϕ: $S \longrightarrow 2^{S'}$ (a function into nonvoid subsets of S'),

θ: $O' \longrightarrow O$ (a surjective partial function) ,

and this relations satisfy the following conditions:

$$\phi(s)\delta'_{\psi(x)} \subseteq \phi(s\delta_x)$$

and

$$s\lambda_x = \theta(s'\lambda'_{\psi(x)}) \qquad \text{(for a Mealy machine)}$$

or

$$s\lambda = \theta(s'\lambda') \qquad \text{(for a Moore machine)}$$

for all $s \in S$, $s' \in \phi(s)$ and $x \in I$.

Let I^* be a set of all input sequences $x_1x_2\ldots x_n$ ($n=0,1,\ldots$), let $\vec{x} = \vec{x}'x$ for $\vec{x}' \in I^*$ and $x \in I$ and let $\vec{\lambda}$ and $\vec{\delta}$ be two functions calculating the last output and the last state reached by a machine from the state s under the input sequence $\vec{x}$:

$\vec{\delta}$: $SxI^* \longrightarrow S$, $\vec{\delta}(s,\vec{x}) = \delta(\vec{\delta}(s,\vec{x}'),x)$,

$\vec{\lambda}$: $SxI^* \longrightarrow O$, $\vec{\lambda}(s,\vec{x}) = \lambda(\vec{\delta}(s,\vec{x}'),x)$ (Mealy case),

$\vec{\lambda}(s,\vec{x}) = \lambda(\vec{\delta}(s,x))$ (Moore case).

It can be proved that if M' is a realization of M in the sense of definition 2.1 then $\forall s \in S$ $\forall s' \in \phi(s)$ and $\forall \vec{x} \in I^*$: $\vec{\lambda}(s,\vec{x}) = \theta(\vec{\lambda}'(s',\psi(x)))$, i.e. for all possible input sequences outputs reached by machine M and its imitation M' are, after a renaming, identical. Because of this fact, the realization in the sense of definition 2.1 will be called by us the ***realization of the output behaviour***.

In some cases, we are concerned with not only the output changes of the machine but also with the state changes. Therefore, we will consider also realizations of the state behaviour of sequential machines.

<u>DEFINITION</u> <u>2.3</u> Machine $M' = (I', S', O', \delta', \lambda')$, ***realizes the state and output behaviour*** of machine $M = (I, S, O, \delta, \lambda)$ *if and only if* the following relations exist:

ψ: $I \longrightarrow I'$ (a function),

ϕ: $S' \longrightarrow S$ (a surjective partial function)

θ: $O' \longrightarrow O$ (a surjective partial function) such that:

$$\phi(s')\delta_x = \phi(s'\delta'_{\psi(x)})$$

and

$$\phi(s')\lambda_x = \theta(s'\lambda'_{\psi(x)}) \quad \text{(for a Mealy machine)}$$

or

$$\phi(s')\lambda = \theta(s'\lambda') \quad \text{(for a Moore machine).}$$

The realization of the state and output behaviour is a special case of the realization of the output behaviour. If function ϕ in definition 2.2 maps each state of M onto a single state of M' and ϕ is a one-to-one function then definition 2.2 is equivalent to definition 2.3.

In a ***full-decomposition***, we are interested in finding the partial machines M_1 and M_2 and the mappings:

$\psi: I \longrightarrow I_1 x I_2$,
$\phi: S \longrightarrow 2^{S_1 \times S_2}$, (the realization of the output behaviour)
$\theta: O_1 x O_2 \longrightarrow O$,
or
$\psi: I \longrightarrow I_1 \times I_2$, (the realization of the state)
$\phi: S_1 \times S_2 \longrightarrow S$, (and output behaviour)
$\theta: O_1 \times O_2 \longrightarrow O$,

that the machines M_1 and M_2 together with the mappings ψ, ϕ, θ realize the behaviour of a machine M.

We will say that a full-decomposition is ***nontrivial*** *if and only if*:

$|I_1|<|I| \wedge |I_2|<|I| \vee |S_1|<|S| \wedge |S_2|<|S| \vee |O_1|<|O| \wedge |O_2|<|O|$, where $|Z|$ – number of elements *in* the set Z.

In the case of a ***state-decomposition***, we are interested in finding machines M_1 and M_2 and, in fact, only one mapping $\phi: S_1 \times S_2 \longrightarrow S$.

It is evident that state-decomposition is a special case of full-decomposition.

3. Classification of full-decompositions.

Decompositions can be classified according to the kind of connections that exist between the component machines.

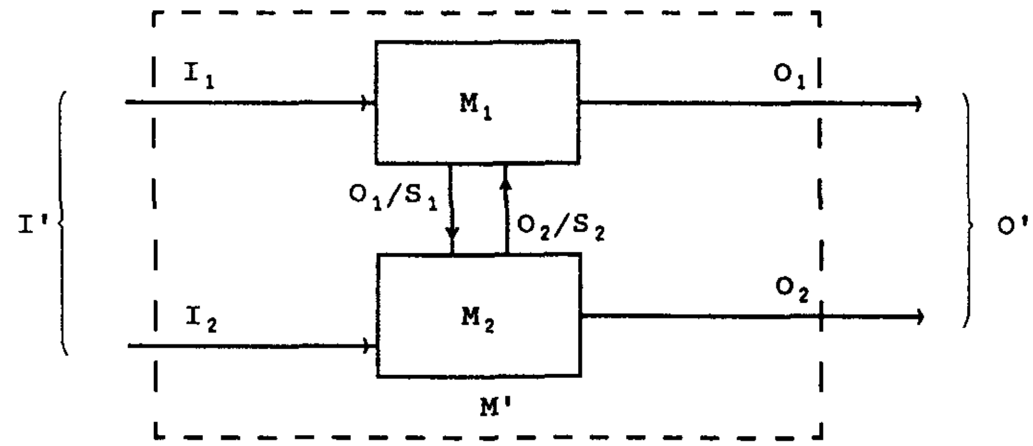

Fig 3.1 The information flow between the component machines in full-decomposition.

In general, each of the component machines can use the information about the state or output of the other component machine in order to compute its own next state and output (Fig.3.1).

From the point of view of the strength of the connections between the component machines we can distinguish the following sorts of full-decompositions:

(i) *parallel full-decomposition* - each of the component machines can calculate its next states and outputs independently of the other component machine, based only on the information about its own internal state and partial information about inputs (Fig.3.2),

(ii) ***serial full-decomposition*** - one of the component machines, called the *tail* or *dependent machine* (say M_2), uses the information about the outputs or states of the second machine, called the *head* or *independent machine* (say M_1), and partial information about inputs in order to calculate its next states and outputs (Fig.3.3),

(iii) ***general full-decomposition*** - each of the component machines uses information about the outputs or states of the other component machine and partial information about inputs in order to calculate its next states and outputs (Fig.3.4).

The parallel full-decomposition and the serial full-decomposition can be treated as special cases of a general full-decomposition with zero information about one submachine used by another submachine.

From the point of view of the sort of information about a given submachine used by another submachine in order to calculate its next states and outputs, we can distinguish the following two types of full-decomposition:

(i) the decomposition with **information about outputs**, called by us ***type O***,

(ii) the decomposition with **information about internal states**, called ***type S***.

A given submachine can use the information about the "present" or the "next" state or output of the other submachine. So, we distinguish the following two classes of full-decomposition:

(i) ***class P*** - a decompositions with information about the **present** state or output,

(ii) ***class N*** - a decompositions with information about the **next** state or output.

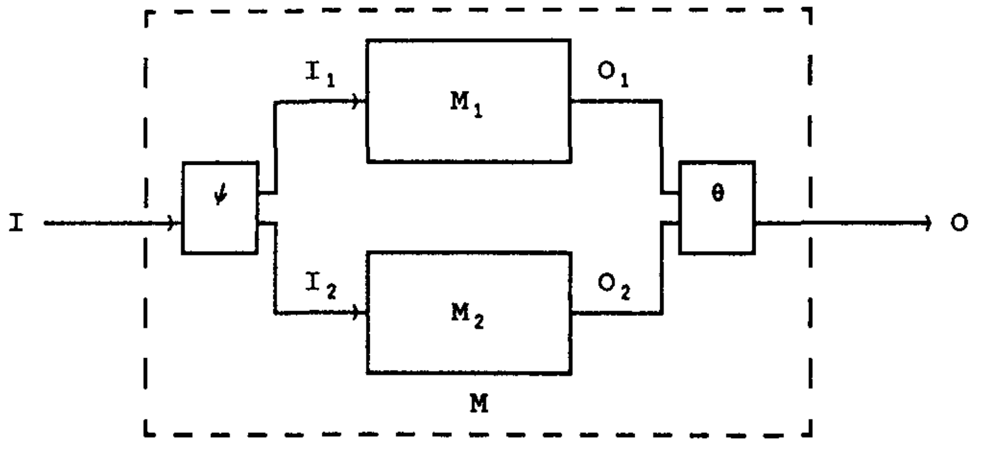

Fig 3.2 Parallel full-decomposition of a machine M into component machines M_1 and M_2.

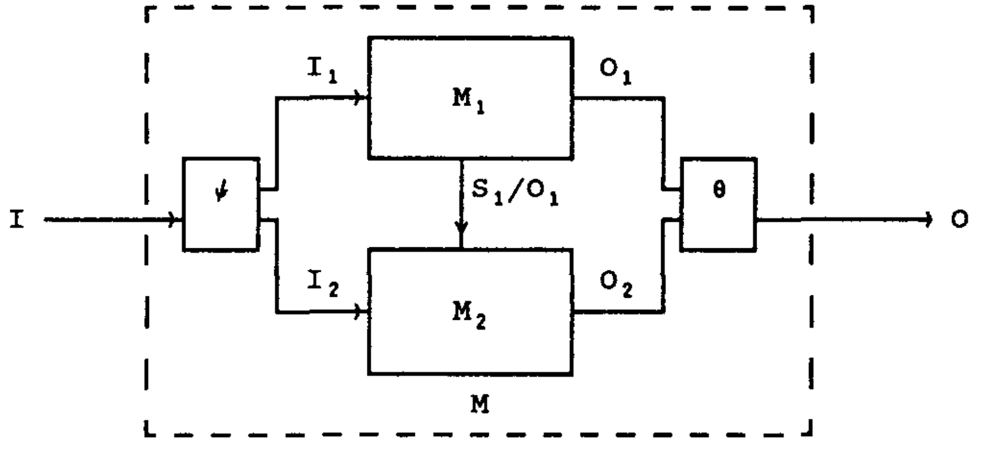

Fig 3.3 Serial full-decomposition of a machine M into component machines M_1 and M_2.

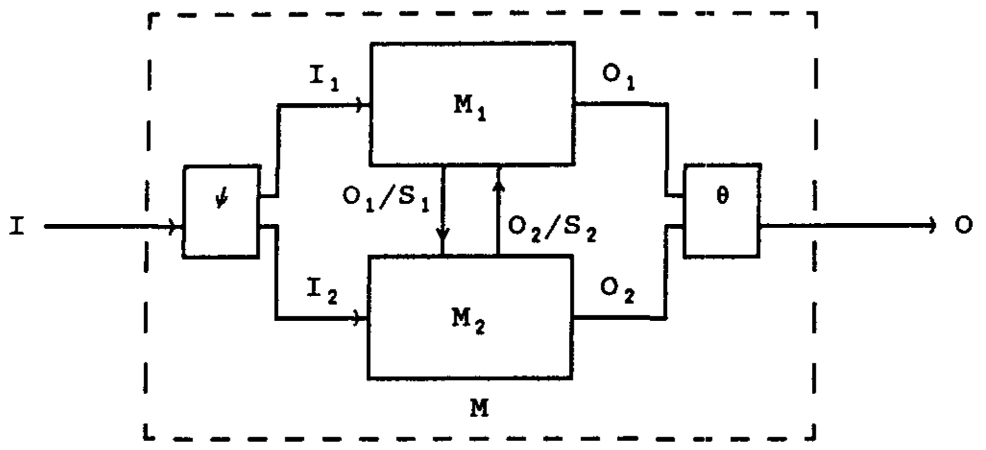

Fig 3.4 General full-decomposition of a machine M into component machines M_1 and M_2 .

From the classifications given above, it immediately follows, that the following cases of full-decompositions are feasible:
- one sort of parallel full-decomposition;
- four sorts of serial full decomposition: PS, NS, PO, and NO ,
- two sorts of general full-decomposition: PS, PO.

For a general full-decomposition, it is possible to have not only the "pure" cases PS and PO but also the "mixture" of types S and O and classes P and N (the first submachine can use the information about the state of the second and the second about the output of the first and vice versa ; the first submachine can use the information about the present state/output of the second submachine and the second can use the information about the next state/output of the first). In this report, we do not take into account "mixed" types, because definitions and theorems for them can be formulated easily as "mixtures" of the adequate definitions and theorems for the "pure" cases considered here.

The formal definitions of all types of full-decompositions which we consider in the paper are introduced below.

Let $s\epsilon S_1$, $t\epsilon S_2$, $x_1\epsilon I_1$, $x_2\epsilon I_2$.

<u>DEFINITION</u> <u>3.1</u> A *parallel connection* of two machines:

$$M_1 = (I_1,\ S_1,\ O_1,\ \delta^1,\ \lambda^1)$$

and

$$M_2 = (I_2,\ S_2,\ O_2,\ \delta^2,\ \lambda^2)$$

is the machine:

$$M_1||M_2 = (I_1xI_2, S_1xS_2, O_1xO_2, \delta^*, \lambda^*)$$

where:

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,x_1),\delta^2(t,x_2))$$

and

$$\lambda^*((s,t),(x_1,x_2)) = (\lambda^1(s,x_1),\lambda^2(t,x_2))$$

(for Mealy machine)

or

$$\lambda^*((s,t)) = (\lambda^1(s),\lambda^2(t))$$

(for Moore machine)

<u>DEFINITION</u> <u>3.2</u> The machine $M_1||M_2$ is a ***parallel full-decomposition*** of the machine M *if and only if* the parallel connection of M_1 and M_2 realizes M

<u>DEFINITION</u> <u>3.3</u> A *serial connection of type PS* of two machines:

$$M_1 = (I_1,\ S_1,\ O_1,\ \delta^1,\ \lambda^1)$$

and

$$M_2 = (I_2',\ S_2,\ O_2,\ \delta^2,\ \lambda^2)\ ,$$

for which $I_2' = S_1xI_2$,

is the machine $M_1 \longrightarrow M_2 = (I_1xI_2, S_1xS_2, O_1xO_2, \delta^*, \lambda^*)$,

where:

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,x_1),\delta^2(t,(s,x_2)))$$

and

$$\lambda^*((s,t),(x_1,x_2)) = (\lambda^1(s,x_1),\lambda^2(t,(s,x_2)))$$

(for a Mealy machine)

or

$$\lambda^*((s,t)) = (\lambda^1(s),\lambda^2(t))$$

(for a Moore machine).

<u>DEFINITION</u> <u>3.4</u> The machine $M_1 \longrightarrow M_2$ is a ***serial full-decomposition of type PS*** of the machine M *if and only if* the serial connection of type PS of M_1 and M_2 realizes M.

DEFINITION 3.5 A *serial connection of type NS* of two machines:

$$M_1 = (I_1, S_1, O_1, \delta^1, \lambda^1)$$

and

$$M_2 = (I_2', S_2, O_2, \delta^2, \lambda^2),$$

for which $I_2' = S_1 x I_2$,

is the machine $M_1 \longrightarrow M_2 = (I_1 x I_2, S_1 x S_2, O_1 x O_2, \delta^*, \lambda^*)$,

where:

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,x_1), \delta^2(t,(\delta^1(s,x_1),x_2))$$

and

$$\lambda^*((s,t),(x_1,x_2)) = (\lambda^1(s,x_1), \lambda^2(t,(\delta^1(s,x_1),x_2))$$

(for a Mealy machine)

or

$$\lambda^*((s,t)) = (\lambda^1(s), \lambda^2(t))$$

(for a Moore machine)

DEFINITION 3.6 The machine $M_1 \longrightarrow M_2$ is a *serial full-decomposition of type NS* of the machine M *if and only if* the serial connection of type NS of M_1 and M_2 realizes M.

DEFINITION 3.7 A *serial connection of type PO* of two machines:

$$M_1 = (I_1, S_1, O_1, \delta^1, \lambda^1)$$

and

$$M_2 = (I_2', S_2, O_2, \delta^2, \lambda^2) ,$$

for which $I_2' = O_1 x I_2$,

is the machine $M_1 \longrightarrow M_2 = (I_1 x I_2, S_1 x S_2, O_1 x O_2, \delta^*, \lambda^*)$,

where:

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,x_1), \delta^2(t,(y_1,x_2)))$$

$$\lambda^*((s,t),(x_1,x_2)) = (\lambda^1(s,x_1), \lambda^2(t,(y_1,x_2)))$$

and $y_1 \in O_1$: y_1 is the present output of M_1

(the output of M_1 contemporary with the state s of M_1)

(for a Mealy machine)

or

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,x_1), \delta^2(t,(\lambda^1(s),x_2))))$$

$$\lambda^*((s,t)) = (\lambda^1(s), \lambda^2(t))$$

(for a Moore machine)

<u>DEFINITION</u> <u>3.8</u> The machine $M_1 \longrightarrow M_2$ is a *serial full-decomposition of type PO* of the machine M *if and only if* the serial connection of type PO of M_1 and M_2 realizes M

<u>DEFINITION</u> <u>3.9</u> A *serial connection of type NO* of two machines:

$$M_1 = (I_1, S_1, O_1, \delta^1, \lambda^1)$$

and

$$M_2 = (I_2', S_2, O_2, \delta^2, \lambda^2),$$

for which $I_2' = O_1 x I_2$

is the machine $M_1 \longrightarrow M_2 = (I_1 x I_2, S_1 x S_2, O_1 x O_2, \delta^*, \lambda^*)$,

where:

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,x_1), \delta^2(t,(\lambda^1(s,x_1),x_2)))$$

$$\lambda^*((s,t),(x_1,x_2)) = (\lambda^1(s,x_1), \lambda^2(t,(\lambda^1(s,x_1),x_2)))$$

(for a Mealy machine)

or

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,x_1), \delta^2(t,(\lambda^1(\delta^1(s,x_1)),x_2)))$$

$$\lambda^*((s,t)) = (\lambda^1(s), \lambda^2(t))$$

(for a Moore machine)

<u>DEFINITION</u> <u>3.10</u> The machine $M_1 \longrightarrow M_2$ is a *serial full-decomposition of type NO* of the machine M *if and only if* the serial connection of type NO of M_1 and M_2 realizes M.

<u>DEFINITION</u> <u>3.11</u> A *general connection of type PS* of two machines :

$$M_1 = (I_1', S_1, O_1, \delta^1, \lambda^1)$$

and

$$M_2 = (I_2', S_2, O_2, \delta^2, \lambda^2)$$

where:

$$I_1' = S_2 x I_1 , \quad I_2' = S_1 x I_2 ,$$

is the machine:

$$M_1 \longleftrightarrow M_2 = (I_1 x I_2, S_1 x S_2, O_1 x O_2, \delta^*, \lambda^*) ,$$

where:

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,(t,x_1)), \delta^2(t,(s,x_2))$$

and

$$\lambda^*((s,t),(x_1,x_2)) = (\lambda^1(s,(t,x_1)), \lambda^2(t,(s,x_2))$$

(for a Mealy machine)

or

$$\lambda^*((s,t)) = (\lambda^1(s), \lambda^2(t))$$

(for a Moore machine)

DEFINITION 3.12 The machine $M_1 \leftrightarrow M_2$ is a *general full-decomposition of type PS* of the machine M *if and only if* the general connection of type PS of M_1 and M_2 realizes M.

DEFINITION 3.13 A *general connection of type PO* of two machines:

$$M_1 = (I_1', S_1, O_1, \delta^1, \lambda^1)$$

and

$$M_2 = (I_2', S_2, O_2, \delta^2, \lambda^2) ,$$

where:

$$I_1' = O_2 x I_1 , I_2' = O_1 x I_2$$

is the machine:

$$M_1 \leftrightarrow M_2 = (I_1 x I_2, S_1 x S_2, O_1 x O_2, \delta^*, \lambda^*) ,$$

where:

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,(y_2,x_1)),\delta^2(t,(y_1,x_2)))$$
$$\lambda^*((s,t),(x_1,x_2)) = (\lambda^1(s,(y_2,x_1)),\lambda^2(t,(y_1,x_2)))$$

and $y_1 \epsilon O_1$, $y_2 \epsilon O_2$ (present outputs of M_1 and M_2)

(for a Mealy machine)

or

$$\delta^*((s,t),(x_1,x_2)) = (\delta^1(s,(\lambda^2(t),x_1)),\delta^2(t,(\lambda^1(s),x_2)))$$
$$\lambda^*((s,t)) = (\lambda^1(s),\lambda^2(t))$$

(for a Moore machine)

DEFINITION 3.14 The machine $M_1 \leftrightarrow M_2$ is a *general full decomposition of type PO* of the machine M *if and only if* the general connection of type PO of machines M_1 and M_2 realizes M.

Each of the above defined types of a full-decomposition can be considered as a full-decomposition with the realization of the output behaviour or as a full-decomposition with the realization of the state and output behaviour. In next paragraphs, we will formulate and prove, for the case of state and output behaviour realizations, the theorems about the existence of different types of full- decomposition defined above. In order to formulate these theorems we will introduce the notions of "output-dependent trinity", "state-dependent trinity", "semitrinity" and "induced semitrinity". Only the proves for a Mealy machine are presented in the report, because the proves for a Moore machine are analogous. The theorems for the case of output behaviour realizations will be presented in a separate report.

4. Partitions, partition pairs and partition trinities.

The concepts of partitions and partition pairs introduced by Hartmanis [11][12] and partition trinities introduced by Hou [14][15] are very useful tools for analyzing the information flow in machines and between machines; therefore they will be used in this work.
Let S be any set of elements.

DEFINITION 4.1 *Partition* π on S is defined as follows:

$$\pi = \{B_i \mid B_i \subseteq S \text{ and } B_i \cap B_j = 0 \text{ for } i \neq j \text{ and } \bigcup_i B_i = S\},$$

i.e. a partition π on S is a set of disjoint subsets of S whose set union is S.

For a given $s \in S$, the block of a partition π containing s is denoted as $[s]\pi$ and we will write $[s]\pi = [t]\pi$ to denote that s and t are in the same block of π. Similarly, the block of a partition π containing S', where $S' \subseteq S$, is denoted by $[S']\pi$.

The partition containing only one element of S in each block is called a *zero partition* and denoted by $\pi_S(0)$. The partition containing all the elements of S in one block is called a *one partition* and is denoted by $\pi_S(I)$.

Let π_1 and π_2 be two partitions on S.

DEFINITION 4.2 *Partition product* $\pi_1 \cdot \pi_2$ is the partition on S such that $[s]\pi_1 \cdot \pi_2 = [t]\pi_1 \cdot \pi_2$ *if and only if* $[s]\pi_1 = [t]\pi_1$ and $[s]\pi \cdot =$ $[t]\pi_2$.

DEFINITION 4.3 *Partition sum* $\pi_1 + \pi_2$ is the partition on S such that $[s]\pi_1 + \pi_2 = [t]\pi_1 + \pi_2$ *if and only if* a sequence: $s=s_0, s_1, \ldots, s_n=t$, $s_i \in S$ for i=1..n , exists for which either
$[s_i]\pi_1 = [s_{i+1}]\pi_1$ either $[s_i]\pi_2 = [s_{i+1}]\pi_2$, $0 \leq i \leq n-1$.

From the above definitions, it follows that the blocks of $\pi_1 \cdot \pi_2$ are obtained by intersecting the blocks of π_1 and π_2, while the blocks of $\pi_1+\pi_2$ are obtained by making union of all those blocks of π_1 and π_2 which contain common elements.

__DEFINITION__ __4.4__ π_2 is *greater than or equal to* π_1: $\pi_1 \le \pi_2$ *if and only if* each block of π_1 is included in a block of π_2.

Thus $\pi_1 \le \pi_2$ *if and only if* $\pi_1 \cdot \pi_2 = \pi_1$ *if and only if* $\pi_1+\pi_2 = \pi_2$.

Let S_π be the set of all partitions on S. Because the relation $\le$ is a relation of *partial ordering* (i.e. it is reflexive, antisymmetric and transitive), $(S_\pi, \le)$ is a *partially ordered set.*

Let $(Z, \le)$ be a partially ordered set and T be a subset of Z.

__DEFINITION__ __4.5__ z, $z \in Z$, is the *least upper bound (LUB)* of T *if and only if* :

(i) $\forall t \in T$: $z \ge t$,

(ii) $\forall t \in T$: *if* $z' \ge t$ *then* $z' \ge z$.

z, $z \in Z$, is the *greatest lower bound (GLB)* of T *if and only if*:

(i) $\forall t \in T$: $z \le t$,

(ii) $\forall t \in T$: *if* $z' \le t$ *then* $z' \le z$.

__DEFINITION__ __4.6__ A partially ordered set $L = (Z, \le)$, which has a LUB and a GLB for every pair of elements, is called a *lattice.*

It is evident that the set of all partitions on S together with the relation of a partial ordering $\le$ form a lattice with $GLB(\pi_1,\pi_2) = \pi_1 \cdot \pi_2$ and $LUB(\pi_1,\pi_2) = \pi_1+\pi_2$.

Let π_S, τ_S, π_I, π_O be the partitions on $M=(I, S, O, \delta, \lambda)$, in particular: π_S, τ_S on S, π_I on I, π_O on O.

__DEFINITION__ __4.7__

(i) (π_S,τ_S) is an __*S-S partition pair*__ *if and only if* $\forall B \in \pi_S \ \forall x \in I : B\bar{\delta}_x \subseteq B', \ B' \in \tau_S$.

(ii) (π_I,π_S) is an __*I-S partition pair*__ *if and only if* $\forall A \in \pi_I \ \forall s \in S : s\bar{\delta}_A \subseteq B \ , \ B \in \pi_S$.

(iii) (π_S, π_0) is an ***S-O** **partition** **pair*** *if and only if*
$\forall B \in \pi_S \; \forall x \in I : B\overline{\lambda}_x \subseteq C \;,\; C \in \pi_0$ (Mealy case)
or
$\forall B \in \pi_S : B\overline{\lambda} \subseteq C \;,\; C \in \pi_0$ (Moore case).

(iv) (π_I, π_0) is an ***I-O** **partition** **pair*** *if and only if*
$\forall A \in \pi_I \; \forall s \in S : s\overline{\lambda}_A \subseteq C \;,\; C \in \pi_0$ (Mealy case)
or
$\forall A \in \pi_I \; \forall s \in S : s\lambda \subseteq C \;,\; C \in \pi_0$ (Moore case).

The practical meaning of the notions introduced above is as follows:

(π_S, τ_S) is an S-S partition pair *if and only if* the blocks of π_S are mapped by M into the blocks of τ_S. Thus, if we know the block of π_S which contains the present state of the machine M and we know the present input of M, we can compute unambiguously the block of τ_S which contains the next state of M for the states from a given blocks of π_S and a given input. The interpretation of the notions of I-S, S-O and I-O partition pairs is similar.

In the case of Moore machine, the definition of an I-O pair is trivial, besause each (π_I, π_S) satisfies it (the output of M is defined by the state of M unambiguously).

DEFINITION 4.8 Partition π_S has a *substitution property* (it is an *SP-partition*) *if and only if* (π_S, π_S) is an S-S pair.

DEFINITION 4.9 *Partition trinity* $T = (\pi_I, \pi_S, \pi_0)$ on the machine $M = (I, S, O, \delta, \lambda)$ is an ordered triple of partitions on sets I, S and O, respectively, which satisfies the following conditions:

$\forall A \in \pi_I \; \forall B \in \pi_S : B\overline{\delta}_A \subseteq B', \; B' \in \pi_S$ *and* $B\overline{\lambda}_A \subseteq C \;,\; C \in \pi_0$.

Thus, if (π_I, π_S, π_0) is a partition trinity on M and we know the block B of π_S which contains the present state of M and we know the block A of π_I which contains the present input of M, we can compute unambiguously block B' of π_S containing the next state of M and block C of π_0 containing the output of M for the states from block B and inputs from block A.

For completely spacified machines, it has been proved that

(π_I, π_S, π_O) is a partition trinity on M *if and only if* (π_S, π_S) is an S-S pair, (π_I, π_S) is an I-S pair , (π_S, π_O) is an S-O pair and (π_I, π_O) is an I-O pair on M [14][15].

It has been shown in [14] that the set of trinities on a machine M forms a finite trinity lattice with

$$GLB(T_1, T_2) = T_1 \odot T_2 \quad \text{and} \quad LUB(T_1, T_2) = T_1 \oplus T_2 ,$$

where $\odot$ and $\oplus$ are defined as a collection of pairwise operations "$\cdot$" and "+" on partitions of the same type (input,state,output) of trinities of T_1 and T_2 .

5. Parallel full-decomposition.

An important theorem about the existence of a parallel full-decomposition has been proved in [14] and [15]. Below we will introduce a similar theorem. The differences between this theorem and that proved in [14] and [15] are following: we did not require $\pi_I \cdot \tau_I = \pi_I(0)$, which was required in [14] and [15] and we defined the nontriviality of a full decomposition in another way. This means that the theorem below is formulated with weaker assumptions and therefore it is satisfied for a broader class of cases.

THEOREM 5.1 A machine $M = (I,S,O,\delta,\lambda)$ has a nontrivial parallel full-decomposition with the realization of the state and output behaviour *if* two partition trinities on M: (π_I, π_S, π_O) and (τ_I, τ_S, τ_O) exist and they satisfy the following conditions:

(i) $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$,

(ii) $|\pi_I|<|I| \wedge |\tau_I|<|I| \vee |\pi_S|<|S| \wedge |\tau_S|<|S| \vee |\pi_O|<|O| \wedge |\tau_O|<|O|$.

Proof of theorem 5.1 is similar to that for the appropriate theorem presented in [14] and [15].

The interpretation of theorem 5.1 is as follows.

Let $M_1 = (\pi_I, \pi_S, \pi_O, \delta^1, \lambda^1)$ and $M_2 = (\tau_I, \tau_S, \tau_O, \delta^2, \lambda^2)$, where:

$$B1\delta^1{}_{A1} = B1\bar{\delta}_{A1} \ , \ B1\lambda^1{}_{A1} = B1\bar{\lambda}_{A1} \ ,$$

$$B2\delta^2{}_{A2} = B2\bar{\delta}_{A2} \ , \ B2\lambda^2{}_{A2} = B2\bar{\lambda}_{A2} \ ,$$

for all $A1 \epsilon \pi_I$, $B1 \epsilon \pi_S$, $A2 \epsilon \tau_I$, $B2 \epsilon \tau_S$

and let M be a parallel connection of M_1 and M_2

Since (π_I, π_S, π_O) is a partition trinity, based only on the information about the block of π_I containing the input of M and the block of π_S containing the present state of M (i.e information about the input and present state of M_1) machine M_1 can calculate unambiguously the block of π_S in which the next state of M is contained and the block of π_O that contains the output of M for the input from a given block of π_I and the present state from a given block of π_S (i.e. M_1 can calculate its next state and output). Similarly, since (τ_I, τ_S, τ_O) is a partition trinity, machine M_2, based only on the information about its input and present state (i.e. knowledge of the adequate block of τ_I and block of τ_S), can calculate its next state and output (i.e. the adequate blocks of τ_S and τ_O).

Since $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$, having the knowledge of the block of π_S and the block of τ_S in which the state of M is contained, it is possible to calculate this state and, having the knowledge of the block of π_O and the block of τ_O in which the output of M is contained it is possible to calculate this output. So, the machines M_1 and M_2 together can calculate the next state and output of M unambiguously.

The special case of theorem 5.1 for:

$$|\pi_I|<|I| \wedge |\tau_I|<|I| \wedge (|\pi_S|=|S| \wedge |\pi_O|=|O| \vee |\tau_S|=|S| \wedge |\tau_O|=|O|)$$

express, in fact, the input redundancy. In this case machine M should be replaced with machine M_1 or M_2, having fewer inputs and realizing M, instead to be decomposed. Similar special cases exist for all other theorems presented in this report.

6. Serial full-decomposition of type PS.

Let τ_I, τ_S, τ_O be partitions on a machine M on I, S and O respectively.

DEFINITION 6.1 (τ_I, τ_S, τ_O) is a *partition semitrinity if and only if* τ_I, τ_S and τ_O satisfy the following conditions:

(i) (τ_I, τ_S) is an I-S partition pair,

(ii) (τ_I, τ_O) is an I-O partition pair (for a Mealy machine),
or
(τ_S, τ_O) is a S-O partition pair (for a Moore machine)

In other words, (τ_I, τ_S, τ_O) is a semitrinity if and only if, based only on the knowledge of the block of a partition τ_I containing the input of M and the knowledge of the present state of M, it is possible to calculate the block of τ_S in which the next state of M will be contained and, in the case of a Mealy machine, based on the same information, it is possible to calculate the block of τ_O in which the output of M will be contained for the given input and state or, in the case of Moore machine, based on the knowledge of the block of a partition τ_S in which the state of M is contained, it is possible to calculate the block of τ_O in which the output of M will be contained for the state from a given block of τ_S. The triple of partitions (τ_I, τ_S, τ_O) is called "semitrinity", because it has to satisfy half of the conditions for a trinity.

THEOREM 6.1 A machine M has a nontrivial serial full-decomposition of type PS with the realization of the state and output behaviour *if* a partition trinity (π_I, π_S, π_O) and a partition semitrinity (τ_I, τ_S, τ_O) exist and they satisfy the following conditions:

(i) $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$,

(ii) $|\pi_I|<|I| \wedge |\pi_S| \cdot |\tau_I|<|I| \vee |\pi_S|<|S| \wedge |\tau_S|<|S| \vee |\pi_O|<|O| \wedge$
$\wedge |\tau_O|<|O|$.

<u>Proof</u> (for the case of a Mealy machine)

Let $M_1 = (\pi_I, \pi_S, \pi_0, \delta^1, \lambda^1)$ and $M_2 = (\pi_S x \tau_I, \tau_S, \tau_0, \delta^2, \lambda^2)$ be two machines satisfying the following conditions:

(1) (π_I, π_S, π_0) and (τ_I, τ_S, τ_0) satisfy the conditions of the theorem 6.1 ,

(2) $\forall B1 \epsilon \pi_S \; \forall A1 \epsilon \pi_I \; : \; B1\delta^1{}_{A1} = [B1\overline{\delta}_{A1}]\pi_S \; , \; B1\lambda^1{}_{A1} = [B1\overline{\lambda}_{A1}]\pi_0$,

(3) $\forall B1 \epsilon \pi_S \; \forall B2 \epsilon \tau_S \; \forall A2 \epsilon \tau_I$:

$B2\delta^2{}_{(B1,A2)} = [(B1 \cap B2)\overline{\delta}_{A2}]\tau_S , \; B2\lambda^2{}_{(B1,A2)} = [(B1 \cap B2)\overline{\lambda}_{A2}]\tau_0$.

Since (π_I, π_S, π_0) is a partition trinity (1), $B1\overline{\delta}_{A1}$ is placed in just one block of π_S and $B1\overline{\lambda}_{A1}$ in only one block of π_0 . This means, that $B1\delta^1{}_{A1}$ and $B1\lambda^1{}_{A1}$ are defined unambiguously.

Since (τ_I, τ_S, τ_0) is a semitrinity and $\pi_S \cdot \tau_S = \pi_S(0)$ (1), $(B1 \cap B2)\overline{\delta}_{A2}$ is placed in just one block of τ_S and $(B1 \cap B2)\overline{\lambda}_{A2}$ is placed in only one block of τ_0 . This means, that $B2\delta^2{}_{(B1,A2)}$ and $B2\lambda^2{}_{(B1,A2)}$ are defined unambigously.

Let $\psi: I \longrightarrow \pi_I x \tau_I$ be an injective function,

$\phi: \pi_S x \tau_S \longrightarrow S$ be a surjective partial function,

$\theta: \pi_0 x \tau_0 \longrightarrow O$ be a surjective partial function

and

(4) $\psi(x) = ([x]\pi_I, [x]\tau_I)$,

(5) $\phi(B1,B2) = B1 \cap B2$ if $B1 \cap B2 \neq 0$,

(6) $\theta(C1,C2) = C1 \cap C2$ if $C1 \cap C2 \neq 0$.

We will prove below that the serial connection of defined above machines M_1 and M_2 realizes machine M.

Since $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_0 \cdot \tau_0 = \pi_0(0)$ (1) , ϕ and θ are one-to-one functions and for $B1 \cap B2 \neq 0$ and $C1 \cap C2 \neq 0$:

(7) $\phi(B1,B2) \epsilon S$, $\theta(C1,C2) \epsilon O$.

Therefore, $\forall B1 \epsilon \pi_S \; \forall B2 \epsilon \tau_S \; \forall x \epsilon I$ and $B1 \cap B2 \neq 0$:

$\phi((B1,B2)\delta^*{}_{\psi(x)}) =$

$= \phi((B1,B2)\delta^*{}_{([x]\pi_I,[x]\tau_I)})$ ((4))

$= \phi(B1\delta^1{}_{[x]\pi_I}, B2\delta^2{}_{(B1,[x]\tau_I)})$ (definition 3.3)

$= B1\delta^1{}_{[x]\pi_I} \cap B2\delta^2{}_{(B1,[x]\tau_I)}$ ((5))

$= [B1\overline{\delta}_{[x]\pi_I}]\pi_S \cap [(B1\cap B2)\overline{\delta}_{[x]\tau_I}]\tau_S$ ((2), (3))

$= [B1\overline{\delta}_x]\pi_S \cap [(B1\cap B2)\overline{\delta}_x]\tau_S$ ($B\overline{\delta}_x \subseteq B\overline{\delta}_{[x]\pi}$)

$= [(B1\cap B2)\overline{\delta}_x]\pi_S \cap [(B1\cap B2)\overline{\delta}_x]\tau_S$ ($B1\cap B2 \subseteq B1$)

$= [(B1\cap B2)\delta_x]\pi_S \cap [(B1\cap B2)\delta_x]\tau_S$ ((7))

$= (B1\cap B2)\delta_x$ ($\pi_S\cdot\tau_S=\pi_S(0)$)

$= \phi(B1,B2)\delta_x$ ((5))

and similary:

$\theta((B1,B2)\lambda^*{}_{\psi(x)}) =$

$= \theta((B1,B2)\lambda^*{}_{([x]\pi_I,[x]\tau_I)})$ ((4))

$= \theta(B1\lambda^1{}_{[x]\pi_I}, B2\lambda^2{}_{(B1,[x]\tau_I)})$ (definition 3.3)

$= B1\lambda^1{}_{[x]\pi_I} \cap B2\lambda^2{}_{(B1,[x]\tau_I)}$ ((6))

$= [B1\overline{\lambda}_{[x]\pi_I}]\pi_0 \cap [(B1\cap B2)\overline{\lambda}_{[x]\tau_I}]\tau_0$ ((2), (3))

$= [B1\overline{\lambda}_x]\pi_0 \cap [(B1\cap B2)\overline{\lambda}_x]\tau_0$ ($B\overline{\lambda}_x \subseteq B\overline{\lambda}_{[x]\pi}$)

$= [(B1\cap B2)\overline{\lambda}_x]\pi_0 \cap [(B1\cap B2)\overline{\lambda}_x]\tau_0$ ($B1\cap B2 \subseteq B1$)

$= [(B1\cap B2)\lambda_x]\pi_0 \cap [(B1\cap B2)\lambda_x]\tau_0$ ((7))

$= (B1\cap B2)\lambda_x$ ($\pi_0\cdot\tau_0=\pi_0(0)$)

$= \phi(B1,B2)\lambda_x$ ((5))

From the above calculations and definitions 2.3, 3.3 and 3.4, it follows immediately that the serial connection of type PS of machines M_1 and M_2 realizes M, i.e. M has a serial full-decomposition of type PS. If condition (ii) of theorem 6.1 is satisfied, the decomposition is nontrivial. □

Theorem 6.1 has a straightforward interpretation.

Since (π_I, π_S, π_0) is a partition trinity, based only on the information about the block of a partition π_I containing the input and the block of a partition π_S containing the present state of machine M (i.e. information about the input and present state of M_1), machine M_1 can calculate unambiguously the block of π_S in which the next state of M is contained and the block of π_0 in which the output of M is contained for the given input and present state (i.e M_1 can calculate its next state and output).

Since (τ_I, τ_S, τ_O) is a partition semitrinity and $\tau_S \cdot \pi_S = \pi_S(0)$, based only on the information about the block of a partition τ_I containing the input and the blocks of partitions τ_S and π_S containing the present state of the machine M (i.e. information about the primary input and the present state of M_2 and about the present state of M_1 which is a part if the input of M_2), machine M_2 can calculate unambiguously the block of τ_S in which the next state of M is contained and, in the case of a Mealy machine, the block of τ_O in which the output of M is contained for the given input and present state (i.e. M_2 can calculate its next state and output). In the case of a Moore machine, M_2 can calculate the block of τ_O in which the output of M is contained, based only on information about the block of τ_S in which the state of M is contained.

Since $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$, having information about the blocks of π_S and π_O calculated by M_1 and the blocks of τ_S and τ_O calculated by M_2 (i.e. information about the next states and outputs of M_1 and M_2) it is possible to calculate unambiguously the next states and outputs of machine M.

In [14], for the Mealy case, the other theorem about the existence of a serial full-decomposition of type PS has been proved. However, theorem 6.1 includes also the Moore case and two important differences occur between our theorem 6.1 and the one proved in [14].

In theorem 6.1 we did not use the notion of "forced-trinity" which was used in [14] - instead, we introduced the notion of "semitrinity". This notion is natural, simple and posesses a straightforward interpretation.

We formulated and proved theorem 6.1 with weaker assumptions (for example we did not require $\pi_I \cdot \tau_I = \pi_I(0)$, as was required in [14]). This means that theorem 4.1 is more general than the one proved in [14].

7. Serial full-decomposition of type NS.

Let τ_I, τ_S, τ_O be partitions on machine M, on I, S and O respectiviely, and ξ_S be another partition on S.

DEFINITION 7.1 (τ_I, τ_S, τ_O) is a (next) *state-dependent trinity* for *an independent state partition* ξ_S *if and only if* τ_I, τ_S, τ_O satisfy one of the following conditions for a given ξ_S:

(i) $\forall s,t \in S \ \forall x_1,x_2 \in I$:

if $[s]\tau_S=[t]\tau_S \wedge [x_1]\tau_I=[x_2]\tau_I \wedge [s\delta_{x_1}]\xi_S=[t\delta_{x_2}]\xi_S$

then $[s\delta_{x_1}]\tau_S=[t\delta_{x_2}]\tau_S \wedge [s\lambda_{x_1}]\tau_O=[t\lambda_{x_2}]\tau_O$

(for a Mealy machine),

(ii) $\forall s,t \in S \ \forall x_1,x_2 \in I$:

if $[s]\tau_S=[t]\tau_S \wedge [x_1]\tau_I=[x_2]\tau_I \wedge [s\delta_{x_1}]\xi_S=[t\delta_{x_2}]\xi_S$

then $[s\delta_{x_1}]\tau_S=[t\delta_{x_2}]\tau_S \wedge [(s\delta_{x_1})\lambda]\tau_O=[(t\delta_{x_2})\lambda]\tau_O$

(for a Moore machine).

In other words, (τ_I, τ_S, τ_O) is a state-dependent trinity for an independent state partition ξ_S if and only if, based only on the knowledge of the block of a partition τ_I containing the input of machine M , knowledge of the block of a partition τ_S containing the present state of M and knowledge of the block of a partition ξ_S in which the next state of M is contained for a given input and state, it is possible to calculate the block of τ_S in which the next state of M will be contained and the block of τ_O in which the output of M will be contained.

THEOREM 7.1 A machine M has a nontrivial serial full-decomposition of type NS with the realization of the state and output behaviour *if* such a partition trinity (π_I, π_S, π_O) and such a state-dependent trinity (τ_I, τ_S, τ_O) for $\xi_S=\pi_S$ exist that the following conditions are satisfied:

(i) $\pi_S \cdot \tau_S = \pi_S(0)$ *and* $\pi_O \cdot \tau_O = \pi_O(0)$,

(ii) $|\pi_I|<|I|$, $|\pi_S|<|S|$, $|\pi_O|<|O|$, $|\pi_S|\cdot|\tau_I|<|I|$, $|\tau_S|<|S|$, $|\tau_O|<|O|$.

<u>Proof</u> (for the case of a Mealy machine)

Let $M_1 = (\pi_I, \pi_S, \pi_O, \delta^1, \lambda^1)$ and $M_2 = (\tau_I, \tau_S, \tau_O, \delta^2, \lambda^2)$ be two machines for which the following conditions are satisfied:

(1) (π_I, π_S, π_O) and (τ_I, τ_S, τ_O) satisfy the conditions of the theorem 7.1 ,

(2) $\forall B1 \in \pi_S\ \forall A1 \in \pi_I$: $B1\delta^1{}_{A1} = [B1\overline{\delta}_{A1}]\pi_S$, $B1\lambda^1{}_{A1} = [B1\lambda_{A1}]\pi_O$,

(3) $\forall B2 \in \tau_S\ \forall A2 \in \tau_I\ \forall B1' \in \pi_S$:

$$B2\delta^2{}_{(B1',A2)} = [\{s\delta_x |\ s \in B2,\ x \in A2,\ s\delta_x \in B1'\}]\tau_S ,$$
$$B2\lambda^2{}_{(B1',A2)} = [\{s\lambda_x |\ s \in B2,\ x \in A2,\ s\delta_x \in B1'\}]\tau_O .$$

Since (π_I, π_S, π_O) is a partition trinity (1), $B1\overline{\delta}_{A1}$ is placed in just one block of π_S and $B1\overline{\lambda}_{A1}$ is placed in only one block of π_O. This means that $B1\delta^1{}_{A1}$ and $B1\lambda^1{}_{A1}$ are defined unambiguously.

Since (τ_I, τ_S, τ_O) is a state dependent trinity for $\xi_S = \pi_S$ (1), the following condition is satisfied:

(4) $\forall s,t \in S\ \forall x_1, x_2 \in I$:

if $[s]\tau_S = [t]\tau_S \wedge [x_1]\tau_I = [x_2]\tau_I \wedge [s\delta_{x_1}]\pi_S = [t\delta_{x_2}]\pi_S$

then $[s\delta_{x_1}]\tau_S = [t\delta_{x_2}]\tau_S \wedge [s\lambda_{x_1}]\tau_O = [t\lambda_{x_2}]\tau_O$.

From (4), it follows that $B2\delta^2{}_{(B1',A2)}$ and $B2\lambda^2{}_{(B1',A2)}$ are defined unambiguously because $\{s\delta_x |\ s \in B2,\ x \in A2,\ s\delta_x \in B1'\}$ is located in only one block of τ_S and $\{s\lambda_x |\ s \in B2,\ x \in A2,\ s\delta_x \in B1'\}$ in just one block of τ_O.

Let ψ: $I \longrightarrow \pi_I x \tau_I$ be an injective function,
ϕ: $\pi_S x \tau_S \longrightarrow S$ be a surjective partial function,
θ: $\pi_O x \tau_O \longrightarrow O$ be a surjective partial function

and

(5) $\psi(x) = ([x]\pi_I, [x]\tau_I)$,

(6) $\phi(B1,B2) = B1 \cap B2$ if $B1 \cap B2 \neq 0$,

(7) $\theta(C1,C2) = C1 \cap C2$ if $C1 \cap C2 \neq 0$.

Since $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$ (1) , ϕ and θ are one-to-one and for $B1 \cap B2 \neq 0$ and $C1 \cap C2 \neq 0$:

(8) $\phi(B1,B2)\epsilon S$, $\theta(C1,C2)\epsilon O$.

Therefore, $\forall B1\epsilon\pi_s$ $\forall B2\epsilon\tau_s$ $\forall x\epsilon I$ and $B1\cap B2 \neq 0$:

$$\phi((B1,B2)\delta^*_{\psi(x)}) =$$

$$= \phi((B1,B2)\delta^*_{([x]\pi_I,[x]\tau_I)}) \qquad ((5))$$

$$= \phi(B1\delta^1_{[x]\pi_I}, B2\delta^2_{(B1\delta^1_{[x]\pi_I},[x]\tau_I)}) \qquad \text{(definition 3.5)}$$

$$= B1\delta^1_{[x]\pi_I} \cap B2\delta^2_{(B1\delta^1_{[x]\pi_I},[x]\tau_I)} \qquad ((6))$$

$$= [B1\bar{\delta}_{[x]\pi_I}]\pi_s \cap [\{s\delta_x|\ s\epsilon B2\wedge s\delta_y\epsilon[B1\bar{\delta}_{[y]\pi_I}]\pi_s\wedge y\epsilon[x]\tau_I\}]\tau_s \qquad ((2),(3))$$

$$= [B1\bar{\delta}_x]\pi_s \cap [\{s\delta_x|\ s\epsilon B2\wedge s\delta_x\epsilon[B1\bar{\delta}_x]\pi_s\}]\tau_s \qquad (B\bar{\delta}_x \subseteq B\delta_{[x]\pi})$$

$$= [B1\bar{\delta}_x]\pi_s \cap [\{s\delta_x|\ s\epsilon B2 \wedge s\epsilon B1\}]\tau_s \qquad (\pi_s \text{ is SP-partition})$$

$$= [(B1\cap B2)\bar{\delta}_x]\pi_s \cap [(B1\cap B2)\bar{\delta}_x]\tau_s \qquad (\ B1\cap B2 \subseteq B1\ )$$

$$= [(B1\cap B2)\delta_x]\pi_s \cap [(B1\cap B2)\delta_x]\tau_s \qquad ((8))$$

$$= (B1\cap B2)\delta_x \qquad (\ \pi_s\cdot\tau_s=\pi_s(0)\ )$$

$$= \phi(B1,B2)\delta_x \qquad ((6))$$

and similary:

$$\theta((B1,B2)\lambda^*_{\psi(x)}) =$$

$$= \theta((B1,B2)\lambda^*_{([x]\pi_I,[x]\tau_I)}) \qquad ((5))$$

$$= \theta(B1\lambda^1_{[x]\pi_I}, B2\lambda^2_{(B1\delta^1_{[x]\pi_I},[x]\tau_I)}) \qquad \text{(definition 3.5)}$$

$$= B1\lambda^1_{[x]\pi_I} \cap B2\lambda^2_{(B1\delta^1_{[x]\pi_I},[x]\tau_I)} \qquad ((7))$$

$$= [B1\bar{\lambda}_{[x]\pi_I}]\pi_0 \cap [\{s\lambda_x|\ s\epsilon B2\wedge s\delta_y\epsilon[B1\bar{\delta}_{[y]\pi_I}]\pi_s\wedge y\epsilon[x]\tau_I\}]\tau_0 \qquad ((2),(3))$$

$$= [B1\bar{\lambda}_x]\pi_0 \cap [\{s\lambda_x|\ s\epsilon B2\wedge s\delta_x\epsilon[B1\bar{\delta}_x]\pi_s\}]\tau_0 \qquad (B\bar{\delta}_x \subseteq B\delta_{[x]\pi})$$

$$= [B1\bar{\lambda}_x]\pi_0 \cap [\{s\lambda_x|\ s\epsilon B2 \wedge s\epsilon B1\}]\tau_0 \qquad (\pi_s \text{ is SP-partition})$$

$$= [(B1\cap B2)\bar{\lambda}_x]\pi_0 \cap [(B1\cap B2)\bar{\lambda}_x]\tau_0 \qquad (\ B1\cap B2 \subseteq B1\ )$$

$$= [(B1\cap B2)\lambda_x]\pi_0 \cap [(B1\cap B2)\lambda_x]\tau_0 \qquad ((8))$$

$$= (B1\cap B2)\lambda_x \qquad (\ \pi_s\cdot\tau_s=\pi_s(0)\ )$$

$$= \phi(B1,B2)\lambda_x \qquad ((6))$$

From the above calculations and definitions 2.3, 3.5 and 3.6, it follows immediately that the serial connection of type NS of machines M_1 and M_2 realizes M, i.e. M has a serial full-decomposition of type NS. If condition (ii) of the theorem 7.1 is satisfied, the decomposition is nontrivial. □

Theorem 7.1 has a straightforward interpretation.

Since (π_I,π_S,π_O) is a partition trinity, machine M_1, based only on the information about its input and present state (i.e. knowledge of the adequate block of π_I and block of π_S), can calculate its next state and output (i.e. the adequate blocks of π_S and π_O).

Since (τ_I,τ_S,τ_O) is a state-dependent partition trinity for $\xi_S=\pi_S$, based only on information about the block of τ_I containing the input, the block of τ_S containing the present state of M and the block of π_S containing the next state of M for the given input and present state (i.e. information about the primary input and present state of M_2 and the next state of M_1 which is part of the input of M_2), machine M_2 can calculate unambiguously the block of τ_S in which the next state of M is contained and the block of τ_O in which the output of M is contained for the given input and present state (i.e. M_2 can calculate its next state and output).

Since $\tau_S\cdot\pi_S = \pi_S(0)$ and $\tau_O\cdot\pi_O = \pi_O(0)$, having information about blocks of π_S and π_O calculated by M_1 and blocks of τ_S and τ_O calculated by M_2 , it is possible to calculate unambiguously the next states and outputs of machine M.

8. Serial full-decomposition of type PO.

Let $\pi_S^!$ and ξ_O be partitions on M on S and O respectively.

DEFINITION 8.1 $\pi_S^!$ is *a state partition induced by an output partition* ξ_O *if and only if* one of the following conditions is satisfied:

(i) $\forall s,t\in S\ \forall x,y\in I$: *if* $[s\lambda_x]\xi_O = [t\lambda_y]\xi_O$ *then* $[s\delta_x]\pi_S^! = [t\delta_y]\pi_S^!$ (for a Mealy machine),

(ii) $\forall s,t\in S$: $[s]\pi_S^! = [t]\pi_S^!$ *if and only if* $[s\lambda]\xi_O = [t\lambda]\xi_O$ (for a Moore machine) .

In other words, if $\pi_s^!$ is a state partition induced by an output partition ξ_0 and if we know that the present output y of M is contained in a block C: $C\epsilon\xi_0$ then we know that the present state s of M is contained in a block B: $B\epsilon\pi_s^!$, which is indicated unambiguously by block C. We can say, that *block* B of $\pi_s^!$ *is induced by block* C of ξ_0 and denote this by: B = ind(C).

Let τ_I, τ_s, τ_0 be partitions on a machine M, on I, S and O respectively, and ξ_0 be the other partition on O.

<u>DEFINITION</u> <u>8.2</u> (τ_I,τ_s,τ_0) is a ***partition semitrinity induced by an output partition*** ξ_0 *if and only if* such a state partition $\pi_s^!$ induced by ξ_0 exists, that τ_I, τ_s and τ_0 satisfy the following conditions for this $\pi_s^!$:

(i) (τ_I,τ_s) is an I-S partition pair,

(ii) $(\tau_s\cdot\pi_s',\tau_s)$ is a S-S partition pair,

(iii) $(\tau_s\cdot\pi_s',\tau_0)$ is a S-O partition pair,

and

(τ_I,τ_0) is an I-O partition pair (for a Mealy machine),

or

(τ_s,τ_0) is a S-O partition pair (for a Moore machine).

In other words, (τ_I,τ_s,τ_0) is a semitrinity induced by an output partition ξ_0 if and only if, based on the knowledge of the block of a partition τ_I containing the input of M and the knowledge of the block of a partition τ_s and the block of an induced partition $\pi_s^!$ containing the present state of M, it is possible to calculate the block of τ_s in which the next state of M will be contained and, in the case of a Mealy machine, based on the same information it is possible to calculate the block of τ_0 in which the output of M will be contained for the given input and state or, in the case of a Moore machine, based on the knowledge of the blocks of partitions τ_s and π_s' containing the state of M, it is possible to calculate the block of τ_0 containing the output of M for the given state.

<u>THEOREM</u> <u>8.1</u> A machine M has a nontrivial serial full-decomposition of type PO with the realization of the state and output behaviour *if* such a partition trinity (π_I, π_S, π_O) and such a partition semitrinity (τ_I, τ_S, τ_O) induced by $\xi_O = \pi_O$ exist that the following conditions are satisfied:

(i) $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$,

(ii) $|\pi_I|<|I| \wedge |\pi_O| \cdot |\tau_I|<|I| \vee |\pi_S|<|S| \wedge |\tau_S|<|S| \vee |\pi_O|<|O| \wedge$
$\wedge |\tau_O|<|O|$.

<u>Proof</u> (for the case of a Mealy machine)

Let $M_1 = (\pi_I, \pi_S, \pi_O, \delta^1, \lambda^1)$ and $M_2 = (\pi_O x \tau_I, \tau_S, \tau_O, \delta^2, \lambda^2)$ be the two machines for which the following conditions are satisfied:

(1) (π_I, π_S, π_O) and (τ_I, τ_S, τ_O) satisfy the conditions of the theorem 8.1 ,

(2) $\forall B1 \epsilon \pi_S \; \forall A_1 \epsilon \pi_I \; : \; B1\delta^1{}_{A1} = [B1\overline{\delta}_{A1}]\pi_S$, $B1\lambda^1{}_{A1} = [B1\overline{\lambda}_{A1}]\pi_O$,

(3) $\forall C1 \epsilon \pi_O \; \forall B2 \epsilon \tau_S \; \forall A2 \epsilon \tau_I$:

$$B2\delta^2{}_{(C1,A2)} = [\{s\delta_x | \; s\epsilon B2 \wedge s\epsilon ind(C1) \wedge x\epsilon A2\}]\tau_S,$$
$$B2\lambda^2{}_{(C1,A2)} = [\{s\lambda_x | \; s\epsilon B2 \wedge s\epsilon ind(C1) \wedge x\epsilon A2\}]\tau_O.$$

Since (π_I, π_S, π_O) is a partition trinity (1), $B1\delta^1{}_{A1}$ and $B1\lambda^1{}_{A1}$ are defined unambiguously.

Since (τ_I, τ_S, τ_O) is a semitrinity induced by $\xi_O = \pi_O$ (1), the following conditions are satisfied:

(4) $(\tau_S \cdot \pi_S', \tau_S)$ is a S-S pair and $(\tau_S \cdot \pi_S', \tau_O)$ is a S-O pair,

(5) (τ_I, τ_S) is an I-S pair,

(6) (τ_I, τ_O) is an I-O pair.

From (4) and (5), it follows that $\{s\delta_x | \; s\epsilon B2 \wedge s\epsilon ind(C1) \wedge x\epsilon A2\}$ is located in just one block of τ_S. From (4) and (6), it follows that $\{s\lambda_x | \; s\epsilon B2 \wedge s\epsilon ind(C1) \wedge x\epsilon A2\}$ is located in just one block of τ_O. This means, that $B2\delta^2{}_{(C1,A2)}$ and $B2\lambda^2{}_{(C1,A2)}$ are defined unambigously

Let ψ: $I \longrightarrow \pi_I x \tau_I$ be an injective function,
ϕ: $\pi_S x \tau_S \longrightarrow S$ be a surjective partial function,
θ: $\pi_O x \tau_O \longrightarrow O$ be a surjective partial function

and

(7) $\psi(x) = ([x]\pi_I, [x]\tau_I)$,

(8) $\phi(B1,B2) = B1 \cap B2$ if $B1 \cap B2 \neq 0$,

(9) $\theta(C1,C2) = C1 \cap C2$ if $C1 \cap C2 \neq 0$.

Since $\pi_s \cdot \tau_s = \pi_s(0)$ and $\pi_0 \cdot \tau_0 = \pi_0(0)$ (1) , ϕ and θ are one-to-one functions and

(10) $\phi(B1,B2) \in S$, $\theta(C1,C2) \in O$.

Therefore, $\forall C1 \in \pi_0\ \forall B1 \in \pi_s\ \forall B2 \in \tau_s\ \forall x \in I$ and $B1 \cap B2 \neq 0$:

$$
\begin{aligned}
&\phi((B1,B2)\delta^*_{\psi(x)}) = \\
&= \phi((B1,B2)\delta^*_{([x]\pi_I,[x]\tau_I)}) && ((7)) \\
&= \phi(B1\delta^1_{[x]\pi_I}, B2\delta^2_{(C1,[x]\tau_I)}) && (\text{definition } 3.7) \\
&= B1\delta^1_{[x]\pi_I} \cap B2\delta^2_{(C1,[x]\tau_I)} && ((8)) \\
&= [B1\overline{\delta}_{[x]\pi_I}]\pi_s \cap [(\text{ind}(C1)\cap B2)\overline{\delta}_{[x]\tau_I}]\tau_s && ((2),\ (3)) \\
&= [B1\overline{\delta}_x]\pi_s \cap [(\text{ind}(C1)\cap B2)\overline{\delta}_x]\tau_s && (\ B\overline{\delta}_x \subseteq B\overline{\delta}_{[x]\pi}\ ) \\
&= [(B1\cap B2)\overline{\delta}_x]\pi_s \cap [(\text{ind}(C1)\cap B2)\overline{\delta}_x]\tau_s && (\ B1\cap B2 \subseteq B1\ ) \\
&= [(B1\cap B2)\overline{\delta}_x]\pi_s \cap [(B1\cap B2)\overline{\delta}_x]\tau_s && (\ B1\cap B2 \subseteq \text{ind}(C1)\cap B2\ ) \\
& && ((4),\ (10)) \\
&= [(B1\cap B2)\delta_x]\pi_s \cap [(B1\cap B2)\delta_x]\tau_s && ((10)) \\
&= (B1\cap B2)\delta_x && (\ \pi_s\cdot\tau_s=\pi_s(0)\ ) \\
&= \phi(B1,B2)\delta_x && ((8))
\end{aligned}
$$

and similary:

$$
\begin{aligned}
&\theta((B1,B2)\lambda^*_{\psi(x)}) = \\
&= \theta((B1,B2)\lambda^*_{([x]\pi_I,[x]\tau_I)}) && ((7)) \\
&= \theta(B1\lambda^1_{[x]\pi_I}, B2\lambda^2_{(C1,[x]\tau_I)}) && (\text{definition } 3.7) \\
&= B1\lambda^1_{[x]\pi_I} \cap B2\lambda^2_{(C1,[x]\tau_I)} && ((9)) \\
&= [B1\overline{\lambda}_{[x]\pi_I}]\pi_0 \cap [(\text{ind}(C1)\cap B2)\overline{\lambda}_{[x]\tau_I}]\tau_0 && ((2),\ (3)) \\
&= [B1\overline{\lambda}_x]\pi_0 \cap [(\text{ind}(C1)\cap B2)\overline{\lambda}_x]\tau_0 && (\ B\overline{\lambda}_x \subseteq B\overline{\lambda}_{[x]\pi}\ ) \\
&= [(B1\cap B2)\overline{\lambda}_x]\pi_0 \cap [(\text{ind}(C1)\cap B2)\overline{\lambda}_x]\tau_0 && (\ B1\cap B2 \subseteq B1\ ) \\
&= [(B1\cap B2)\overline{\lambda}_x]\pi_0 \cap [(B1\cap B2)\overline{\lambda}_x]\tau_0 && (\ B1\cap B2 \subseteq \text{ind}(C1)\cap B2\ ) \\
& && ((4),\ (10)) \\
&= [(B1\cap B2)\lambda_x]\pi_0 \cap [(B1\cap B2)\lambda_x]\tau_0 && ((10)) \\
&= (B1\cap B2)\lambda_x && (\ \pi_0\cdot\tau_0=\pi_0(0)\ )
\end{aligned}
$$

$= \phi(B1,B2)\lambda_x$ ((8))

From the above calculations and definitions 2.3, 3.7 and 3.8, it follows immediately that the serial connection of type PO of machines M_1 and M_2 realizes M, i.e. M has a serial full-decomposition of type PO. If condition (ii) of theorem 8.1 is satisfied, the decomposition is nontrivial. □

The interpretation of theorem 8.1 is as follows:

Since (π_I, π_S, π_O) is a partition trinity, machine M_1, based only on the information about its input and present state (i.e. knowledge of the adequate block of π_I and block of π_S), can calculate its next state and output (i.e. the adequate blocks of π_S and π_O).

Since (τ_I, τ_S, τ_O) is a partition semitrinity induced by π_O and $\tau_S \cdot \pi_S^! = \pi_S(0)$, where $\pi_S^!$ is the state partition induced by π_O, based only on the information about the block of a partition τ_I containing the input and the blocks of partitions τ_S and $\pi_S^!$ containing the present state of the machine M (i.e. information about the primary input and the present state of M_2 and about the present output of M_1 which is a part if the input of M_2), machine M_2 can calculate unambiguously the block of τ_S in which the next state of M will be contained and, in the case of Mealy machine, the block of τ_O in which the output of M will be contained for the given input and present state (i.e. M_2 can calculate its next state and output). In the case of Moore machine, M_2 can calculate the block of τ_O in which the output of M will be contained based only on information about the block of τ_S in which the state of M is contained.

Since $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$, having information about blocks of π_S and π_O calculated by M_1 and blocks of τ_S and τ_O calculated by M_2 , it is possible to calculate unambiguously the next states and outputs of the machine M.

9. Serial full-decomposition of type NO.

Let τ_I, τ_S, τ_O be partitions on a machine M, on I, S, O respectiviely, and ξ_O be the other partition on O.

DEFINITION 9.1 (τ_I, τ_S, τ_O) is an *output-dependent trinity* for the *independent output partition* ξ_O *if and only if* τ_I, τ_S and τ_O satisfy one of the following conditions for a given ξ_O:

(i) $\forall s,t \in S \ \forall x_1, x_2 \in I$:

if $[s]\tau_S = [t]\tau_S \wedge [x_1]\tau_I = [x_2]\tau_I \wedge [s\lambda_{x_1}]\xi_O = [t\lambda_{x_2}]\xi_O$

then $[s\delta_{x_1}]\tau_S = [t\delta_{x_2}]\tau_S \wedge [s\lambda_{x_1}]\tau_O = [t\lambda_{x_2}]\tau_O$

(for a Mealy machine),

(ii) $\forall s,t \in S \ \forall x_1, x_2 \in I$:

if $[s]\tau_S = [t]\tau_S \wedge [x_1]\tau_I = [x_2]\tau_I \wedge [(s\delta_{x_1})\lambda]\xi_O = [(t\delta_{x_2})\lambda]\xi_O$

then $[s\delta_{x_1}]\tau_S = [t\delta_{x_2}]\tau_S \wedge [(s\delta_{x_1})\lambda]\tau_O = [(t\delta_{x_2})\lambda]\tau_O$

(for a Moore machine).

In other words, (τ_I, τ_S, τ_O) is an output-dependent trinity for the independent output partition ξ_O if and only if, based on the *knowledge of the block of a partition* τ_I *in which* the input of a machine M is contained, the block of a partition τ_S in which the present state of M is contained and the block of a partition ξ_O in which the outputs of M are contained for inputs from a given block of τ_I and states from a given block of τ_S, it is possible to calculate the block of τ_S in which the next state of M is contained and the block of τ_O in which the output of M is contained for the present state from a given block of τ_S and input from a given block of τ_I.

<u>THEOREM</u> <u>9.1</u> A machine M has a nontrivial serial full-decomposition of type NO with the realization of the state and output behaviour *if* such a partition trinity (π_I, π_S, π_O) and such an output-dependent trinity (τ_I, τ_S, τ_O) for $\xi_O = \pi_O$ exist that the following conditions are satisfied:

(i) $\pi_S \cdot \tau_S = \pi_S(0)$ *and* $\pi_O \cdot \tau_O = \pi_O(0)$,

(ii) $|\pi_I|<|I| \wedge |\pi_O| \cdot |\tau_I|<|I| \vee |\pi_S|<|S| \wedge |\tau_S|<|S| \vee |\pi_O|<|O| \wedge$
$\wedge |\tau_O|<|O|$.

<u>Proof</u> (for the case of Mealy machine)

Let $M_1 = (\pi_I, \pi_S, \pi_O, \delta^1, \lambda^1)$ and $M_2 = (\tau_I, \tau_S, \tau_O, \delta^2, \lambda^2)$ be two machines for which the following conditions are satisfied:

(1) (π_I, π_S, π_O) and (τ_I, τ_S, τ_O) satisfy the conditions of theorem 9.1 ,

(2) $\forall B1 \epsilon \pi_S \; \forall A1 \epsilon \pi_I$: $B1\delta^1_{A1} = [B1\overline{\delta}_{A1}]\pi_S \wedge B1\lambda^1_{A1} = [B1\overline{\lambda}_{A1}]\pi_O$,

(3) $\forall B2 \epsilon \tau_S \; \forall A2 \epsilon \tau_I \; \forall C1 \epsilon \pi_O$:

$$B2\delta^2_{(C1,A2)} = [\{s\delta_x | \; s \epsilon B2, \; x \epsilon A2, \; s\lambda_x \epsilon C1 \;\}]\tau_S \;,$$
$$B2\lambda^2_{(C1,A2)} = [\{s\lambda_x | \; s \epsilon B2, \; x \epsilon A2, \; s\lambda_x \epsilon C1 \;\}]\tau_O \;.$$

Since (π_I, π_S, π_O) is a partition trinity (1), $B1\overline{\delta}_{A1}$ is placed in just one block of π_S and $B1\overline{\lambda}_{A1}$ is placed in just one block of π_O. This means that $B1\delta^1_{A1}$ and $B1\lambda^1_{A1}$ are unambiguously defined.

Since (τ_I, τ_S, τ_O) is an output dependent trinity for $\xi_O = \pi_O$ (1), the following condition is satisfied:

(4) $\forall s,t \epsilon S \; \forall x_1, x_2 \epsilon I$:

if $[s]\tau_S = [t]\tau_S \wedge [x_1]\tau_I = [x_2]\tau_I \wedge [s\lambda_{x_1}]\pi_O = [t\lambda_{x_2}]\pi_O$

then $[s\delta_{x_1}]\tau_S = [t\delta_{x_2}]\tau_S \wedge [s\lambda_{x_1}]\tau_O = [t\lambda_{x_2}]\tau_O$.

From (4), it follows that $B2\delta^2_{(C1,A2)}$ and $B2\lambda^2_{(C1,A2)}$ are defined unambiguously, because $\{s\delta_x | \; s \epsilon B2, \; x \epsilon A2, \; s\lambda_x \epsilon C1\}$ is located in just one block of τ_S and $\{s\lambda_x | \; s \epsilon B2, \; x \epsilon A2, \; s\lambda_x \epsilon C1\}$ in just one block of τ_O.

Let ψ: $I \longrightarrow \pi_I x \tau_I$ be an injective function,

ϕ: $\pi_s x \tau_s \longrightarrow S$ be a surjective partial function,

θ: $\pi_0 x \tau_0 \longrightarrow O$ be a surjective partial function

and

(5) $\psi(x) = ([x]\pi_I, [x]\tau_I)$,

(6) $\phi(B1,B2) = B1 \cap B2$ if $B1 \cap B2 \neq 0$,

(7) $\theta(C1,C2) = C1 \cap C2$ if $C1 \cap C2 \neq 0$.

Since $\pi_s \cdot \tau_s = \pi_s(0)$ and $\pi_0 \cdot \tau_0 = \pi_0(0)$ (1) , ϕ and θ are one-to-one and

(8) $\phi(B1,B2) \in S$, $\theta(C1,C2) \in O$.

Therefore $\forall B1 \in \pi_s$ $\forall B2 \in \tau_s$ $\forall x \in I$ and $B1 \cap B2 \neq 0$:

$\phi((B1,B2)\delta^*_{\psi(x)})$ =

$= \phi((B1,B2)\delta^*_{([x]\pi_I,[x]\tau_I)})$ ((5))

$= \phi(B1\delta^1_{[x]\pi_I}, B2\delta^2_{(B1\lambda^1_{[x]\pi_I},[x]\tau_I)})$ (definition 3.9)

$= B1\delta^1_{[x]\pi_I} \cap B2\delta^2_{(B1\lambda^1_{[x]\pi_I},[x]\tau_I)}$ ((6))

$= [B1\bar{\delta}_{[x]\pi_I}]\pi_s \cap [\{s\delta_x | s \in B2 \wedge s\lambda_y \in [B1\bar{\lambda}_{[y]\pi_I}]\pi_0 \wedge y \in [x]\tau_I\}]\tau_s$ ((2),(3))

$= [B1\bar{\delta}_x]\pi_s \cap [\{s\delta_x | s \in B2 \wedge s\lambda_x \in [B1\bar{\lambda}_x]\pi_s\}]\tau_0$ $(B\bar{\lambda}_x \subseteq B\lambda_{[x]\pi})$

$= [B1\bar{\delta}_x]\pi_s \cap [\{s\delta_x | s \in B2 \wedge s \in B1\}]\tau_s$ ((π_s,π_0) is SO-pair)

$= [(B1 \cap B2)\bar{\delta}_x]\pi_s \cap [(B1 \cap B2)\bar{\delta}_x]\tau_s$ ($B1 \cap B2 \subseteq B1$)

$= [(B1 \cap B2)\delta_x]\pi_s \cap [(B1 \cap B2)\delta_x]\tau_s$ ((8))

$= (B1 \cap B2)\delta_x$ ($\pi_s \cdot \tau_s = \pi_s(0)$)

$= \phi(B1,B2)\delta_x$ ((6))

and similary:

$\theta((B1,B2)\lambda^*_{\psi(x)})$ =

$= \theta((B1,B2)\lambda^*_{([x]\pi_I,[x]\tau_I)})$ ((5))

$= \theta(B1\lambda^1_{[x]\pi_I}, B2\lambda^2_{(B1\lambda^1_{[x]\pi_I},[x]\tau_I)})$ (definition 3.9)

$= B1\lambda^1_{[x]\pi_I} \cap B2\lambda^2_{(B1\lambda^1_{[x]\pi_I},[x]\tau_I)}$ ((7))

$= [B1\bar{\lambda}_{[x]\pi_I}]\pi_0 \cap [\{s\lambda_x | s \in B2 \wedge s\lambda_y \in [B1\bar{\lambda}_{[y]\pi_I}]\pi_0 \wedge y \in [x]\tau_I\}]\tau_0$ ((2),(3))

$= [B1\bar{\lambda}_x]\pi_0 \cap [\{s\lambda_x | s \in B2 \wedge s\lambda_x \in [B1\bar{\lambda}_x]\pi_0\}]\tau_0$ $(B\bar{\lambda}_x \subseteq B\lambda_{[x]\pi})$

$= [B1\overline{\lambda}_x]\pi_0 \cap [\{s\lambda_x | s\in B2 \wedge s\in B1\}]\tau_0$ $((\pi_s,\pi_0)$ is SO-pair)

$= [(B1\cap B2)\overline{\lambda}_x]\pi_0 \cap [(B1\cap B2)\overline{\lambda}_x]\tau_0$ $(B1\cap B2 \subseteq B1)$

$= [(B1\cap B2)\lambda_x]\pi_0 \cap [(B1\cap B2)\lambda_x]\tau_0$ ((8))

$= (B1\cap B2)\lambda_x$ $(\pi_s \cdot \tau_s = \pi_s(0))$

$= \phi(B1,B2)\lambda_x$ ((6))

From the above calculations and definitions 2.3, 3.9 and 3.10, it follows immediately that the serial connection of type NO of machines M_1 and M_2 realizes M, i.e. M has a serial full-decomposition of type NO. If condition (ii) of the theorem 9.1 is satisfied, the decomposition is nontrivial. □

Theorem 9.1 has a straightforward interpretation.

Since (π_I,π_s,π_0) is a partition trinity, machine M_1, based only on the information about its input and present state (i.e. knowledge of the adequate block of π_I and block of π_s), can calculate its next state and output (i.e. the adequate blocks of π_s and π_0).

Since (τ_I,τ_s,τ_0) is an output-dependent partition trinity for $\xi_0=\pi_0$, based only on information about the block of τ_I containing the input, the block of τ_s containing the present state of M and the block of π_0 containing the output of M for the given input and present state (i.e. information about the primary input and present state of M_2 and the output of M_1 which is a part of the input of M_2), machine M_2 can calculate unambiguously the block of τ_s in which the next state of M is contained and the block of τ_0 in which the output of M is contained for the given input and present state (i.e. M_2 can calculate its next state and output).

Since $\tau_s \cdot \pi_s = \pi_s(0)$ and $\tau_0 \cdot \pi_0 = \pi_0(0)$, having information about blocks of π_s and π_0 calculated by M_1 and blocks of τ_s and τ_0 calculated by M_2 , it is possible to calculate unambiguously the next states and outputs of the machine M.

10. General full-decomposition of type PS

THEOREM 10.1 A machine M has a nontrivial general full-decomposition of type PS with the realization of the state and output behaviour *if* two partition semitrinities: (π_I, π_S, π_O) and (τ_I, τ_S, τ_O) exist and they satisfy the following conditions:

(i) $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$,

(ii) $|\tau_S| \cdot |\pi_I| < |I| \wedge |\pi_S| \cdot |\tau_I| < |I| \vee |\pi_S| < |S| \wedge |\tau_S| < |S| \vee |\pi_O| < |O| \wedge$
$\wedge |\tau_O| < |O|$.

Proof (for the case of a Mealy machine)

Let $M_1 = (\tau_S x \pi_I, \pi_S, \pi_O, \delta^1, \lambda^1)$ and $M_2 = (\pi_S x \tau_I, \tau_S, \tau_O, \delta^2, \lambda^2)$ be the two machines for which the following conditions are satisfied:

(1) (π_I, π_S, π_O) and (τ_I, τ_S, τ_O) satisfy the conditions of theorem 10.1 ,

(2) $\forall B1 \in \pi_S \ \forall B2 \in \tau_S \ \forall A_1 \in \pi_I$:
$B1\delta^1_{(B2,A1)} = [(B1 \cap B2)\overline{\delta}_{A1}]\pi_S$, $B1\lambda^1_{(B2,A1)} = [(B1 \cap B2)\overline{\lambda}_{A1}]\pi_O$,

(3) $\forall B1 \in \pi_S \ \forall B2 \in \tau_S \ \forall A2 \in \tau_I$:
$B2\delta^2_{(B1,A2)} = [(B1 \cap B2)\overline{\delta}_{A2}]\tau_S$, $B2\lambda^2_{(B1,A2)} = [(B1 \cap B2)\overline{\lambda}_{A2}]\tau_O$.

Since (π_I, π_S, π_O) and (τ_I, τ_S, τ_O) are semitrinities and $\pi_S \cdot \tau_S = \pi_S(0)$ (1), $(B1 \cap B2)\overline{\delta}_{A1}$ is placed in just one block of π_S, $(B1 \cap B2)$ is placed in just one block of π_O, $(B1 \cap B2)\overline{\delta}_{A2}$ is placed in only one block of τ_S and $(B1 \cap B2)\overline{\lambda}_{A2}$ is placed in only one block of τ_O . This means, that $B1\delta^1_{(B2,A1)}$, $B1\lambda^1_{(B2,A1)}$, $B2\delta^2_{(B1,A2)}$ and $B2\lambda^2_{(B1,A2)}$ are defined unambiguously.

Let ψ: $I \longrightarrow \pi_I x \tau_I$ be an injective function,
ϕ: $\pi_S x \tau_S \longrightarrow S$ be a surjective partial function,
θ: $\pi_O x \tau_O \longrightarrow O$ be a surjective partial function

and

(4) $\psi(x) = ([x]\pi_I, [x]\tau_I)$,

(5) $\phi(B1,B2) = B1 \cap B2$ if $B1 \cap B2 \neq 0$,

(6) $\theta(C1,C2) = C1 \cap C2$ if $C1 \cap C2 \neq 0$.

Because $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_0 \cdot \tau_0 = \pi_0(0)$ (1) , ϕ and θ are one-to-one functions and

(7) $\quad \phi(B1,B2) \in S \;,\; \theta(C1,C2) \in O$.

Therefore $\forall B1 \in \pi_S$ $\forall B2 \in \tau_S$ $\forall x \in I$ and $B1 \cap B2 \neq 0$:

$$\phi((B1,B2)\delta^*_{\psi(x)}) =$$

$$= \phi((B1,B2)\delta^*_{([x]\pi_I,[x]\tau_I)}) \qquad ((4))$$

$$= \phi(B1\delta^1_{(B2,[x]\pi_I)}, B2\delta^2_{(B1,[x]\tau_I)}) \qquad \text{(definition 3.11)}$$

$$= B1\delta^1_{(B2,[x]\pi_I)} \cap B2\delta^2_{(B1,[x]\tau_I)} \qquad ((5))$$

$$= [(B1 \cap B2)\overline{\delta}_{[x]\pi_I}]\pi_S \cap [(B1 \cap B2)\overline{\delta}_{[x]\tau_I}]\tau_S \qquad ((2),\ (3))$$

$$= [(B1 \cap B2)\overline{\delta}_x]\pi_S \cap [(B1 \cap B2)\overline{\delta}_x]\tau_S \qquad (\ B\overline{\delta}_x \subseteq B\overline{\delta}_{[x]\pi}\ )$$

$$= [(B1 \cap B2)\delta_x]\pi_S \cap [(B1 \cap B2)\delta_x]\tau_S \qquad ((7))$$

$$= (B1 \cap B2)\delta_x \qquad (\ \pi_S \cdot \tau_S = \pi_S(0)\ )$$

$$= \phi(B1,B2)\delta_x \qquad ((5))$$

and similary:

$$\theta((B1,B2)\lambda^*_{\psi(x)}) =$$

$$= \theta((B1,B2)\lambda^*_{([x]\pi_I,[x]\tau_I)}) \qquad ((4))$$

$$= \theta(B1\lambda^1_{(B2,[x]\pi_I)}, B2\lambda^2_{(B1,[x]\tau_I)}) \qquad \text{(definition 3.11)}$$

$$= B1\lambda^1_{(B2,[x]\pi_I)} \cap B2\lambda^2_{(B1,[x]\tau_I)} \qquad ((6))$$

$$= [(B1 \cap B2)\overline{\lambda}_{[x]\pi_I}]\pi_0 \cap [(B1 \cap B2)\overline{\lambda}_{[x]\tau_I}]\tau_0 \qquad ((2),\ (3))$$

$$= [(B1 \cap B2)\overline{\lambda}_x]\pi_0 \cap [(B1 \cap B2)\overline{\lambda}_x]\tau_0 \qquad (\ B\overline{\lambda}_x \subseteq B\overline{\lambda}_{[x]\pi}\ )$$

$$= [(B1 \cap B2)\lambda_x]\pi_0 \cap [(B1 \cap B2)\lambda_x]\tau_0 \qquad ((7))$$

$$= (B1 \cap B2)\lambda_x \qquad (\ \pi_0 \cdot \tau_0 = \pi_0(0)\ )$$

$$= \phi(B1,B2)\lambda_x \qquad ((5))$$

From the above calculations and definitions 2.3, 3.11 and 3.12, it follows immediately that the general connection of type PS of machines M_1 and M_2 realizes M, i.e. M has a general full-decomposition of type PS. If condition (ii) of theorem 10.1 is satisfied, the decomposition is nontrivial. □

The interpretation of theorem 10.1 is similar to the interpretation of theorem 6.1.

11. General full-decomposition of type PO

THEOREM 11.1 A machine M has a nontrivial general full-decomposition of type PO with the realization of the state and output behaviour *if* two partition semitrinities (π_I, π_S, π_O) induced by $\xi_{O2} = \tau_O$ and (τ_I, τ_S, τ_O) induced by $\xi_{O1} = \pi_O$ exist and they satisfy the following conditions:

(i) $\pi_S \cdot \tau_S = \pi_S(0)$ and $\pi_O \cdot \tau_O = \pi_O(0)$,

(ii) $|\tau_O| \cdot |\pi_I| < |I| \wedge |\pi_O| \cdot |\tau_I| < |I| \vee |\pi_S| < |S| \wedge |\tau_S| < |S| \vee |\pi_O| < |O| \wedge$
$\wedge |\tau_O| < |O|$.

Proof (for the case of a Mealy machine)

Let $M_1 = (\tau_O x \pi_I, \pi_S, \pi_O, \delta^1, \lambda^1)$ and $M_2 = (\pi_O x \tau_I, \tau_S, \tau_O, \delta^2, \lambda^2)$ be the two machines for which the following conditions are satisfied:

(1) (π_I, π_S, π_O) and (τ_I, τ_S, τ_O) satisfy the conditions of theorem 11.1 ,

(2) $\forall C2 \epsilon \tau_O \ \forall B1 \epsilon \pi_S \ \forall A_1 \epsilon \pi_I$:

$$B1\delta^1_{(C2,A1)} = [\{s\delta_x | \ s\epsilon B1 \wedge s\epsilon ind(C2) \wedge x\epsilon A1]\pi_S ,$$
$$B1\lambda^1_{(C2,A1)} = [\{s\lambda_x | \ s\epsilon B1 \wedge s\epsilon ind(C2) \wedge x\epsilon A1]\pi_O ,$$

(3) $\forall C1 \epsilon \pi_O \ \forall B2 \epsilon \tau_S \ \forall A2 \epsilon \tau_I$:

$$B2\delta^2_{(C1,A2)} = [\{s\delta_x | \ s\epsilon B2 \wedge s\epsilon ind(C1) \wedge x\epsilon A2\}]\tau_S ,$$
$$B2\lambda^2_{(C1,A2)} = [\{s\lambda_x | \ s\epsilon B2 \wedge s\epsilon ind(C1) \wedge x\epsilon A2\}]\tau_O .$$

Since (π_I, π_S, π_O) is a semitrinity induced by $\xi_{O2}=\tau_O$ and (τ_I, τ_S, τ_O) is a semitrinity induced by $\xi_{O1}=\pi_O$ (1), the following conditions are satisfied:

(4) $(\pi_S' \cdot \tau_S, \tau_S)$ is a S-S pair,

(5) $(\pi_S \cdot \tau_S', \pi_S)$ is a S-S pair,

(6) $(\pi_s'\cdot\tau_s,\tau_0)$ is a S-O pair,
(7) $(\pi_s\cdot\tau_s',\pi_0)$ is a S-O pair,
(8) (π_I,π_s) is an I-S pair,
(9) (π_I,π_0) is an I-O pair,
(10) (τ_I,τ_s) is an I-S pair,
(11) (τ_I,τ_0) is an I-O pair.

From (5) and (8), it follows that $\{s\delta_x \mid s\in B1 \wedge s\in ind(C2) \wedge x\in A1\}$ is located in just one block of π_s. From (7) and (9), it follows that $\{s\lambda_x \mid s\in B1 \wedge s\in ind(C2) \wedge x\in A1\}$ is located in only one block of π_0. This means, that $B1\delta^1{}_{(C2,A1)}$ and $B1\lambda^1{}_{(C2,A1)}$ are unambiguously defined.

Similarly, from (4) and (10), it follows that $\{s\delta_x \mid s\in B2 \wedge s\in ind(C1) \wedge x\in A2\}$ is located in just one block of τ_s and, from (6) and (11), it follows that
$\{s\lambda_x \mid s\in B2 \wedge s\in ind(C1) \wedge x\in A2\}$ is located in just one block of τ_0. So, $B2\delta^2{}_{(C1,A2)}$ and $B2\lambda^2{}_{(C1,A2)}$ are unambigously defined.

Let ψ: $I \longrightarrow \pi_I x \tau_I$ be an injective function,
ϕ: $\pi_s x \tau_s \longrightarrow S$ be a surjective partial function,
θ: $\pi_0 x \tau_0 \longrightarrow O$ be a surjective partial function

and

(12) $\psi(x) = ([x]\pi_I,[x]\tau_I)$,
(13) $\phi(B1,B2) = B1\cap B2$ if $B1\cap B2 \neq 0$,
(14) $\theta(C1,C2) = C1\cap C2$ if $C1\cap C2 \neq 0$.

Since $\pi_s\cdot\tau_s = \pi_s(0)$ and $\pi_0\cdot\tau_0 = \pi_0(0)$ (1) , ϕ and θ are one-to-one functions and

(15) $\phi(B1,B2)\in S$, $\theta(C1,C2)\in O$.

Therefore, $\forall C1\in\pi_0$ $\forall C2\in\tau_0$ $\forall B1\in\pi_s$ $\forall B2\in\tau_s$ $\forall x\in I$ and $B1\cap B2 \neq 0$:

$\phi((B1,B2)\delta^*{}_{\psi(x)}) =$

$= \phi((B1,B2)\delta^*{}_{([x]\pi_I,[x]\tau_I)})$ ((12))

$= \phi(B1\delta^1{}_{(C2,[x]\pi_I)},B2\delta^2{}_{(C1,[x]\tau_I)})$ (definition 3.13)

$= B1\delta^1{}_{(C2,[x]\pi_I)} \cap B2\delta^2{}_{(C1,[x]\tau_I)}$ ((13))

$= [(ind(C2)\cap B1)\overline{\delta}_{[x]\pi_I}]\pi_s \cap [(ind(C1)\cap B2)\overline{\delta}_{[x]\tau_I}]\tau_s$ ((2),(3))

$= [(ind(C2)\cap B1)\overline{\delta}_x]\pi_s \cap [(ind(C1)\cap B2)\overline{\delta}_x]\tau_s$ $(B\overline{\delta}_x \subseteq B\overline{\delta}_{[x]\pi})$

$= [(B1\cap B2)\overline{\delta}_x]\pi_s \cap [(B1\cap B2)\overline{\delta}_x]\tau_s$ ($B1\cap B2 \subseteq ind(C1)\cap B2$) ($B1\cap B2 \subseteq ind(C2)\cap B1$) ((4), (5), (15))

$= [(B1\cap B2)\delta_x]\pi_s \cap [(B1\cap B2)\delta_x]\tau_s$ ((15))

$= (B1 \cap B2)\delta_x$ $\qquad$ $(\pi_S \cdot \tau_S = \pi_S(0))$

$= \phi(B1,B2)\delta_x$ $\qquad$ $((13))$

and similary:

$$\theta((B1,B2)\lambda^*_{\psi(x)}) =$$

$= \theta((B1,B2)\lambda^*_{([x]\pi_I,[x]\tau_I)})$ $\qquad$ $((12))$

$= \theta(B1\lambda^1_{(c1,[x]\pi_I)}, B2\lambda^2_{(c1,[x]\tau_I)})$ $\qquad$ (definition 3.13)

$= B1\lambda^1_{(c1,[x]\pi_I)} \cap B2\lambda^2_{(c1,[x]\tau_I)}$ $\qquad$ $((14))$

$= [(ind(C2)\cap B1)\bar{\delta}_{[x]\pi_I}]\pi_0 \cap [(ind(C1)\cap B2)\bar{\lambda}_{[x]\tau_I}]\tau_0$ $\qquad$ $((2),(3))$

$= [(ind(C2)\cap B1)\bar{\lambda}_x]\pi_0 \cap [(ind(C1)\cap B2)\bar{\lambda}_x]\tau_0$ $\qquad$ $(B\bar{\lambda}_x \subseteq B\bar{\lambda}_{[x]\pi})$

$= [(B1\cap B2)\bar{\lambda}_x]\pi_0 \cap [(B1\cap B2)\bar{\lambda}_x]\tau_0$ $\qquad$ $(B1\cap B2 \subseteq ind(C1)\cap B2)$ $(B1\cap B2 \subseteq ind(C2)\cap B1)$ $((6), (7), (15))$

$= [(B1\cap B2)\lambda_x]\pi_0 \cap [(B1\cap B2)\lambda_x]\tau_0$ $\qquad$ $((15))$

$= (B1\cap B2)\lambda_x$ $\qquad$ $(\pi_0 \cdot \tau_0 = \pi_0(0))$

$= \phi(B1,B2)\lambda_x$ $\qquad$ $((13))$

From the above calculations and definitions 2.3, 3.13 and 3.14, it follows immediately that the serial connection of type PO of machines M_1 and M_2 realizes M, i.e. M has a serial full-decomposition of type PO. If condition (ii) of theorem 11.1 is satisfied, the decomposition is nontrivial. □

The interpretation of theorem 11.1 is similar to the interpretation of theorem 8.1.

12. Full-decompositions of state machines.

After modifying theorems 5.1, 6.1, 7.1 and 10.1, they can be applied to state machines.

A state machine is a special case of the sequential machine for which the output set O and the output function λ are not defined. If we take this into account and we define the full-decompositions of state machines in a manner analogous to the definitions for the general sequential machines and then, we remove from the listed theorems all the conditions concerning the output set O and the output function λ, we obtain the following theorems:

__THEOREM__ __12.1__ The state machine $M = (I, S, \delta)$ has a nontrivial parallel full-decomposition *if* such two partitions π_I and τ_I on I and such two partitions π_S and τ_S on S exist that the following conditions are satisfied:

(i) (π_S, π_S) is a S-S partition pair,
(ii) (π_I, π_S) is an I-S partition pair,
(iii) (τ_S, τ_S) is a S-S partition pair,
(iv) (τ_I, τ_S) is an I-S partition pair,
(v) $\pi_S \cdot \tau_S = \pi_S(0)$,
(vi) $|\pi_I|<|I| \wedge |\tau_I|<|I| \vee |\pi_S|<|S| \wedge |\tau_S|<|S|$.

__THEOREM__ __12.2__ The state machine $M = (I, S, \delta)$ has a nontrivial serial full decomposition of type PS if such two partitions π_I and τ_I on I and such two partitions π_S and τ_S on S exist that the following conditions are satisfied:

(i) (π_S, π_S) is a S-S partition pair,
(ii) (π_I, π_S) is an I-S partition pair,
(iii) (τ_I, τ_S) is an I-S partition pair,
(iv) $\pi_S \cdot \tau_S = \pi_S(0)$,
(v) $|\pi_I|<|I| \wedge |\pi_S| \cdot |\tau_I|<|I| \vee |\pi_S|<|S| \wedge |\tau_S|<|S|$.

__THEOREM__ __12.3__ The state machine $M = (I, S, \delta)$ has a nontrivial serial full-decomposition of type NS *if* such two partitions π_S and τ_S on S and such two partitions π_I and τ_I on I exist that the following conditions are satisfied:

(i) (π_S, π_S) is a S-S partition pair,
(ii) (π_I, π_S) is an I-S partition pair,
(iii) $\forall s,t \in S \ \forall x_1, x_2 \in I$:

$$\text{if } [s]\tau_S=[t]\tau_S \wedge [x_1]\tau_I=[x_2]\tau_I \wedge [s\delta_{x_1}]\pi_S=[t\delta_{x_2}]\pi_S$$

$$\text{then } [s\delta_{x_1}]\tau_S=[t\delta_{x_2}]\tau_S ,$$

(iv) $\pi_S \cdot \tau_S = \pi_S(0)$,
(v) $|\pi_I|<|I| \wedge |\pi_S| \cdot |\tau_S|<|I| \vee |\pi_S|<|S| \wedge |\tau_S|<|S|$.

THEOREM 12.4 The state machine $M = (I, S, \delta)$ has a nontrivial general full decomposition of type PS if and only if such two partitions π_I and τ_I on I and such two partitions π_S and τ_S on S exist that the following conditions are satisfied:

(i) (π_I, π_S) is an I-S partition pair,

(ii) (τ_I, τ_S) is an I-S partition pair,

(iii) $\pi_S \cdot \tau_S = \pi_S(0)$,

(iv) $|\tau_S| \cdot |\pi_I| < |I| \wedge |\pi_S| \cdot |\tau_I| < |I| \vee |\pi_S| < |S| \wedge |\tau_S| < |S|$.

Proof and interpretation of the theorems given above are analogous to those for theorems 5.1, 6.1, 7.1 and 10.1.

13. Conclusion.

The notions and theorems presented in the previous sections have straightforward practical interpretations. Based on them, a set of algorithms has been developed and a system of programs has been made for computing the different sorts of decompositions. We are going to present this algorithms and some practical conclusions in a separate report.

Here, we want only to stress three important facts:

Full-decompositions of type N are not so attractive from the practical point of view as decompositions of type P, because decompositions of type N introduce some timing problems. In decompositions of type N, one of the component machines has to compute its next state or output, before the second component machine, using the information about the computed next state or output, can compute its own next state or output. If we assume that computation of the next state and output for one component machine takes one time interval, a valid next state and output for the whole machine appears after two such time intervals. In this situation we have to limit the frequency of input signals and to use the two-phase clock.

Solving the practical tasks, we should first try to find a parallel full-decomposition which satisfies given requirements and only in the case of failure, we should look for a serial decomposition or, in the case of failure, for a general decomposition.This is so, because in the case of the serial and general decompositions, the connections between the partial machines have to be implemented and because the reduction of the functional dependences between input, state and output variables of the machine is decrising from the parallel through the serial to the general decomposition, i.e. the complexity of the combinational logic of each of the component machines is lowest for the parallel decomposition and highest for the general decomposition.

In some practical tasks, it is more economical to consider separately the realization of the next-state function δ and separately the realization of the output function λ than to consider them simultaneously. It is possible to abstract from the output function λ and to decompose first the state machine defined by the next-state function δ. It is passible to realize then the output function λ, where λ is treated as a function of inputs (in the Mealy case) and states of the partial state machines in a full-decomposition of the state machine defined by δ.

The results presented in this report are easy to extend in order to cover the case of incompletely specified sequential machines. It can be done by using the concepts of weak partition pairs or extended partition pairs introduced by Hartmanis [12].

REFERENCES

[1] M.A. Arbib : Theories of abstract automata, Englewood Cliffs,N.J.: Prentice - Hall, 1969.

[2] G. Cioffi, E. Constantini, S. de Julio : A new approach to the decomposition of sequential systems, Digital Processes, vol.3, p. 35-48, 1977.

[3] G. Cioffi, S. de Julio, M. Lucertini : Optimal decomposition of sequential machines via integer nonlinear programming: A computational algorithm, Digital Processes, vol.5, p. 27-41, 1979.

[4] A.D. Friedman, P.R. Menon : Theory and design of switching circuits, Woodland Hills, Cal.: Computer Science Press, 1975.

[5] A. Ginzburg : Algebraic theory of automata, N.Y.: Academic Press, 1968.

[6] J. Hartmanis : On the state assignment problem for sequential machines I, IRE Trans. Electron. Comput., vol.EC-10, p.157-165, 1961.

[7] J. Hartmanis, R.E. Stearns : On the state assignment problem for sequential machines II, IRE Trans. Electron. Comput., vol.EC-10, p.593-603, 1961.

[8] J. Hartmanis : Loop-free structure of sequential machines, Inf. & Control, vol.5, p.25-43, 1962.

[9] J. Hartmanis : Further results on the structure of sequential machines, J. Assoc. Comput. Mach., vol.10,p.78-88, 1963.

[10] J. Hartmanis, R.E. Stearns : Some danger in state reduction of sequential machines, Inf. & Control, vol.5, p.252-260, 1962.

[11] J. Hartmanis, R.E. Stearns : Pair algebra and its application to automata theory, Inf. & Control, vol.7, p.485-507, 1964.

[12] J. Hartmanis, R.E. Stearns : Algebraic structure theory of sequential machines, Englewood Cliffs, N.J.: Prentice-Hall, 1966.

[13] W.M.L. Holcombe : Algebraic Automata Theory, Cambridge University Press, 1982. (Cambridge studies in advanced mathematics, vol.1).

[14] Y. Hou : Trinity algebra and full-decompositions of sequential machines, Ph.D. thesis, Eindhoven University of Technology, The Netherlands, 1986.

[15] Y. Hou : Trinity algebra and its application to machine decompositions, Information Processing Letters, vol.26, p.127-134, 1987.

[16] Yu.V. Pottosin, E.A. Shestakov : Approximate algorithms for parallel decomposition of automata, Autom. Contr. & Comput. Sci., vol.15, No 2, p.24-31, 1981. (Translation of: Avtom. & Vytchisl. Techn.).

[17] Yu.V. Pottosin, E.A. Shestakov E.A. : Decomposition of an automaton into a two-component network with constraints on internal connections, Autom. Contr. & Comput. Sci., vol.16, No 6, p.24-31, 1982.

[18] Yu.V. Pottosin : Decompositional method for coding the states of a parallel automaton, Autom. Contr. & Comput. Sci., vol.21, No 1, p.78-84, 1987.

[19] M. Yoeli : The cascade decomposition of sequential machines, IRE Trans. Electron. Comput., vol.EC-10, p.587-592, 1961.

[20] M. Yoeli : Cascade-parallel decompositions of sequential machines, IEEE Trans. Electron. Comput., vol.EC-12, p.322-324, 1963.

Eindhoven University of Technology Research Reports
Faculty of Electrical Engineering

ISSN 0167-9708
Coden: TEUEDE

(168) Linnartz, J.P.M.G.
SPATIAL DISTRIBUTION OF TRAFFIC IN A CELLULAR MOBILE DATA NETWORK.
EUT Report 87-E-168. 1987. ISBN 90-6144-168-4

(169) Vinck, A.J. and Pineda de Gyvez, K.A. Post
IMPLEMENTATION AND EVALUATION OF A COMBINED TEST-ERROR CORRECTION PROCEDURE FOR MEMORIES WITH DEFECTS.
EUT Report 87-E-169. 1987. ISBN 90-6144-169-2

(170) Hou Yibin
DASM: A tool for decomposition and analysis of sequential machines.
EUT Report 87-E-170. 1987. ISBN 90-6144-170-6

(171) Monnee, P. and M.H.A.J. Herben
MULTIPLE-BEAM GROUNDSTATION REFLECTOR ANTENNA SYSTEM: A preliminary study.
EUT Report 87-E-171. 1987. ISBN 90-6144-171-4

(172) Bastiaans, M.J. and A.H.M. Akkermans
ERROR REDUCTION IN TWO-DIMENSIONAL PULSE-AREA MODULATION, WITH APPLICATION TO COMPUTER-GENERATED TRANSPARENCIES.
EUT Report 87-E-172. 1987. ISBN 90-6144-172-2

(173) Zhu Yu-Cai
ON A BOUND OF THE MODELLING ERRORS OF BLACK-BOX TRANSFER FUNCTION ESTIMATES.
EUT Report 87-E-173. 1987. ISBN 90-6144-173-0

(174) Berkelaar, M.R.C.M. and J.F.M. Theeuwen
TECHNOLOGY MAPPING FROM BOOLEAN EXPRESSIONS TO STANDARD CELLS.
EUT Report 87-E-174. 1987. ISBN 90-6144-174-9

(175) Janssen, P.H.M.
FURTHER RESULTS ON THE McMILLAN DEGREE AND THE KRONECKER INDICES OF ARMA MODELS.
EUT Report 87-E-175. 1987. ISBN 90-6144-175-7

(176) Janssen, P.H.M. and P. Stoica, T. Söderström, P. Eykhoff
MODEL STRUCTURE SELECTION FOR MULTIVARIABLE SYSTEMS BY CROSS-VALIDATION METHODS.
EUT Report 87-E-176. 1987. ISBN 90-6144-176-5

(177) Stefanov, B. and A. Veefkind, L. Zarkova
ARCS IN CESIUM SEEDED NOBLE GASES RESULTING FROM A MAGNETICALLY INDUCED ELECTRIC FIELD.
EUT Report 87-E-177. 1987. ISBN 90-6144-177-3

(178) Janssen, P.H.M. and P. Stoica
ON THE EXPECTATION OF THE PRODUCT OF FOUR MATRIX-VALUED GAUSSIAN RANDOM VARIABLES.
EUT Report 87-E-178. 1987. ISBN 90-6144-178-1

(179) Lieshout, G.J.P. van and L.P.P.P. van Ginneken
GM: A gate matrix layout generator.
EUT Report 87-E-179. 1987. ISBN 90-6144-179-X

(180) Ginneken, L.P.P.P. van
GRIDLESS ROUTING FOR GENERALIZED CELL ASSEMBLIES: Report and user manual.
EUT Report 87-E-180. 1987. ISBN 90-6144-180-3

(181) Bollen, M.H.J. and P.T.M. Vaessen
FREQUENCY SPECTRA FOR ADMITTANCE AND VOLTAGE TRANSFERS MEASURED ON A THREE-PHASE POWER TRANSFORMER.
EUT Report 87-E-181. 1987. ISBN 90-6144-181-1

(182) Zhu Yu-Cai
BLACK-BOX IDENTIFICATION OF MIMO TRANSFER FUNCTIONS: Asymptotic properties of prediction error models.
EUT Report 87-E-182. 1987. ISBN 90-6144-182-X

(183) Zhu Yu-Cai
ON THE BOUNDS OF THE MODELLING ERRORS OF BLACK-BOX MIMO TRANSFER FUNCTION ESTIMATES.
EUT Report 87-E-183. 1987. ISBN 90-6144-183-8

(184) Kadete, H.
ENHANCEMENT OF HEAT TRANSFER BY CORONA WIND.
EUT Report 87-E-184. 1987. ISBN 90-6144-184-6

(185) Hermans, P.A.M. and A.M.J. Kwaks, I.V. Bruza, J. Dijk
THE IMPACT OF TELECOMMUNICATION ON RURAL AREAS IN DEVELOPING COUNTRIES.
EUT Report 87-E-185. 1987. ISBN 90-6144-185-4

(186) Fu Yanhong
THE INFLUENCE OF CONTACT SURFACE MICROSTRUCTURE ON VACUUM ARC STABILITY AND ARC VOLTAGE.
EUT Report 87-E-186. 1987. ISBN 90-6144-186-2

(187) Kaiser, F. and L. Stok, R. van den Born
DESIGN AND IMPLEMENTATION OF A MODULE LIBRARY TO SUPPORT THE STRUCTURAL SYNTHESIS.
EUT Report 87-E-187. 1987. ISBN 90-6144-187-0

(188) Jóźwiak, L.
THE FULL DECOMPOSITION OF SEQUENTIAL MACHINES WITH THE STATE AND OUTPUT BEHAVIOUR REALIZATION.
EUT Report 88-E-188. 1988. ISBN 90-6144-188-9