ASYNCHRONOUS DYNAMICAL SYSTEMS PART II: NON-DETERMINISM AND REALIZATION

by

Kemal Inan

Memorandum No. UCB/ERL M89/77

23 June 1989

COVER PAGE

ASYNCHRONOUS DYNAMICAL SYSTEMS
PART II: NON-DETERMINISM AND REALIZATION

by

Kemal Inan

Memorandum No. UCB/ERL M89/77

23 June 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

ASYNCHRONOUS DYNAMICAL SYSTEMS PART II: NON-DETERMINISM AND REALIZATION

by

Kemal Inan

Memorandum No. UCB/ERL M89/77

23 June 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ASYNCHRONOUS DYNAMICAL SYSTEMS PART II : NON-DETERMINISM AND REALIZATION [1]

Kemal Inan

Department of Electrical Engineering and Computer Sciences
and Electronics Research Laboratory
University of California, Berkeley, CA 94720

ABSTRACT

The shortcoming of the deterministic ADS model is illustrated by examples of 'deadlock' and 'livelock' that is ignored by deterministic semantics . The examples motivate the generalization of the deterministic ADS theory to nondeterministic environments. This is accomplished first by defining non-deterministic images of deterministic marked process spaces and then using an extended signal theory to define nondeterministic ADS and its environment .

The realization problem for ADS is defined and solved for the deterministic case and generalized to the nondeterministic case . The relation between buffer-free representation through the response function and buffer dependent representation by the state function of the ADS raises issues peculiar to nondeterminism such as computing realizations that preserve deterministic behaviour. Different equivalence definitions including *bisimulation* [5] are presented in the context of ADS and the expressive power of the underlying semantics are compared . The distinction between weak and strong equivalence is shown to be related to scenario analysis for data flow networks [7] .

[1] Research supported in part by NSF Grant ECS 8719779, by the MICRO program of the State of California and a grant from Pacific Bell .

1. Introduction

This is the second part of a three part paper on asynchronous dynamical systems. In this paper two problems are addressed : first the ADS theory presented in part I is generalized to incorporate nondeterministic signals and second , the problem of realization of response functions by ADS is formulated and solved first for the deterministic case and then extended to the nondeterministic case. Based on these results various definitions of ADS equivalence are presented and related to each other.

In the context of dynamical systems the term 'nondeterminism' points to the existence of externally observed sequence of event transitions that does not correspond to a unique set of internal transitions of the system. The underlying system is typically a state machine or a *synchronization tree* [5] which is a special kind of state machine , possibly infinite and acyclic so that it has no prior commitment to a specific definition of equivalence among its states. State transitions are labeled by events and a special symbol denotes the invisible event for modeling unobservable transitions . In such a machine a unique sequence of visible events may drive the system to a set of different states . This may , for example , happen if from a given state there are different transitions with identical event labels or there are unobservable transitions . Under such conditions the system is said to behave in a nondeterministic way and models are suggested that capture relevant features of such behaviour .

A *nondeterministic semantics* is a formalism that differentiates nondeterministic behaviour by directly or indirectly imposing an equivalence relation on the set of all state machines described above . The expressive power of a nondeterministic semantics is measured by the level of refinement induced by its equivalence classes . For example two machines are defined to be *trace equivalent* if they accept the same sequences of visible transitions . This is one of the coarser equivalence definitions that is used for deterministic modeling .Milner's *observational equivalence* and its related concept of *bisimulation* [5] or Hoare's definition of equivalence through *failures* and *divergences* [4] are alternative definitions of nondeterministic equivalence. In the context of computer languages , different definitions of equivalence correspond to different expressive powers of the corresponding languages that support nondeterminism [6] .

In this paper we model nondeterministic behaviour using a modified form of Hoare's approach to nondeterminism . Our perspective is slightly different from that of a language designer . We implicitly assume that at some detailed level of modeling ADS environment behaves deterministically. As we simplify interconnected ADS by the appropriate use of the projection operator we introduce nondeterministic behaviour to the reduced models. We would like the model of nondeterminism to be expressive enough such that the reduced models are simpler , yet they incorporate the necessary information that allows us to predict the worst case behaviour of the original deterministic model. Therefore , for example , we allow a process to recover from diverging conditions instead of imposing CHAOS after divergence as Hoare does. Doing so we retain greater amount of information about the behaviour of the original system at a negligible cost of representation complexity . The definition and concepts of nondeterministic marked processes and extension of ADS to nondeterministic environments are all treated in section 2 .

In section 4 we introduce the *asynchronous dynamical realization problem* . Two fundamental results are derived that state the conditions for a response function to be realized by ADS . The first result states the necessary and sufficient conditions for a deterministic response function to be realized by a deterministic ADS and constructs an ADS in the proof . The second result again states the necessary and sufficient conditions , under mild hypotheses , for a nondeterministic response function to be realized by nondeterministic ADS that *behaves deterministically* on *trajectory* inputs , that is , inputs with a single trace and its prefixes . These results characterize response functions as *buffer-free* representations of asynchronous dynamics and problems like optimal use of buffering for maximal parallelism or deterministic behaviour of a synthesis become practically relevant questions .

The distinction of strong and weak equivalence of section 3 is related to the *scenario analysis* of Dennis used for data flow models [8] . In the context of data flow models scenario analysis is an approach that allows a unique characterization of interconnected actors when input-output description of each actor is not sufficient to characterize the interconnected behaviour uniquely. This situation has a natural formulation in ADS environments where the input-output histories correspond to response functions and scenario analysis coincides with fitting a specific realization to such response functions. The ADS approach looks richer both in its signal variety and nondeterministic representation power . A detailed treatment of this issue with concrete applications to input-output computations will be presented in part III of this sequence .

Our operational perspective of nondeterministic modeling culminates in the concepts of strong and weak *nondeterministic equivalence* given in section 3 , which stand for equivalence relations defined on the space of original , non-reduced , deterministic ADS representations . The refinement corresponding to strong nondeterministic equivalence corresponds to the differentiating power of the nondeterministic ADS model used in this paper . We distinguish the semantics of our definition by relating it to the concept of *bisimulation* [5] , a dynamically invariant definition of equivalence in the context of ADS environment , and show that any *bisimulation* refines our model of nondeterminism . Finally the results of the paper are discussed in section 4 by pointing to some open problems .

2. Nondeterministic ADS Environment

In this section we generalize the results of the part I of this paper to incorporate nondeterministic phenomena . This generalization , founded upon an operationally simple and efficient representation of nondeterministic signals leads to an algebra of ADS block diagram manipulation techniques as to be demonstrated in part III of this sequence of papers.

The deterministic definition of an ADS given in part I of this paper disregards problems of nondeterminism inherent in ADS dynamics. This point is illustrated by the examples below . For these examples we make the simplifying assumption that all the input and state signals in U and X behave deterministically in a representation $R = (U,X,Y)$. By this it is meant that for any such signal v :

$s\hat{}<a> \in trv \Rightarrow$ the process v/s cannot remain idle indefinitely .

This is the implicit assumption of determinism which is an externally imposed semantics to the formal definition of deterministic processes . Now let $s\hat{}<a> \in trH(u)$ then two types of phenomenon may violate this deterministic assumption :

(1) There may exist a trace t of $S(u)$ such that $t\downarrow_Y = s \in trH(u)$ and $tr(S(u)/t) = \{< >\}$. This implies that the state signal $S(u)$, if trapped at t cannot progress any further and $H(u)/s$ cannot progress also.

(2) There may exist an infinite set Γ of traces of $S(u)$ such that $t \in \Gamma \Rightarrow t\downarrow_U = s$. This implies that the state signal $S(u)$ makes an arbitrary number of transitions that are invisible at the output $H(u)$. Again $H(u)/s$ cannot may not be able to make any progress at all .

Consider the example given in Figure 2.1 . If an input u is applied with $tru := \{< >,<a>\}$ then the state signal $S(u)$ may generate an arbitrary number of c's invisible at the output (case (2) above) ; as it generates b's at the output it may end up at state 2 and the output cannot develop any further although $trH(u) = b^*$ (case (1) above) . We shall loosely use the words 'deadlock' and 'livelock' to refer to situations illustrated in cases 1 and 2 respectively .

In simplifying an interconnected ADS one uses repeated layers of state projections eventually to compute a simple input-output relation. At each reduction step further nondeterministic behaviour is introduced accumulatively of the type mentioned above. If an interconnected system reduction method is to accomplish more than an input-output 'logical correctness' check , that is , if it is to detect 'deadlock' or 'livelock' situations also , then it is essential that the accumulative effects of the kind of phenomena described above is captured by a nondeterministic model that is simple and computationally effective. This is achieved in two steps : first a model is developed for representing nondeterministic marked processes ; second this model is used for re-defining the ADS operators to handle nondeterministic signals such that the construction of a logical nondeterministic ADS environment is complete .

We shall define a *nondeterministic process space* in the image of a given deterministic process space and use this definition to model the nondeterministic signals in ADS. For reasons of consistency we start at the embedding space level [2] .

Let $W = W(A, M, \Phi)$ be a given (deterministic) embedding space with the usual partial order and length projection operators '$\leq$' and '$\uparrow n$' . We define the **nondeterministic image** of W as the embedding space [3] given by the triple ($\overline{W}$, $\lesssim$, $\hat{\uparrow} n$) where the embedding set $\overline{W} := \overline{W}(A, \overline{M}, \Psi)$ is given by the following definitions of $\overline{M}$ and Ψ

$$\overline{M} := 2^{2^A} \times \{0,1\} \times M$$

and Ψ is the set of partial functions μ defined on prefix-closed subsets of A^* with the general structure

$$\mu = (\mu_r, \mu_d, \mu_m) : A^* \rightarrow 2^{2^A} \times \{0,1\} \times M$$

where $\mu_m \in \Phi$ and for any allowable $s \in A^*$

$$(B \in \mu_r(s) \, ; C \subseteq B \,) \Rightarrow C \in \mu_r(s) \tag{2.1}$$

Following Hoare [4] we call $\mu_r(s)$ the **refusals** of the process at the trace s and a member of the refusals a **refusal set** or simply a refusal . A refusal B is called a **maximal refusal** if $B \subseteq D$ and $D \in \mu_1(s)$ implies that $B = D$. The interpretation of a refusal B is simple : if after the execution of the trace s , the environment offers the process the elements of B as the only possible next transitions , the process *may* stop without any further progress . The mark associated with μ_d denotes infinite hidden traces of a process named , again after Hoare , a **divergence** . If at a trace s the mark $\mu_d(s) = 1$ then this signifies the possibility that the process may engage in an infinite activity of hidden transitions after executing the trace s . Our definition , however , differs from that of Hoare's , since we allow for recovery after divergence whereas Hoare assumes a chaotic state of affairs - that is , every trace and every refusal is possible after a divergence - after the process diverges. The difference arises from our perspective of efficiency . If nondeterminism is viewed as a byproduct of system reduction either by hiding events or relabeling them , then efficiency in nondeterministic representation means preserving the maximum amount of information on the original system's behaviour with minimum representation complexity for the reduced system .

In order to keep the notation simple we shall use the same symbols '$\leq$' and '$\uparrow n$' instead of $\lesssim$ and $\hat{\uparrow} n$ for the new nondeterministic partial order and length projection operators. The context should fix the meaning. The new partial order '$\leq$' for the embedding space is defined by letting $v \leq w$ if and only if

[2] For a definition of embedding space see part I of this paper , for a detailed treatment see [3] .

[3] This is a much simpler version of the nondeterministic embedding space suggested in [3] . The reason for the simplification is due to the special structure of the ADS environment .

(1) $trv \subseteq trw$,

(2) $\mu_m v(s) = \mu_m w(s)$ for all $s \in trv$,

(3) $\mu_r w(s) \subseteq \mu_r v(s)$ for all $s \in trv$.

(4) $\mu_d w(s) \le \mu_d v(s)$ for all $s \in trv$.

According to this order the larger of two processes with identical traces is more deterministic , that is to say more predictable , because of requirements (3) and (4) .

The length projection operator '$\uparrow n$' is defined by

$$tr(v \uparrow n) := (s \in trv \mid \#s \le n)$$
$$\mu_m(v \uparrow n)(s) := \mu_m v(s)$$
$$\mu_d(v \uparrow n)(s) := \mu_d v(s)$$

and

$$\mu_r(v \uparrow n)(s) := \begin{cases} \mu_r v(s) & \text{if } \#s < n \\ 2^A & \text{if } \#s = n \end{cases}$$

Fact 2.1

The embedding set $\overline{W}$ together with the partial order and length projection operators constitutes an **embedding space** [4] . In particular if v_j is a chain in $\overline{W}$ then it converges to v where

$$trv := \bigcup_j trv_j$$
$$\mu_r v(s) := \bigcap_{j \ge J} \mu_r v_j(s) \tag{2.2a}$$
$$\mu_m v(s) := \mu_m v_J(s)$$

where J is any integer that satisfies : $j \ge J \Rightarrow s \in trv_j$; finally

$$\mu_d v(s) := \begin{cases} 0 \text{ , if for some } j \text{ , } \mu_d v_j(s) = 0 \\ 1 \text{ , otherwise} \end{cases} \tag{2.2b}$$

We first observe that deterministic behaviour is a special case of the nondeterminism defined above . More precisely a process $w \in \overline{W}$ is said to **behave deterministically** if for each $s \in trw$

$$\mu_r w(s) = 2^{\overline{B}_s}$$
$$\mu_d w(s) = 0$$

where $\overline{B}_s$ is defined by

$$a \notin \overline{B}_s \iff s \,\hat{}\, \langle a \rangle \in trw \tag{2.3}$$

After defining a nondeterministic embedding space in the image of a given deterministic embedding space we establish a correspondence between deterministic and nondeterministic processes in a simple way. For this we use the symbols 'η' and 'δ' as operators that convert a deterministic process into a nondeterministic one and vice-versa . More precisely let $w \in W$ then we define $\eta(w) \in \overline{W}$ as

[4] The fundamental property of an embedding space is that the partial order is complete and the basic calculus operators , namely the choice function , post-process function and the length projection function are all *continuous* with respect to the partial order defined. The reader is referred to [2] for the exact definition of an embedding space .

follows :

$$tr\,\eta(w) := trw$$
$$\mu_m\eta(w)(s) := \mu w(s)$$
$$\mu_r\eta(w)(s) := \overline{B}_s$$
$$\mu_d\eta(w)(s) := 0$$

where $\overline{B}_s$ is given by (2.3) . Note that $\eta(w)$ is the *largest* nondeterministic process in $\overline{W}$ that agrees in traces and the deterministic mark with the deterministic process w . In particular it behaves deterministically .

In a similar way let $w \in \overline{W}$ and define $\delta(w) \in W$ as the process obtained from w by deleting the μ_r and μ_d components of its marking function and setting $\mu\delta(w)(s) := \mu_m w(s)$.

The following relation is immediate :

$$\begin{aligned} &\delta(\eta(w)) = w \\ &v \le \eta(\delta(v)) \end{aligned} \tag{2.4}$$

for all $w \in W$ and $v \in \overline{W}$. In particular η is a *one-to-one* mapping of W into $\overline{W}$ whereas δ maps $\overline{W}$ *onto* W . We define below a marked process space for the nondeterministic case where there are two universal nondeterminism axioms that must be satisfied in addition to any arbitrary set of marking axioms.

Definition 2.1

Any subset Π of $\overline{W}$ is called a **nondeterministic marked process space** provided that it satisfies the axioms of a marked process space (see [1]) and the following general marking axioms of nondeterminism. For all $P \in \Pi$ and $s \in trP$:

(1) If $<a> \notin trP/s$ then $<a> \in B$ for all *maximal refusals* B in $\mu_r P(s)$,

(2) If $\mu_d P(s) = 1$ then $\mu_r P(s) = 2^A$.

Fact 2.2

The function δ maps marked process spaces in $\overline{W}$ into marked process spaces in W . Conversely if Π is a marked process space in W then the set of processes given by

$$\overline{\Pi} := (Q \in \overline{W} \mid \text{ there exist } P \in \Pi \text{ such that } Q \le \eta(P)\,) \tag{2.5}$$

constitutes the *largest* marked process space in $\overline{W}$ (with respect to set inclusion) that satisfies the relation

$$\delta(\overline{\Pi}) = \Pi$$

Before we generalize the domain of the ADS operators to cover nondeterministic arguments we define the *sum* of nondeterministic spaces below. Let $\overline{W},\overline{V}$ be the nondeterministic images of the embedding spaces W and V respectively . We define

$$(\overline{W} \oplus \overline{V}) := \overline{(W \oplus V)}$$

Similarly let Π and Θ be process spaces in W and V respectively then we define the sum space of $\overline{\Pi}$ and $\overline{\Theta}$ as :

$$\overline{\Pi} \oplus \overline{\Theta} := \overline{(\Pi \oplus \Theta)}$$

Recall that $[w]$ in the deterministic case is a notation used for the set of all subprocesses of w [1] . For the nondeterministic case it will acquire a slightly different meaning . If $w \in \overline{W}$ where $\overline{W}$ is the nondeterministic image of W we define the set $[w]$ as a subset of subprocesses of w such that if v is in this subset then each maximal refusal of v at a trace s includes some maximal refusal of w at s . In other words if D is a maximal refusal of v at s then $D = E \cup F$ where E is some *maximal* refusal of w at s and F is a subset of A . Therefore , for example , any subprocess of w that possesses a maximal refusal that does not contain some maximal refusal of w at some given trace is not in $[w]$. Intuitively $[w]$ stands for all subprocesses of w in which the nondeterminism is cumulative with respect to w . As an example take $trw \ \{<>,<a>,<b>,<c>\}$, where $\mu_r w(<>) = 2^{\{a\}}$ and $trv = \{<>,<a>,<b>\}$. If $\mu_r v(<>) = 2^{\{a,b,c\}}$, then $v \in [w]$ whereas if $\mu_r v(<>) = 2^{\{b,c\}}$ then v is not in $[w]$ since a is not in any maximal refusal of v.

We shall extend the definitions of the three ADS operators , namely sum '$\oplus$' , internal sum '+' and projection '.' , to incorporate nondeterministic arguments in their domains and produce nondeterministic signals in their ranges. We shall use the same symbols for these extended operators so as not to overcrowd the symbolism . Since nondeterminism covers the deterministic case no meaning is lost in this symbol superimposition. Also throughout we assume that all the nondeterministic process spaces are nondeterministic images of given deterministic spaces in accordance with (2.5) above .

The following definition extends the sum operator and its restricted version the internal sum operator , to nondeterministic process spaces.

Definition 2.2

Let $x \in \overline{\Pi}$ and $y \in \overline{\Theta}$ then we define the **nondeterministic sum operator** as follows

$$tr(x \oplus y) := tr(\delta(x) \oplus \delta(y))$$
$$\mu_m(x \oplus y)(s) := \mu(\delta(x) \oplus \delta(y))(s)$$
$$\mu_d(x \oplus y)(s) := \max(\mu_d x(s\downarrow_x), \mu_d y(s\downarrow_y))$$

and $\mu_r(x \oplus y)(s)$ is the smallest subset of 2^A that satisfies :

(1) $2^B \subseteq \mu_r(x \oplus y)(s)$ if $B = C \cap D$ where $C \in \mu_r x(s\downarrow_x)$ and $D \in \mu_r y(s\downarrow_y)$.

(2) If $\mu_d(x \oplus y)(s) = 1$ then $\mu_r(x \oplus y)(s) = 2^A$.

We define the **nondeterministic internal sum operator** of x and y with respect to the cover sets $[X]$ and $[Y]$ where X, Y are processes in $\overline{\Pi}$ and $\overline{\Theta}$ respectively with $x \in [X]$ and $y \in [Y]$ by

$$tr(x + y) := tr(\delta(x) + \delta(y))$$
$$\mu_m(x + y)(s) := \mu(\delta(x) + \delta(y))(s)$$
$$\mu_d(x+y)(s) := \max(\mu_d x(s\downarrow_X), \mu_d y(s\downarrow_Y))$$

and again $\mu_r(x + y)(s)$ is the smallest subset of 2^A such that :

(1) For each pair $C \in \mu_r x(s\downarrow_X)$ and $D \in \mu_r y(s\downarrow_Y)$ define B as

$$B := (C \cap D) \cup (C \cap B_X) \cup (D \cap B_Y) \qquad (2.6)$$

where B_X is the set of next transitions of $X/(s\downarrow_X)$ and B_Y is defined similarly . For each such B $2^B \subseteq \mu_r(x + y)(s)$.

(2) If $\mu_d(x + y)(s) = 1$ then $\mu_r(x + y)(s) = 2^A$.

First , observe that $(x + y) \in [X \oplus Y]$ since by the first definition above every maximal refusal of $X \oplus Y$ at s is of the form $C \cap D$ with C and D maximal refusals of $X/(s \downarrow_X)$ and $Y/(s \downarrow_Y)$ and therefore by (2.6) above maximal refusals of $x + y$ are increments to those of $X \oplus Y$ as required by the bracket notation. Second , observe that both definitions satisfy the associativity condition where for the internal sum operator it can be shown by induction that for the process $x_1 + \cdots + x_n$ with cover sets $[X_j]$, B in (2.6) can be replaced by :

$$B = (C_1 \cap \cdots \cap C_n) \cup (C_1 \cap B_1) \cup \cdots \cup (C_n \cap B_n)$$

where each C_j is a refusal of x_j at $s \downarrow_{X_j}$ and B_j is the set of next transitions of $X_j/(s \downarrow_{X_j})$.

The following definition extends the projection operator to nondeterministic range and domain.

Definition 2.3

$P \in \overline{\Pi}$ and $Q \in \overline{\Theta}$ where Π and Θ are deterministic process spaces . We define $P.Q \in \overline{\Theta}$ as follows :

$$tr(P.Q) := tr(\delta(P).\delta(Q))$$
$$\mu_m(P.Q)(s) := \mu(\delta(P).\delta(Q))$$

and if T_s is defined as

$$T_s := (t \in trP \mid t \downarrow_Q = s)$$

The divergence at s , $\mu_d(P.Q)(s)$ is set to 1 iff one or more of the following conditions hold :

(1) For some $q \in T_s$ the set

$$T := (t \in tr(P/q) \mid t \downarrow_{Q/s} = < >)$$

is *infinite* ;

(2) For some $t \in T_s$, $\mu_d P(t) = 1$;

(3) $\mu_d Q(s) = 1$.

The refusals at s , $\mu_r(P.Q)(s)$, is defined as the smallest subset of 2^A that satisfies the following requirements :

(1) If $t \in T_s$ and $B \in \mu_r P(t)$ has the property that :

$$t \,\hat{}\, <a> \in T_s \Rightarrow a \in B$$

and $C \in \mu_r Q(s)$ then $2^{(B \cup C)} \subseteq \mu_r(P.Q)(s)$.

(2) If $\mu_d(P.Q)(s) = 1$ then $\mu_r(P.Q)(s) = 2^A$.

The definition of the nondeterministic projection is important in the sense that it is here that the new refusals and divergences are generated. According to the conditions that set the divergence index to 1 the process diverges if either the process P makes inifinite transitions transparent to Q after generating the trace s ; (see also Lemma 2.1 below) or along a trace of P that moves on the inverse image of s the process P diverges ; or Q diverges at s. In most applications the process Q behaves deterministically therefore the first two conditions are the typical sources of divergence.

One peculiarity of nondeterministic projection is the multiplicative effect of refusals when a process is consecutively projected onto itself. Consider a process P such that $trP = \{< >, <a_1>, \cdots , <a_n>\}$ and the maximal refusals are $\{a_j\}$ for $j = 1, \ldots , n$. It follows from the definition above that $P.P$ has the same traces as P but the maximal refusals are $\{a_j , a_i\}$ for all distinct pairs a_j and a_i . Similar arguments apply to $P.P.P$ where pairs are replaced by triples and finally $P. \cdots .P$ where projection is repeated n times the whole set of events is the maximal refusal.

The condition that the set T defined above is infinite is equivalent to the existence of an infinite trajectory that is transparent to Q as stated precisely by the following lemma .

Lemma 2.1

The set T in Definition 2.3 is *infinite* if and only if there exists and infinite sequence of traces $\{t_j\}$ such that for each j

(i) $t_j \in T$,

(ii) $\#t_j \geq j$,

(iii) For all i,j the traces t_j and t_i agree on the first $min(i,j)$ entries .

The hidden infinite trajectory τ is defined as the limit of the trajectories τ_j defined by their maximal traces given by $t_j \uparrow j$.

The condition that generates the diverging infinite trajectory is more delicate then it may look. To appreciate this consider the processes P and Q where $trP := :\{a^n c b^n ; n \geq 0\}$ [5] and $trQ = \{c^*\}$. Clearly $\mu_d(P.Q)(< >) = 1$ since P can generate arbitrarily many a's hidden to Q. Yet $\mu_d(P.Q)(c) = 0$ although there are infinitely many traces of P that generate c for $P.Q$. The reason why the process does not diverge is that it cannot make infinite transitions that are hidden to Q after generating c . This is captured by the requirement that a trace q should exist such that $q \downarrow_Q = c$ and infinite hidden traces are in P/q. For the example above , once a q is given , a finite integer n is fixed and only b^n can be generated after producing the event c. Further consequences of this observation follow in the restatement and the proof of the nondeterministic version of formula (3.13) in [1] (see Fact 2.4 item (3) below).

The first condition for the refusals in Definition 2.3 is subtle and requires justification. The justification is supplied by Lemma 2.2 below

Lemma 2.2

The refusals given by the Definition 2.3 satisfies the general axioms of nondeterminism .

Proof of Lemma 2.2

We have to show that if $<b> \notin tr^{-}(P.Q)/s$ then it belongs to all the maximal refusal sets in $\mu_r(P.Q)(s)$. There are two cases.

In the first case there exists $t \in T_s$ such that there does not exist any $a \in A$ and $t\hat{}<a> \in T_s$. Then by condition (1) any refusal set B of $P(t)$ - in particular the empty set - can be 'unioned' with any refusal set F of $Q(s)$ to generate a refusal $B \cup F$ of $(P.Q)(s)$. On the other hand if $<b> \notin tr((P.Q)/s)$ then b must either be a member of some maximal F of Q at s ; or $<b>$ belongs to trQ/s but it does not get perturbed by P , i.e., b belongs to every maximal refusal of $P(t')$ for $t' \in T_s$, hence it belongs to the maximal refusal B of $P(t)$. In both cases $b \in B \cup F$ and we are done .

In the second case starting from any $t \in T_s$ we can paste invisible transitions of P to it indefinitely. But this implies that $\mu_d(P.Q)(s) = 1$ by condition (1) of divergences which further implies that $\mu_r(P.Q)(s) = 2^A$.

[5] Note that P cannot be generated by a finite state machine .

The following result relates the deterministic and nondeterministic ADS operators through the δ and η maps. Identical notation is used for both the deterministic and nondeterministic ADS operators and the distinction follows from the context.

Fact 2.3

(1) Let v, w be nondeterministic processes in nondeterministic spaces then

$$\delta(v \oplus w) = \delta v \oplus \delta w$$
$$\delta(v + w) = \delta v + \delta w$$
$$\delta(v.w) = \delta v.\delta w$$

(2) Let v, w be deterministic processes in deterministic spaces then

$$\eta(v \oplus w) = \eta v \oplus \eta w$$
$$\eta(v + w) = \eta v + \eta w$$

and

$$\eta(v.w) \geq \eta v.\eta w$$

It turns out that the important results derived for the deterministic versions of the ADS operators are either valid for the nondeterministic case or can be made so by slight modifications . One operation to be clarified in this re-interpretation of formulas under nondeterminism is the union of signals for the nondeterministic case since it is used both for the post-process and choice operations under projections (see Fact 2.2 , (5) and (6) in [1]). If v_t is a set of nondeterministic signals indexed by t in a nondeterministic signal space $[V]$ we define $\bigcup_t v_t$ as

$$\begin{aligned} tr(\bigcup_t v_t) &:= \bigcup_t trv_t \\ \mu_m(\bigcup_t v_t)(s) &:= \mu v_{t'}(s) \\ \mu_r(\bigcup_t v_t)(s) &:= \bigcup_{t \in T'} \mu_r v_t(s) \end{aligned} \qquad (2.7a)$$

where

$$T' := (t \mid s \in trv_t)$$

and $t' \in T'$; finally

$$\mu_d(\bigcup_t v_t)(s) := 1 \text{ iff for some } t \text{ , } \mu_d v_t(s) = 1 \qquad (2.7b)$$

Observe that unlike the deterministic case the union is not necessarily larger than its components in the nondeterministic partial order. This is because if two indices t and t' correspond to two signals sharing a trace s then at s the divergences and refusals are combined to generate a less predictable process .

Finally we extend the definition of a **solid process space** to the nondeterministic case .

A nondeterministic process space Π is said to be **solid** if for each P in Π , any process that inherits a prefix closed subset of the traces of P and legitimate subsets - i.e. those subsets that satisfy the universal nondeterministic marking axiom of Definition 2.1 - of its refusals at each trace s , is also in Π . Observe that if Π is a solid deterministic space then $\overline{\Pi}$ defined by (2.5) is also a solid nondeterministic process space. Henceforth we assume all the nondeterministic process spaces to be solid [6] . The basic novelty of nondeterminism is the possibility of differentiating signals in $[V]$ not only by their

[6] For a given nondeterministic process V in a solid marked space the subset $[V]$ is *closed* but not necessarily *solid* since there are subprocesses of V that are not in $[V]$.

traces but also by their divergences and refusals on identical traces which measure the behavioral predictability of the signal at that trace.

Unlike the deterministic case projection operator is not continuous with respect to its first argument . We give counter-examples below.

For the counter-examples considered we take the event alphabet as $A := \{a, b\}$. First take P^n to be a deterministically behaving chain where $trP^n := a^n$ and let the deterministically behaving process Q be defined by $trQ := b^*$. Then for each finite n we have $\mu_d(P^n.Q)(< >) = 0$ since P has no infinite trajectories . However the chain P^n converges to P and in the limit the projection diverges , that is $\mu_d(P.Q)(< >) = 1$ since P can generate infinitely many a's concealed from Q .

A similar situation arises also in refusals. This time define recursively the traces of the elements of deterministically behaving chain P^n as[7] :

$$trP_n := \begin{cases} \{a^n\} \bigcup trP_{n-1} & , n \quad \text{even} \\ \{a^n b\} \bigcup \{a^n\} \bigcup trP_{n-1} & , n \quad \text{odd} \end{cases}$$

where $trP_0 := \{< >\}$ and take Q as in the previous example . Then whenever n is even , P^n can make n hidden a transitions and refuse everything after that. Therefore according to the definition of refusals we have for n even

$$\mu_r(P^n.Q)(<>) = 2^A$$

On the other hand if n is odd P^n cannot refuse b at any stage since at each stage there is an invisible path of a's to generate the transition b. Therefore for n odd we have

$$\mu_r(P^n.Q)(< >) = 2^{\{a\}}$$

and the sequence of refusals oscillate between these two values where convergence is not possible.

Properties of extended ADS operators are summarized by Fact 2.4 below.

Fact 2.4

The extended projection operator is not necessarily continuous in any of its arguments but the nondeterministic versions of the sum and the internal sum operators are *associative* and *ndes* functions of their arguments. The internal sum operator is also *continuous* in its arguments . The following properties are enjoyed by the extended operators where all processes are assumed to be members of nondeterministic process spaces .

(1) For any w and v

$$\mu_d v(s) \leq \mu_d(w.v)(s)$$
$$\mu_r v(s) \subseteq \mu_r(w.v)(s)$$

and formulas in [1] , namely (3.3)-(3.6) extend to the nondeterministic case. On the other hand the idempotency result (Fact 3.2 item (4) in [1]) is replaced by the corresponding inequality

$$P.(P.Q) \leq P.Q$$

for the nondeterministic case.

[7] It can easily be checked that this recursive definition of traces is prefix-closed .

(2) For the post-process and choice function formulas the relations

$$\begin{aligned} \mu_d((P.\hat{P})/s)(t) &= \mu_d\{ \bigcup_{m\downarrow_{\hat{p}}=s} (P/m.\hat{P}/s)\}(t) \\ \mu_r((P.\hat{P})/s)(t) &= \mu_r\{ \bigcup_{m\downarrow_{\hat{p}}=s} (P/m.\hat{P}/s)\}(t) \end{aligned} \tag{2.8}$$

hold and therefore the post-process formula in (3.7) in [1] extends to the nondeterministic case. Similarly (3.9) in [1] also extends to the nondeterministic case .

(3) The divergences and refusals satisfy the following relations :

$$\begin{aligned} \mu_d((w \oplus v)/s)(t) &= \mu_d(w/(s\downarrow_w) \oplus v/(s\downarrow_v))(t) \\ \mu_r((w \oplus v)/s)(t) &= \mu_r(w/(s\downarrow_w) \oplus v/(s\downarrow_v))(t) \end{aligned} \tag{2.9}$$

$$\begin{aligned} \mu_d w(s) &\leq \mu_d((w\oplus v).w)(s) \\ \mu_r w(s) &\subseteq \mu_r((w\oplus v).w)(s) \end{aligned} \tag{2.10}$$

$$\begin{aligned} \mu_d(u.w)(s) &\leq \mu_d(u.(w\oplus v).w)(s) \\ \mu_r(u.w)(s) &\subseteq \mu_r(u.(w\oplus v).w)(s) \end{aligned} \tag{2.11}$$

and if the following conditions are satisfied :

(C1) w and v behave deterministically ,

(C2) The process u is generated by a *finite state machine* ,

then the inequalities in (2.11) can be replaced by equalities .

These results imply that (3.12) and (3.13) in [1] are replaced for the nondeterministic case by

$$(w\oplus v).w \leq w \tag{2.12}$$

and

$$u.(w\oplus v).w \leq u.w \tag{2.13}$$

and the inequality may be replaced by an equality in (2.13) if conditions (C1) and (C2) hold .

(4) For the internal sum operation the relations

$$\begin{aligned} \mu_d((w + v)/s)(t) &= \mu_d((w/(s\downarrow_W) + v/(s\downarrow_V))(t) \\ \mu_r((w + v)/s)(t) &= \mu_r((w.L(s\downarrow_W) + v/(s\downarrow_V))(t) \end{aligned} \tag{2.14}$$

and

$$\begin{aligned} \mu_d v(s) &\leq \mu_d(v.v_1 + \cdots +v.v_n)(s) \\ \mu_r v(s) &\subseteq \mu_r(v.v_1 + \cdots +v.v_n)(s) \end{aligned} \tag{2.15a}$$

where $v := v_1 + ... + v_n$ and

$$\begin{aligned} \mu_d v(s) &\leq \mu_d((w + v).V)(s) \\ \mu_r v(s) &\subseteq \mu_r((w + v).V)(s) \end{aligned} \tag{2.15b}$$

hold. Therefore the formulas (3.15) and (3.17) in [1] are valid whereas (3.16) is replaced by the version below [8] .

$$v.v_1 + \cdots + v.v_n \leq v \tag{2.16}$$

[8] For an example where the inequality in (2.16) becomes strict see the proof of Fact 2.4 in the appendix .

We define a **nondeterministic asynchronous dynamical system** (nondeterministic ADS) representation as a triple $R = (U, X, Y)$ where the processes U , X and Y are nondeterministic processes in arbitrary marked process spaces and U and Y *behave deterministically* . The **state function** $S : [U] \to [X]$ is defined by

$$trS(u) := (s \in trX \mid s{\downarrow_U} \in tru)$$

$$\mu_m S(u)(t) := \mu X(t)$$

the divergence marking $\mu_d S(u)(s)$ is defined by

$$\mu_d S(u)(s) := \max\{\mu_d u(s{\downarrow_U}) \, , \mu_d X(s)\}$$

and refusals at trace s , $\mu_r S(u)(s)$, is defined as follows :

Let B_s denote the set of all events corresponding to next possible transitions of the process $U/(s{\downarrow_U})$ and define the **restricted refusals** of a signal u at trace $s{\downarrow_U}$, denoted by $R_u(s{\downarrow_U})$ as the set with members consisting of the intersection of refusal sets in $\mu_r u(s{\downarrow_U})$ with B_s. Then

$$\mu_r S(u)(s) := (D \in 2^A \mid D = F \cup G \; ; F \in R_u(s{\downarrow_U}) \; ; G \in \mu_r X(s) \,)$$

The **response function** $H : [U] \to [Y]$ is defined by

$$H(u) := S(u).Y$$

Fact 2.5

(1) When $u{=}U$ the restricted refusals $R_U(s{\downarrow_U})$ consists of the empty set . This follows from the assumption that U behaves deterministically . Consequently we have the familiar formula $X = S(U)$.

(2) As in the deterministic case we have

$$S(u).U \leq u \text{ if } u \in [X.U]$$

and if $u \in [X.U]$ and it *behaves deterministically* then

$$S(u).U = u$$

where the set $[X.U]$ is called the **projected domain** of the state function. For the nondeterministic case the projected domain restricts not only the traces of applicable inputs but it bounds the maximum predictability of input blocking at each trace by setting a lower limit for the refusals imposed by the state process. Any further expansion in refusals is due to the nondeterministic behaviour of the input signal and not the state process X .The projected domain enjoys the property

$$S(\hat{u}) \leq S(u) \tag{2.17}$$

for any $u \in [U]$, where $\hat{u} := S(u).U$.

(3) $S(.)$ is a continuous and *ndes* function of its argument. However $H(.)$ is <u>not</u> necessarily a continuous function since the nondeterministic projection is not a continuous function of any of its arguments. Restriction of S to the deterministically behaving subset of its projected domain is a one-to-one function. $S(u)$ can be expressed as

$$S(u) = (X.(\eta(\delta X) \oplus U) + u).(\eta(\delta X)) \tag{2.18}$$

We present an example of a nnondeterministic ADS below. Recall that the process $HALT_m$ denotes a process with only the empty trace with the mark m ; if the mark m is ignored by writing

HALT , it is implicitly understood that it can be derived uniquely from the context - for example if it is a signal it inherits the mark of the cover process at the empty trace - .

Example 2.1

We use the job-shop example given in Example 4.1 of part 1 of this paper to exhibit the nondeterminism involved when the model is reduced to hide the private state transitions. The representation for the nondeterministic ADS is given in Figure 2.2(a) where the processes U,X and Y represented by the state machines all behave deterministically. As in Example 4.1 the discrete events e,f,c,n and a stand for 'enter shop' , 'job failed' , 'job completed' ,'next job accepted' and 'acknowledge failure' instants respectively. From this we obtain the reduced representation $R = (U,\hat{X},Y)$ where $\hat{X} := X.(U \oplus Y)$. Clearly the state process $\hat{X}$ given in Figure 2.2(b) does not behave deterministically . As a simple notation we mark every state of a nondeterministic process by the maximal refusals and the divergence index at that state where each distinct maximal refusal is specified by its subset consisting of only those events that are possible next transitions (It is implicitly understood that events that are not possible transitions belong to every maximal refusal anyway) . Therefore the notation $[\{a\};0]$ means that the transition $<a>$ can be refused although it is a possible next transition and the process does not diverge at this state . This corresponds to the case where the job is succesfully completed (a hidden event) so that failure cannot be acknowledged.

If u is defined by $u = (n \rightarrow HALT)$ where u is assumed to behave deterministically then the corresponding nondeterministic state signal $\hat{S}(u)$ and the output signal $\hat{H}(u)$ are given in Figure 2.2(c) where $\hat{S}$ and $\hat{H}$ refer to the reduced ADS $(U,\hat{X},Y)$. In both signals $<a>$ will get refused if the job is succesfully completed.

If on the other hand u is given by the recursion $u = (n \rightarrow u)$ where, this time , n is assumed to be a refusable event by u at any stage , as represented by the state machine of Figure 2.2(d) , then the corresponding state and output signals are also given in Figure 2.2(d). Note that in this case the output signal $\hat{H}(u)$ also diverges (thus the 1 in the notation of the state mark) at each trace since arbitrarily large number of n's hidden to the output are possible after each transition of the event a .

Finally we relate nondeterministic ADS to a deterministic ADS by using the relation between the ADS operators - in particular the projection operators - expressed by Fact 2.3 above and apply these relations to extend the *signal* representation of ADS to the nondeterministic case.

Let $R = (U,X,Y)$ be a given nondeterministic ADS and define its **deterministic image** by $\delta R := (\delta U, \delta X, \delta Y)$ then the following relations prevail :

$$\begin{aligned} &\delta(X.U) = \delta X.\delta U \\ &\delta(S_R(u)) = S_{\delta R}(\delta u) \\ &\delta(H_R(u)) = H_{\delta R}(\delta u) \end{aligned} \tag{2.19}$$

Conversely assume that $R = (U,X,Y)$ is a deterministic ADS and define the **nondeterministic image** of R by $\eta R := (\eta U, \eta X, \eta Y)$ then the corresponding relation is given by

$$\begin{aligned} &\eta(X.U) \geq \eta X.\eta U \\ &\eta(S_R(u)) = S_{\eta R}(\eta u) \\ &\eta(H_R(u)) \geq H_{\eta R}(\eta u) \end{aligned} \tag{2.20}$$

The extension of *representation signal* and the *system signal* to nondeterministic environments is straightforward (see part I Definition 4.2) except for the following modification involving the representation signal :

$$r(u) := S(u).(\ \eta(\delta(X)) \oplus U \oplus Y\ )$$

which is viewed as a signal with the cover set $[\eta(\delta(X)) \oplus U \oplus Y]$. The modification of replacing X by its deterministically behaving version ensures that the cover process behaves deterministically.

Using (2.19) and (2.20) one can derive the appropriate inequalities between signals corresponding to deterministic or nondeterministic image representations. For example if $r(u)$ and $\hat{r}(\delta u)$ are the representation signals of R and δR then

$$\hat{r}(\delta u) = \delta(r(u))$$
$$r(u) \leq \eta(\hat{r}(\delta u))$$

3. Equivalence and Realization by ADS

In this section we derive results on the *realization* of a given response function by either a deterministic or a nondeterministic ADS . In the first part we restrict our attention to the deterministic problem. Later we generalize the concepts and results to the nondeterministic case.

Let us start by stating the deterministic realization problem for ADS . Suppose we are given arbitrary processes U,V,Y with $V \leq U$ and a function $G : [V] \rightarrow [Y]$. What conditions should G , as well as V,U and Y , satisfy such that an ADS representation $R = (U,X,Y)$ realizes G in the sense that

$$V = X.U$$
$$G(v) = H_R(v) \quad \text{for all } v \in [V]$$

We already know that H_R is a continuous function , therefore continuity of G is a necessary condition . But it is not a sufficient one. Consider the following example where $V = U = Y$ and $trV = \{a^*\}$, i.e. both input and output processes consist of repetition of the single event a . For any non-negative integer n define $u_n \in [U]$ as the process with the traces consisting of a^n and its prefixes . The function G is defined by letting it map the argument u_n into $G(u_n) := u_{n+1}$. The function G is continuous by construction . On the other hand G cannot have a realization $R = (U,X,U)$ with $X.U = U$. To see this let $u_n \in [U]$ then we must have

$$S(u_n).U = u_n$$

by the property of the state function whereas

$$H(u_n) = S(u_n).U = u_n \neq G(u_n) = u_{n+1}$$

which implies that R cannot be a realization of G.

One way to circumvent this problem is isolating the input transitions from the output ones. For this we define *event decoupled* processes as below.

Definition 3.1

Let $\{P_i\}_{i \in I}$ be a family of processes defined in arbitrary process spaces. The family is said to be **event decoupled** *(ed)* with respect to a process Q if for each $s \in trQ$

$$\bigcap_{i \in I} tr(P_i/s\downarrow_{P_i}) = \{<>\}$$

If the process Q is not specified it is taken to be any process with the traces A^* .

As an example take processes P and R with respect to Q where $maxtrP := \{acb\}$, $maxtrR := \{bda\}$ and $maxtrQ := \{abcd \cdots \}$ where *maxtr* denotes the maximal trace of the process. Then although P and R share the events a and b in their traces they are nevertheless *ed* with respect to Q since the traces of Q guarantee that P and R do <u>not</u> share either a or b as a common transition along the projection of the traces of Q as required by the definition .

A simple special case for the above condition would obtain when the alphabets αP_i of the processes P_i are disjoint sets where for any process P , αP denotes the set of all events that appear at least in one trace of P. It can easily be seen that if either Q has traces A^* or Q is the sum of the P_i processes then the special case stated above has to prevail.

In order to derive a second restriction on G we note an important property of both the state and response functions of ADS , namely the preservation of the least upper bound of signals which we refer to as the **lub property** [9] :

$$\begin{aligned} S(u_1 \sqcup u_2) &= S(u_1) \sqcup S(u_2) \\ H(u_1 \sqcup u_2) &= H(u_1) \sqcup H(u_2) \end{aligned} \tag{3.1}$$

where S and H denote the state and response functions of an ADS representation (U,X,Y) and u_1 and u_2 are arbitrary processes in $[U]$ and for arbitrary signals p and q in the same signal space the *lub* (least upper bound) function ' $\sqcup$ ' is defined by

$$tr(p \sqcup q) := trp \cup trq$$

Clearly if G is to be realized then it must also satisfy the *lub* property stated above. The following realization theorem states the conditions for the existence of an ADS that realize a given response function with the restrictions discussed above . Before we stating the theorem , we define a *trajectory* process which is used both in the proofs of Theorem 3.1 and Theorem 3.2 . A **finite trajectory** is a marked process consisting of a *single* trace s and all its prefixes. The number $\#s$ is called the length of the trajectory. A **trajectory** is defined as the limit of a chain of finite trajectories.

Theorem 3.1 (*Deterministic Realization*)

Let U,V and Y be given marked processes in given marked process spaces where $V \leq U$ and assume that V and Y are *event decoupled* . Then a function $G : [V] \rightarrow [Y]$ is *continuous* and satisfies the *lub property* if and only if there exists an ADS representation $D = (U,X,Y)$ that realizes the function G .

If $R = (U,X,Y)$ and $\hat{R} = (U,\hat{X},Y)$ are two different realizations of a given function G then it does not necessarily follow that $X.(U \oplus Y) = \hat{X}.(U \oplus Y)$. That is , the relative interleaving order of the output with respect to the input may be arbitrary. Consider the following response function G :

$$\begin{aligned} &G(HALT) := HALT \\ &G(\{<a>\}) := \{<b>\} \\ &G(\{ac\}) := \{bd\} \end{aligned}$$

where U=V is given by the state machine in Figure 3.1 (a) and Y in Figure 3.1 (b) . In the above notation we denote the signals in the domain and the range by their maximal traces . We present in Figure 3.2 (a) and Figure 3.2 (b) the state diagrams of two different state processes X and $\tilde{X}$ that realize the function G above . The principal difference of the second realization is that it uses *buffering*. If we

[9] The derivation of (3.1) is obvious and omitted.

think of the input transitions as demands for consecutive services and the output transitions as the execution of these services then the second realization buffers the next demand before it executes the first one.

In order to demonstrate , in general , that the distinction between different realizations can be attributed to different buffering strategies we focus to the proof of the realization theorem given in the appendix. The proof first constructs a pre-state function on the input *trajectories* , which sets a lower bound on the state function and extends these to arbitrary signals by using the continuity and *lub* property of the response function . The inductive step in defining $\hat{S}(u)$ for a trajectory u is given by (A10). Clearly choice of different R_u satisfying (A9) in the proof will yield different ρ_u and hence a different pre-state function by (A10). The choice of R_u as $\hat{R}_u$ corresponds to the maximal (infinite) buffering strategy. To see this observe that $v := s\hat{\ }<a>\hat{\ }q$ is a legitimate trace of $\hat{S}(u)$ where q is any trace of $G(u)$. This corresponds to the choice of a triple $(q,q,< >)$ which certainly satisfies (A6) .Therefore all the output traces , i.e. $trG(u)$, can be generated after buffering the entire input trajectory $s \hat{\ } <a>$.

Now consider the following prudent buffering strategy by choosing R_u as follows :

For each $r \in trG(u)$ decompose the trace r as $r = q\hat{\ }\tau$ where q is the *largest* prefix of r in $trG(u')$. Choose any $p \in trS(u')$ such that $p\downarrow_U$ is a *largest* prefix - note that p may not be unique - of s . We let R_u be taken as the set of all such (r,q,p) subject to $q \neq r$ as r ranges in $trG(u)$.

If for the moment we make the assumption that $\hat{S}(u) = S_D(u)$ this realization corresponds to a more efficient buffering strategy since s - or the largest prefix of s - has already been serviced by q in this definition. So the suffix trace τ corresponds to the service for the additional demand specified by $<a>$ - or $w\hat{\ }<a>$ - . Since choice of p above is not unique and in general $\hat{S}(u) \leq S_D(u)$ it is not clear that this prudent buffering strategy is actually a *minimal* strategy where the word "minimal" is made precise by the following definition.

Definition 3.2

A function G satisfying the hypothesis of Theorem 3.1 is called a **realizable** function. If there exists a realization $R = (U,X,Y)$ where X is generated by a finite state machine G is called a **regular realizable** function. A realization $R = (U,X,Y)$ of a regular realizable G is called a **minimal realization** if for any other realization $\hat{R} = (U,\hat{X},Y)$, $|X| \leq |\hat{X}|$ where $|P|$ denotes the cardinality of the minimal state machine (no. of states) that generates the traces of the process P . The unique cardinality of a minimal realization is called the **complexity index** of the function G.

Example 3.1

Let $U = V$ be a process with traces $\{(a_1, \cdots, a_n)^*\}$ and Y be a process with the traces $\{(\overline{a}_1, \cdots, \overline{a}_n)^*\}$. For each trace s of U there is a trace $\overline{s}$ of Y where each a_j in s is transformed to $\overline{a}_j$ in $\overline{s}$. We define the function $G : [U] \rightarrow [Y]$ by first defining it on each trajectory τ in $[U]$ as

$$G(\tau) := \overline{\tau}$$

where τ has the maximal trace s and $\overline{\tau}$ is the trajectory in $[Y]$ with the maximal trace $\overline{s}$. We extend G to entire $[U]$ by using the *lub* property and and assume it to be continuous so that it uniquely extends to chain limits. A minimal realization of G is given by using the state process in Figure 3.3 . Therefore the complexity index of G is $n+1$ although it may have buffered realizations with arbitrarily large number of states. Therefore one can abstract the complexity of a task defined in an input-output functional form from its buffered implementations as seen in this simple example.

We now extend the realization problem to nondeterministic environments. We replace the processes $V \leq U$ and Y by deterministically behaving nondeterministic processes in appropriate nondeterministic spaces and as before let $G : [V] \rightarrow [Y]$. We shall call a nondeterministic ADS , $R := (U, X, Y)$ a **realization** of G if :

(1) $\delta(X.U) = \delta(V)$,

(2) $\delta(G(v)) = \delta(H_R(v))$ for all $v \in [V]$.

This formulation reduces the problem to a deterministic one by matching only the deterministic images of the response and the projected domain. If we look at the practical aspect of the realization problem then it is reasonable to assume that G is given and behaves deterministically - to be made precise soon - and it is desired that a realization of G behaves deterministically as well. The fact that this is not always achievable is demonstrated below.

It is instructive to understand the possibility of inherent nondeterminism that may arise from the structure of V and G and the realization mechanism which involves the projection operator. Consider the example where $V = U$, $trU = \{<>,<a>\}$, $trY := \{b^*\}$ and $G(v) := Y$ for all $v \in [V]$. Then if (U, X, Y) is any realization of this response function then we must have by definition $\delta(X.U) = \delta V$ and therefore $tr(X.U) = trU$. On the other hand again by definition $\delta(G(V)) = \delta(H_R(V)) = \delta(X.Y) = Y$ which implies that X must generate arbitrarily many b's before or after (or both) generating the input transition $<a>$. But this implies that either $\mu_d(X.U)(<>) = 1$ or $\mu_d(X.U)(<a>) = 1$ (or both are 1), that is the process $X.U$ diverges at some trace. Therefore for any realization we must have $X.U \leq V$ and the (projected domain) process $X.U$ cannot behave deterministically .

Now consider the example by again letting $V = U$ where $trU = \{a^*\}$, Y as above and $G(v) := HALT$ for all $v \in [V]$. Then by definition of a realization $\delta(G(V)) = \delta(H_R(V)) = \delta(X.Y)$ and therefore X is not allowed to generate any trace with a b transition in it. On the other hand again by definition we have $\delta(X.U) = \delta(V)$ which implies that X must generate arbitrarily many a's which in turn implies that $\mu_d(H_R)(V)(<>) = 1$ and so for any realization we have $H_R(V) \leq G(V)$ without equality since $H_R(V)$ cannot behave deterministically.

Finally consider the following typical example of nondeterministic behaviour via the refusals. Let G be defined as

$$G(HALT) := HALT$$
$$trG(<a>) := \{<>,<b>\}$$
$$trG(<c>) := \{<>,<d>\}$$

where $U = V$ and $trU = \{<>,<a>,<c>\}$; Y is any process that includes the traces $<b>$ and $<d>$ and as before arguments of functions are denoted by the maximal traces of the corresponding signals. We assume that G has the *lub* property therefore $trG(<a>,<c>) = \{<>,<b>,<d>\}$. Then for any realization of G we must have

$$\mu_r S(<a>,<c>)(<>) = 2^{\{b,d\}}$$

although both $<b>$ and $<d>$ belong to the traces of $G(<a>,<c>)$ which violates the deterministic behaviour requirement . In order to see this let (U, X, Y) be any realization of G , then $G(u) = S(u).Y$ where $tru = \{<>,<a>,<c>\}$ and if $S(u)$ makes an input transition it can only be an $<a>$ or a $<c>$. But if it is an $<a>$ then $d \notin tr(S(u)/a).Y$ and if it is a $<c>$ then $<b> \notin tr(S(u)/c).Y$. This is because such situations would violate the definition of G above. This example shows that expecting deterministic behaviour in general is ruled out . However , as to be demonstrated below , if the input signal u is restricted to be a *trajectory* then the kind of situation above could be circumvented by appropriate realizations. Since in reality real time operations only allow for trajectory inputs the kind of inherent nondeterminism described above does not seem to have any practical significance .

These examples , degenerate as they may be , exhibit that nondeterministic behaviour , both at the input blocking level and the output generation level , may be an inherent property of the function (and its projected domain) as opposed to being a property of a particular realization of it . We can therefore ask the legitimate question as to what requirements G should satisfy such that it has realizations that do not give rise to nondeterminism whenever the input signal behaves deterministically. In the following part of this section we formulate the nondeterministic realization problem and its solution in a way related to this question .

Let P be any process in a nondeterministic process space $\overline{\Pi}$ and define P^{δ} as the subset of deterministically behaving signals in $[P]$. This set can be defined using the operators δ and η as follows :

$$P^{\delta} := (p \in [P] \mid \eta(\delta(p)) = p) \tag{3.2}$$

It is easy to show that P^{δ} is *closed* under chain limits. It is also evident that P^{δ} is isomorphic to the set $[\delta(P)] \subseteq \Pi$ in the obvious way.

We state below definitions on G that characterize the potential nondeterminism in realization of it. Recall that a process or a signal is called *finite* [10] if its trace set is a finite set. It is called *infinite* if it is not finite.

Definition 3.3

The function $G : [V] \rightarrow [Y]$ is called **bounded from above** on a subset K of its domain if it maps finite signals in K into finite signals . It is called **bounded from below** on a subset L of its range if the inverse image of finite signals in L are finite signals. It is called **stable** relative to (K,L) if it is bounded from above on K and bounded from below on L . It is said to **behave deterministically** on a subset K of $[V]$ if it maps deterministically behaving signals in K into deterministically behaving signals , that is

$$v \in V^{\delta} \cap K \Rightarrow G(v) \in Y^{\delta}$$

The **deterministic image** of G denoted $\delta G : [\delta V] \rightarrow [\delta Y]$ is defined by

$$\delta G(w) := \delta(G(\eta(w)))$$

for all $w \in [\delta V]$.

We can state the nondeterministic realization theorem as below :

Theorem 3.2 (*Nondeterministic Realization*)

Consider the nondeterministic realization problem where the input and output spaces are assumed to be *event decoupled* . There exists a realization $R = (U,X,Y)$ of G with $X \in [U \oplus Y]$ such that :

(1) $X.U = V$,

(2) $v \in V^{\delta} \cap T \Rightarrow H_R(v) \in Y^{\delta}$, where T denotes the subset of all *trajectories* in $[V]$,

(3) $G(v) = H_R(v)$ for all $v \in V^{\delta} \cap T$,

if and only if the function G *behaves deterministically* on T , is *stable* relative to $(T, G(V))$ and its *deterministic image* δG is continuous and satisfies the *lub* property.

[10] This is not to be confused with a process or signal generated by a finite state machine since infinite processes can be generated by finite state machines.

Theorem 3.2 states the necessary and sufficient conditions for a function to be realizable in a deterministically behaving way. If G is not *stable* then any realization will lead to nondeterminism as illustrated by the degenerate examples given above. Under the absence of these deterministic behaviour conditions Theorem 3.1 gives the necessary steps involved in realizing the deterministic image of G given by δG defined above.

As in the deterministic case the buffering strategies and realizations are related in a similar way. The additional aspect that enters into the nondeterministic realization theory is the possibility of constructing a state function so as to avoid nondeterministic behaviour. Although the proof of Theorem 3.2 in the appendix uses an infinite buffer construction for avoiding nondeterminism , specific examples may require finite buffer capacity for maintaining deterministic behaviour.

Example 3.2

Consider the nondeterministic realization problem where $V \leq U$ and Y are given by the state machines in Figure 3.4. The function G is defined on the trajectory subset of $[V]$ as follows :

(1) On the special trajectories below , G is defined as

$$trG(\{<a_1>\}) := \overline{<b_1>}$$

$$trG(\{<a_2>\}) := \overline{<b_2>}$$

$$trG(\{a_1a_2a_3\} := \overline{<b>\bigcup(b_1b_2)}$$

where the argument of G , which is a trajectory , is specified by its maximal trace and the '—' symbol on the right denotes the prefix closure of the trace set in question.

(2) Any trajectory with a maximal trace of the form $(c_1 \cdots c_n)$ maps into a process with traces $\{\tilde{c}_1 \cdots \tilde{c}_n\}$ where each c_j is either an a_1 or an a_2 in which case the corresponding $\tilde{c}_j$ is b_1 or b_2 in the same order ; or some c_j stands for the triple $(a_1a_2a_3)$ (this is the only form in which the transition a_3 can appear according to the definition of V in Figure 3.4) in which case $\tilde{c}_j$ can be both $<b>$ or (b_1b_2) and all possible unions are taken to accomodate both possibilities. Therefore , for example , the trajectory with a maximal trace

$$(a_1a_2a_3a_2a_1a_2a_3)$$

will map under G into a signal with traces equal to the closure of the following :

$$(b_1b_2b_2b_1b_2)\bigcup(bb_2b_1b_2)\bigcup(b_1b_2b_2b)\bigcup(bb_2b)$$

It can easily be observed that the function G defined above is continuous and can be extended to the entire $[V]$ imposing the *lub* property. It is assumed that all the processes in question behave deterministically. In Figure 3.5 two different state processes corresponding to two different realizations of G are given . In Figure 3.5 (a) the realization suffers from the defect that $H((a_1a_2))$ does not behave deterministically at the null trace $< >$. Indeed let $t := a_1a_2$ then although $t\downarrow_Y = < >$ and $<b_1> \in trH((a_1a_2))$ it cannot be completed as $t \hat{} t' \in trS((a_1a_2))$ such that $(t \hat{} t')\downarrow_Y = <b_1>$. Therefore $<b_1>$ is refused by $H((a_1a_2))$, violating a deterministic behaviour requirement.

In Figure 3.5 (b) this is fixed , yet , this time $X.U$ still does not behave deterministically at the trace (a_1a_2) . To see this consider the trace $t := a_1b_1a_2b_2 \in trX$ then although $<a_3> \in tr(X.U)/(a_1a_2)$ and $t\downarrow_U = (a_1a_2)$ there does not exist $t \hat{} t' \in trX$ such that $t'\downarrow_U = a_1a_2a_3$. Therefore $<a_3>$ is refused by $(X.U)/(a_1a_2)$ violating another condition for deterministic behaviour.

The solution for a deterministically behaving synthesis is given in Figure 3.6 . Note that more buffering was required to maintain deterministic behaviour.

The results and discussions above suggest the following definitions of equivalence for ADS representations :

Definition 3.4

Two ADS representations $R = (U,X,Y)$ and $\tilde{R} = (U,\tilde{X},Y)$ are said to be **strongly equivalent** if $X.U = \tilde{X}.U$ and

$$S_R(u).(U \oplus Y) = S_{\tilde{R}}(u).(U \oplus Y) \quad \text{for all } u \in [U] \tag{3.3}$$

They are said to be **weakly equivalent** if the formula (3.3) is replaced by

$$H_R(u) = H_{\tilde{R}}(u) \text{ for all } u \in [U] \tag{3.4}$$

The definitions of strong and weak equivalence above apply for both deterministic and nondeterministic ADS. Strong equivalence is indifferent to *private* state transitions but the input and output transitions must occur in an exact pre-specified order . On the other hand weak equivalence only demands that the output for each input is fixed but the relative order of input transitions with respect to output transitions may vary due to buffering considerations. Weak equivalence is meaningful when concern for the input-output functionality of ADS overrides buffering strategies of implementation. In this sense it captures a *buffer-free* representation of a system. Clearly strong equivalence is a refinement of weak equivalence.

It will be shown in part III of this sequence that a response function of a *loop-free* interconnected set of ADS is only a function of the weakly equivalent representative of each ADS. If , however , there are loops then , in general , it is a function of strongly equivalent representatives of component ADS representations . This result is related to the scenario analysis of Dennis for data flow networks [7] . Scenario analysis is a method of fitting *scenarios* that impose a partial order between the input and output transitions of an interconnected data flow graph when individual input-output *histories* are insufficient to derive the overall history . The correspondence is as follows : histories and scenarios in [7] correspond to the response functions and the strongly equivalent representatives of individual ADS representations of our approach respectively . The ADS approach is more general in the sense that it has a richer signal space of representations and it allows for nondeterminism [11] .

We define an ADS as the equivalence class of strongly equivalent representations . Among these there is a distinguished one given by $D := (U,C,Y)$ [12] , where C is the unique process in $[U \oplus Y]$ given by $C := X.(U \oplus Y)$ and $[X]$ is the state space of any member of the equivalence class. We call this the **canonical representation** of the dynamical system , which is well-defined by the fact that $S(U) = X$.

It is legitimate to ask at this stage whether the functions δ and η preserve equivalence , strong or weak . It is easily observed that if R is strongly equivalent to $\hat{R}$ then it is not necessarily true that ηR is strongly equivalent to $\eta \hat{R}$. Therefore the map η is not well-defined on the set of deterministic ADS identified by strongly equivalent classes . On the other hand the function δ preserves both weak and strong equivalence. Therefore the inverse image of η obtained by applying δ to nondeterministic processes map sets of equivalence classes of nondeterministic ADS representations into *subsets* of the corresponding equivalence classes of deterministic ADS. Based on this observation we define *nondeterministic equivalence* as below

[11] The term nondeterminism used in [7] merely states the possibility of non-trajectory responses to trajectory inputs and has little to do with the concept of nondeterminism in this paper.

Definition 3.5

Let R be a given deterministic ADS representation . We say that R is **strongly (weakly) nondeterministically equivalent** to $\hat{R}$ if ηR is strongly (weakly) equivalent to $\eta\hat{R}$.

The strong nondeterministic equivalence partitioning reflects precisely the differentiating power of the nondeterministic model we use. In other words as was emphasized in section 1 , if one starts with a deterministic ADS model and use projections to obtain a nondeterministic reduced model , then the differentiating power of the nondeterministic model is up to a nondeterministic equivalence class in the original deterministic ADS. It can easily be observed that strong nondeterministic equivalence is a refinement of strong equivalence.

Finally we relate dynamical concepts of equivalence summarized above to the well-known concept of *bisimulation* [5] , a definition of equivalence that is *dynamically invariant* in the terminology of our framework. The definition of bisimulation as given below is an adaptation of the concept to ADS environment .

Definition 3.6

Consider the set of all deterministic ADS representations $\{R_X = (U,X,Y)\}_X$ where U and Y are fixed input and output processes and X is any marked process . An equivalence relation ' $\equiv$ ' on this set is called a **bisimulation** if $R_X \equiv R_{\hat{X}}$ implies that for each nondecreasing sequence of traces $s_j \in trX$ there exists a nondecreasing sequence of traces $t_j \in tr\hat{X}$ and vice-versa such that

(1) $s_j \downarrow_{(U \oplus Y)} = t_j \downarrow_{(U \oplus Y)}$ for each j ,

(2) $\#(s_j)$ is an unbounded sequence if and only if $\#(t_j)$ is an unbounded sequence ,

(3) $R_X / s_j \equiv R_{\hat{X}} / t_j$ for each j where for any representation $R = (U,X,Y)$ the post-representation R/s is defined as $R/s := (U/(s\downarrow_U), X/s, Y/(s\downarrow_Y))$ [1] .

Fact 3.1

(1) Strong nondeterministic equivalence is not a bisimulation .

(2) Any bisimulation refines strong nondeterministic equivalence .

The following example illustrates concepts of equivalence explained above

Example 3.3

Consider the three deterministic ADS representations with the state processes given in Figure 3.7 . It can easily be shown that all these representations are strongly equivalent to each other. On the other hand R_2 is nondeterministically equivalent to R_3 whereas R_1 is nondeterministically equivalent to neither . To see this observe that the process $X_j.(U \oplus Y)$ may refuse both d and e for $j = 2,3$ whereas it cannot refuse these events for $j = 1$ after producing the input trace bc . Further details are obvious and omitted.

We claim that R_2 cannot be equivalent to R_3 under any bisimulation relation. Consider the state trace βb in X_3 (in the definition of bisimulation we take the infinite sequence constant for each j) then there are two possibilities in picking a corresponding trace from X_2 if condition (1) of bisimulation is to be satisfied :

(1) We pick the trace $b\tau$ in X_2 . But then $tr(X_2/(b\tau)) = \{<>,<c>,ce\}$ whereas $tr(X_3/(\beta b)) = \{<>,<c>,cd\}$ and therefore these two post-representations cannot be bisimulation equivalent since ce is incompatible with cd and violate condition (1) .

(2) We pick the trace b in X_2. In order to show that X_2/b cannot be bisimulation equivalent to $X_3/(\beta b)$ we move one more step by considering the trace τc in X_2/b. The only candidate in $X_3/(\beta b)$ is the trace c and again incompatibility arises since

$$tr((X_2/b)/(\tau c)) = \{< >,<e>\}$$
$$tr((X_3/(\beta b))/c) = \{< >,<d>\}$$

4. Conclusions

We have extended the tools and concepts of ADS environment to deal with nondeterministic signals. For that purpose we defined a model of nondeterminism similar to Hoare's CSP [4] and redefined ADS as an input-output dynamical representation on nondeterministic signals . We showed by Fact 3.1 that our model of nondeterminism is superior to deterministic ADS and inferior to any bisimulation based nondeterminism [5] in its expressive power. The real practical advantage of the model will be demonstrated in part III of this series when interconnected nondeterministic ADSs will be used as a representational and computational tool for parallel program specification and verification .

We have defined the realization problem for the deterministic and nondeterministic cases and presented conditions of realizability for response functions. Instead of supplying detailed solutions to computational problems of ADS formulations we have tried to emphasize basic concepts and questions of asynchronous dynamics. For example the problem of computing the complexity index of a given response function or the theoretical problem of resolving whether deterministic behaving synthesis can be achieved via finite buffering (recall that the proof of Theorem 3.2 uses an infinite buffering realization) are problems that have not been tackled here . These problems and others in which the freedom in weak equivalence is used as an optimization parameter - e.g. maximizing parallelism by choosing the largest relevant state process subject to buffering constraints - are open .

APPENDIX

Proofs of Fact 2.1 , Fact 2.2 and Fact 2.3 are straightforward and omitted.

Proof of Lemma 2.1

The proof in one direction is immediate. Let $\{p_{0i}\}_i$ be a countable and infinite subset of T and for each j define inductively a subsequence $\{p_{ji}\}_i$ of the infinite trace set of the previous stage as follows :

> At stage $j-1$ the set $\{p_{(j-1)i}\}_i$ has the property that the first $j-1$ entries of $p_{(j-1)i}$ are identical for each i and at stage j we choose *any* subsequence of this collection denoted $\{p_{ji}\}_i$ such that the first j entries of each trace are identical. Observe that we can choose such infinite subsequences at each stage . This is because when traces are grouped with first j entries identical there are finite such groups and therefore at least one group must have infinite elements and we choose the subsequence with the members of this group. Finally we set $t_j := p_{jj}$ and the result follows.

Proof of Fact 2.4

(1) The proof is straightforward and omitted.

(2) First observe that

$$\mu_d((P.\hat{P})/s)(t) = \mu_d(P.\hat{P})(s\hat{\ }t)$$
$$\mu_r((P.\hat{P})/s)(t) = \mu_r(P.\hat{P})(s\hat{\ }t) \qquad \text{(A1)}$$

by the transitivity of the post-process and for any $m \in trP$ such that $m \downarrow_{\hat{P}} = s \hat{} t$ there is a decomposition (not necessarily unique)

$$m = m_1 \hat{} m_2 \tag{A2}$$

where

$$\begin{aligned} m_1 \downarrow_{\hat{P}} &= s \\ m_2 \downarrow_{\hat{P}/s} &= t \end{aligned} \tag{A3}$$

We first prove the equality of divergences in (2.8) . Let $\mu_d(P.\hat{P})(s\hat{}t) = 1$ then there are three cases to consider :

(i) The set T in the definition of the extended projection is infinite. Therefore there exists $q \in trP$ such that $q \downarrow_{\hat{P}} = s \hat{} t$ and by Lemma 2.1 there is an infinite trajectory τ with traces t_j in P/q which is transparent to $Q/(s\hat{}t)$, i.e. $t_j \downarrow_{Q/(s\hat{}t)} = < >$. Using the decomposition given by (A3) we can identify $q = m_1 \hat{} m_2$ and the trajectory τ satisfies the corresponding condition of divergence for $P/m_1.\hat{P}/s$ at t since $((P/m_1)/m_2) = P/q$ and $m_2 \downarrow_{\hat{P}/s} = t$ by (A3) above. Therefore $\mu_d(P/m_1.\hat{P}/s)(t) = 1$ and by the definition of union operation given by (2.7b) with $v = m_1$ we have

$$\mu_d\{ \bigcup_{v \downarrow_{\hat{P}} = s} (P/v.\hat{P}/s)\}(t) = 1$$

(ii) There exists $q \in trP$ where $q \downarrow_{\hat{P}} = s \hat{} t$ and $\mu_d P(q) = 1$. Then using the decomposition (A3) above the same reasoning holds for $P/m_1.\hat{P}/s$ at t since $\mu_d(P/m_1)(m_2) = \mu_d P(q) = 1$. The rest follows from the same union argument above.

(iii) We have $\mu_d(\hat{P})(s\hat{}t) = 1$ therefore $\mu_d(\hat{P}/s)(t) = 1$ and so for any m , $\mu_d(P/m.\hat{P}/s)(t) = 1$. The rest again follows from the union argument.

These arguments are all reversible and we omit the details . This establishes the equality of the divergences . We next prove the equality of refusals. Again we prove the result in one direction. Reversing the arguments is straightforward. Let $D \in \mu_r(P.\hat{P})(s\hat{}t)$ then ruling out the trivial case where $\mu_d(P.\hat{P})(s\hat{}t) = 1$ we have :

There exists $t' \in trP$, $B \in \mu_r P(t')$ and $C \in \mu_r \hat{P}(s\hat{}t)$ such that

$$\begin{aligned} & t' \downarrow_{\hat{P}} = s \hat{} t \\ & (t'\hat{}<a>) \downarrow_{\hat{P}} = s \hat{} t \Rightarrow a \in B \\ & D = B \bigcup C \end{aligned}$$

Letting $m := t'$ where m is given by (A2) we have $B \in \mu_r(P/m_1)(m_2)$ and $m_2 \downarrow_{\hat{P}/s} = t$ by (A3). If $(m_2\hat{}<a>)\downarrow_{\hat{P}/s} = t$ then $m\hat{}<a> \downarrow_{\hat{P}} = s\hat{}t$ and by hypothesis $a \in B$. Also $C \in \mu_r(\hat{P}/s)(t)$ by transitivity of post-process which proves that $D = B \bigcup C \in \mu_r(P/m_1.\hat{P}/s)(t)$ and the rest follows from the union formula (2.7a) . This completes the proof of the post-process formula.

The proof for the choice function formula is similar and omitted.

(3) We only prove the formula (2.11) both for divergence and refusals which in turn yield (2.13) . We also prove the equality version of (2.11) , hence (2.13) , under the conditions (C1) and (C2). The rest is similarly proved and is omitted .

In order to prove (2.11) for divergence we first let $\mu_d(u.w)(s) = 1$ and show that $\mu_d(u.(w \oplus v).w)(s) = 1$. There are three possibilities :

(i) For some $q \in tru$ and $q \downarrow_w = s$ the process u/q has infinite traces of the form t_j transparent to w/s, i.e., $t_j \downarrow_{w/s} = < >$. We define the partitioning

$$m \,\hat{}\, m_j := (q \,\hat{}\, t_j)\downarrow_{w\oplus v} = q\downarrow_{w\oplus v} \,\hat{}\, t_j\downarrow_{(w\oplus v)/q}$$

and claim that $(u.(w\oplus v))/m$ has traces m_j , tranparent to w/s where $m := q\downarrow_{w\oplus v}$. Clearly , using the deterministic version of the formula under proof

$$(m \,\hat{}\, m_j)\downarrow_w = s$$

and therefore

$$m_j\downarrow_{w/s} = < >$$

which proves the transparency. If the set $\{m_j\}$ is infinite we are done by condition (1) of divergence definition. If , on the other hand , m_j is a finite set then for some J , $m\hat{}m_j = m\hat{}p \in tr(u.(w\oplus v))$ for all $j \ \geq J$. But this implies that

$$\mu_d(u.(w\oplus v))(m\hat{}p) = 1$$

by condition (1) of the divergence definition applied to $u.(w\oplus v)$. The result follows by applying condition (2) of the divergence definition.

(ii) For some $t \in tru$ such that $t\downarrow_w = s$, $\mu_d u(t) = 1$. Then by chain use of condition (2) we have

$$\mu_d(u.(w\oplus v))(t\downarrow_{w\oplus v}) = 1$$
$$\mu_d(u.(w\oplus v).w)(s) = 1$$

(iii) $\mu_d w(s) = 1 \Rightarrow \mu_d(u.(w\oplus v).w)(s) = 1$.

This proves the inequality for divergences.

We next prove the inequality (2.11) for refusals. For this first let

$$D \in \mu_r(u.w)(s)$$

then , ruling out the trivial case where both processes diverge , there exists $t \in tru$ where $t\downarrow_w = s$ and $B \in \mu_r u(t)$ such that

$$(t \,\hat{}\, <a>)\downarrow_w = s \Rightarrow a \in B$$

and $D = B \cup C$ for some $C \in \mu_r w(s)$. We then deduce that

$$B \in \mu_r(u.(w\oplus v))(t\downarrow_{w\oplus v})$$

since

$$[(t \,\hat{}\, <a>)\downarrow_{w\oplus v} = t\downarrow_{w\oplus v}] \Rightarrow [(t \,\hat{}\, <a>)\downarrow_w = t\downarrow_w = s] \Rightarrow a \in B$$

where we took the null set in $\mu_r(w \oplus v)(t\downarrow_{w\oplus v})$ to add to B so that we obtained B again . But this implies that

$$D = B \cup C \in \mu_r(u.(w\oplus v).w)(s)$$

since

$$[((t\downarrow_{w\oplus v}) \,\hat{}\, <a>)\downarrow_w = (t\downarrow_{w\oplus v})\downarrow_w] \Rightarrow [(t \,\hat{}\, <a>)\downarrow_w = t\downarrow_w = s] \Rightarrow a \in B$$

This proves the formula (2.13). We next prove the equality version of (2.11) , hence (2.13) , when (C1) and (C2) hold. Before however consider the following two counter-examples to (2.11) corresponding to the violations of the conditions (C1) and (C2) respectively :

1 - Take

$$tru = \{< >,<a>,<b>,<c>\} \ ; \ \mu_r u(< >) = 2^{\{a\}}$$
$$trw = \{< >,<b>,<c>\} \ ; \ \mu_r w(< >) = 2^{\{a,b\}} \cup 2^{\{a,c\}}$$
$$trv = \{< >,<b>,<c>\} \ ; \ \mu_r w(< >) = 2^{\{a,b,c\}}$$

where all divergence indices are assumed to be 0 . Then we have

$$\mu_r(u.w>(< >) = 2^{\{a,b\}} \cup 2^{\{a,c\}}$$

whereas

$$\mu_r(u.(w \oplus v).w)(< >) = 2^{\{a,b,c\}}$$

since

$$\mu_r(u.(w \oplus v))(< >) = 2^{\{a,b\}} \cup 2^{\{a,c\}}$$

2 - Take tru : $\{a^n cb^n\}$ and trw : $\{c^*\}$, $trv := \{b^*\}$. Clearly $\mu_d(u.w)(c) = 0$ by the paragraph following Definition 2.3 . On the other hand the process $u.(w \oplus v)$ has traces $\{cb^*\}$ and therefore $\mu_d[(u.(w \oplus v)).w](c) = 1$ where after c infinitely many b's are transparent to w/c . Therefore the process $u.(w \oplus v).w$ diverges whereas $u.w$ does not .

We proceed with the proof of (2.11) for the special case . Let $\mu_d(u.(w \oplus u).w)(s) = 1$ then using (C1) only two possibilities arise :

(i) There exists $q \in tr(u.(w \oplus v))$ such that $q \downarrow_w = s$ and $(u.(w \oplus v))/q$ has infinitely many traces t_j transparent to w/s . From this we deduce an infinite set of traces $m_j \in tru$ such that by appropriate partitioning we may write

$$m_j := m_{1j} \hat{} m_{2j}$$

where

$$m_{1j} \downarrow_{w \oplus v} = q$$
$$m_{2j} \downarrow_{(w \oplus v)/q} = t_j$$

By assumption (C2) u is a process generated by a finite state machine therefore we can divide the infinite set of traces $\{m_{1j}\}_j$ into a finite set of groups such that each group represents the common state into which the initial state of u is driven after executing any member trace m_{1j} . Since the original sequence is infinite at least one group must have infinitely many elements. Let r denote any member of the trace group denoted by the infinite index set R corresponding to the state with the infinite subsequence constructed above then u/r has infinitely many traces , namely $\{m_{2j}\}_{j \in R}$, that is transparent to w/s since

$$m_{2j} \downarrow_{w/s} = t_j \downarrow_{w/s} = < >$$

which proves that $\mu_d(u.w)(s) = 1$.

(ii) For $t \in tr(u.(w \oplus v))$ where $t \downarrow_w = s$, $\mu_d(u.(w \oplus v))(t) = 1$. Again by (C1) there are only two possibilities :

(I) There exists $q \in tru$ where $q \downarrow_{w \oplus v} = t$ and u/q has infinite collection of traces r_j that are transparent to $(w \oplus v)/t$. But since

$$(q \downarrow_{w \oplus v}) \downarrow_w = q \downarrow_w = t \downarrow_w = s$$

and

$$r_j \downarrow_{w/s} = < >$$

we have $\mu_d(u.w)(s) = 1$ by condition (1) of divergence .

(II) There exists $t' \in tru$ such that $t' \downarrow_{w \oplus v} = t$ and $\mu_d u(t') = 1$. But then $t' \downarrow_w = t \downarrow_w = s$ and therefore $\mu_d(u.w)(s) = 1$ by condition (2) of divergence .

Next we prove the refusals side of equality version of (2.11) . For this let $D \in \mu_r(u.(w \oplus v).w)(s)$ then either there is $q \in tr(u.(w \oplus v))$ where $q \downarrow_w = s$ and $B \in \mu_r(u.(w \oplus v))(q)$ such that

$$\begin{aligned} &B \in \mu_r(u.(w \oplus v))(q) \\ &(q \mathbin{\hat{}} <a>)\downarrow_w = s \Rightarrow a \in B \end{aligned} \tag{A4}$$

and $D = B \cup C$ for some $C \in \mu_r w(s)$; or $\mu_d(u.(w \oplus v).w)(s) = 1$. The latter implies that $\mu_d(u.w)(s) = 1$ by the previous proof and and therefore $D \in \mu_r(u.w)(s) = 2^A$. Otherwise there are two possibilities :

(I) There is $p \in tr(u)$, $\hat{B} \in \mu_r u(p)$ and $\hat{C} \in \mu_r(w \oplus v)(q)$ such that

$$\begin{aligned} &p \downarrow_{w \oplus v} = q \\ &[(p \mathbin{\hat{}} <a>)\downarrow_{(w \oplus v)} = p \downarrow_{(w \oplus v)}] \Rightarrow a \in \hat{B} \\ &B = \hat{B} \cup \hat{C} \end{aligned} \tag{A5}$$

Using (A4) and $p \downarrow_w = q \downarrow_w = s$

$$[(p \mathbin{\hat{}} <a>)\downarrow_w = s] \Rightarrow [(q \mathbin{\hat{}} <a>)\downarrow_w = s] \Rightarrow a \in B$$

We claim that $a \in \hat{B}$. If not it must belong to $\hat{C}$, but using the condition (C1) that w and v , hence $w \oplus v$, behaves deterministically we cannot have $<a> \in tr((w \oplus v)/q)$ and therefore the hypothesis of the second proposition in (A5) is satisfied which implies $a \in \hat{B}$, which in turn proves the claim .

Using the claim proved above we can write D as $D = \hat{B} \cup (\hat{C} \cup C)$ where $\hat{B}$ satisfies the required conditions and it is enough to show that $(\hat{C} \cup C) \in \mu_r w(s)$. But this follows because first $\hat{C} \in \mu_r(w \oplus v)(q)$ and by definition of the sum operator $\hat{C}$ is a subset of some refusal of w at $q \downarrow_w = s$, and second w behaves deterministically and therefore it has only one maximal refusal which implies that refusals of w are closed under unions.

(II) $\mu_d(u.(w \oplus v))(q) = 1$. Then by condition (C1) the only possibility for divergence is when u has infinite traces hidden to $(w \oplus v)/q$ and therefore to w/s which proves that $u.w$ diverges at s. This completes the proof of (2.13) for the special case with equality .

Finally we present an example where (2.16) holds with strict inequality. Take $trw := \{< >,<a>\}$; $trv := \{< >,<b>\}$ where $trW \ \{a^*\}$; $trV = \{ba^*\}$ and assume that all processes w,v,W,V behave deterministically. Then $\mu_r(w + v)(< >) = \varnothing$ whereas

$$\mu_r((w + v).w + (w + v).v)(< >) = 2^{\{a\}}$$

This follows since v can make in $(w + v).w$ a hidden $<b>$ transition upon which it also locks w to inhibit an $<a>$ transition by definition of V above , so that

$$\mu_r((w + v).w)(<>) = 2^{\{a,b\}}$$

and the rest follows easily.

Proof of Fact 2.5

(1) Straightforward and omitted.

(2) First we prove that $S(u).U \leq u$ on a trace s shared by $S(u).U$ and $u \in [U]$. Let $\mu_d u(s) = 1$ then by definition there exists $t \in trX$ such that $t \downarrow_U = s$ and

$$\mu_d S(u)(t) = max(\mu_d u(s) , \mu_d X(t)) = 1$$

Proof of Fact 2.5

(1) Straightforward and omitted.

(2) First we prove that $S(u).U \le u$ on a trace s shared by $S(u).U$ and $u \in [U]$. Let $\mu_d u(s) = 1$ then by definition there exists $t \in trX$ such that $t\downarrow_U = s$ and

$$\mu_d S(u)(t) = max(\mu_d u(s) , \mu_d X(t)) = 1$$

which implies by condition (2) of divergence for projection

$$\mu_d(S(u).U)(s) = 1$$

Now let $D \in \mu_r u(s)$, then by definition

$$(D \cap B_s) \bigcup F \in \mu_r S(u)(t)$$

where $t \in trX$, $t\downarrow_U = s$, $F \in \mu_r X(t)$ and B_s is the set of next transitions of U/s . In particular one can choose t such that there is no a with the property $(t\hat{\ }<a>)\downarrow_U = s$, for unless this is possible we can indefinitely paste invisible transitions to a given t which implies that $\mu_d(S(u).U)(s) = 1$ and we are then done since

$$D \in \mu_r(S(u).U)(s) = 2^A$$

by definition. Thus with this special t we must have by the definition of refusals for projection

$$((D \cap B_s) \bigcup F \bigcup C) \in \mu_r(S(u).U)(s)$$

where $C \in \mu_r U(s)$ is arbitrary. Choosing $F = \emptyset$ and $C = ((A \setminus B_s) \cap D)$ we obtain

$$(D \cap B_s) \bigcup F \bigcup C = D$$

which proves that $D \in \mu_r(S(u).U)(s)$.

We now prove the reverse relations given that $u \in [X.U]$. Let $\mu_d(S(u).U)(s) = 1$. Then bearing in mind the assumption that U behaves deterministically two possibilities exist :

(i) The process $S(u)$ has an infinite trajectory transparent to U/s . But this implies that for some $q \in trX$ with $q\downarrow_U = s$, X/q has an infinite trajectory transparent to U/s . This implies that $\mu_d(X.U)(s) = 1$ and if $u \in [X.U]$ then $u \le X.U$ so that $\mu_d(X.U)(s) \le \mu_d u(s)$ and therefore $\mu_d u(s) = 1$.

(ii) For some $t \in trS(u)$ with $t\downarrow_U = s$, $\mu_d S(u)(t) = 1$. Then either $\mu_d u(t\downarrow_U) = 1$, in which case we are done , or $\mu_d X(t) = 1$ which implies by condition (2) for divergence in projection that $\mu_d(X.U)(s) = 1$. Again $u \in [X.U]$ implies the result.

We now prove the reverse result for refusals. Let $D \in \mu_r(S(u).U)(s)$ then there either exists $t \in trS(u)$ with $t\downarrow_U = s$ and $B \in \mu_r S(u)$ where $(t\hat{\ }<a>)\downarrow_U = s \Rightarrow a \in B$ such that

$$D = B \bigcup C$$

for some $C \in \mu_r U(s)$; or $\mu_d(S(u).U)(s) = 1$. If the latter is valid then by the previous result $\mu_d u(s) = 1$ and therefore $D \in \mu_r u(s) = 2^A$. Therefore we assume the existence of t with the alleged properties . Then by definition of $S(u)$ the set B is given by

$$B = (K \cap B_s) \bigcup F$$

for some $K \in \mu_r u(s)$ and $F \in \mu_r X(t)$ where , as before , B_s is the next event transition set of U/s . First observe that

$$F \cup C \in \mu_r(X.U)(s)$$

by definitions of F and C . In order to justify this observation we need to show that for the t above $(t\hat{}<a>)\downarrow_U = s$ where $t\hat{}<a> \in trX$ implies that $a \in F$. But by the property of t established above $a \in B$ and by the definition of B above we must have $a \in F$ since a cannot be a member of B_s (recall $<a>\downarrow_{U/s} = < >$). Therefore it remains to show that

$$(K \cap B_{s)} \cup (F \cup C) \in \mu_r u(s)$$

given that $u \in [X.U]$ and it behaves deterministically. Recall that for deterministically behaving signals there is a single maximal refusal and therefore refusals are closed under union operation. But $K \in \mu_r u(s)$ and therefore $K \cap B_s \in \mu_r u(s)$ since the $K \cap B_s$ is a subset of K . Also $(F \cup C) \in \mu_r u(s)$ by the assumption $u \in [X.U]$ and thus the result follows by the closure property of union as stated above.

Proofs of (2.17) , (2.18) and the rest are omitted .

Proof of Theorem 3.1

The proof in one direction is trivial since the response function is continuous and has the *lub* property. Therefore it remains to prove the existence of a realization for the given response function.

We construct a pre-state function $\hat{S}(u)$, defined on $[V]$, taking values in $[V \oplus Y]$, by using induction on its domain. The state process X is then defined as the limit $\hat{S}(V\uparrow n)$ as $n \rightarrow \infty$.

We construct $\hat{S}$ on each finite trajectory on $[V]$ by using induction on the length of the trajectory and use the *lub* property of G for the extension to arbitrary processes on $V\uparrow n$. We use the notation ' $\leq$ ' for the order on traces [13] whereas '$\underline{\leq}$' is used for the partial order on processes .

We start by defining $\hat{S}$ on the trajectory of length zero , namely the null process *HALT* in $[V]$ as

$$tr(\hat{S}(HALT)) := trG(HALT)$$
$$\mu\hat{S}(HALT)(s) := (\mu U(<>), \mu G(HALT)(s))$$

Now set the induction hypotheses as follows

(1) (Input Projection) : $\hat{S}(u).U = u$

(2) (Output Projection) : $\hat{S}(u).Y = G(u)$

(3) (Continuity) : $\hat{u} \leq u \Rightarrow \hat{S}(\hat{u}) \tilde{\underline{\leq}} \hat{S}(u)$

(4) (Consistency) : $\hat{u} \leq u \Rightarrow \hat{S}(\hat{u}) \leq S_D(\hat{u})$ where $D = (U, \hat{S}(u), Y)$ and if $p \in trS_D(\hat{u})$ then there exists $p' \in tr\hat{S}(\hat{u})$ such that

$$p = p'\hat{}\,w$$
$$p\downarrow_U = p'\downarrow_U\hat{}\,w$$

We shall prove these hypotheses for all $u \in [V]$. Initially we prove this for trajectories in $[V]$ using induction on the length . We assume that the hypotheses (1)-(4) hold for all trajectories u of length $n-1$ in the projected domain $[V]$ and prove their validity for trajectories of length n. It can be verified using the construct for $\hat{S}$ on the null input above that (1)-(4) are valid when $n = 0$.

Let $s\hat{}<a>$ be the maximal trace of the trajectory u and let u' denote the trajectory of length $n-1$ with the maximal trace s .

[13] $s \leq t$ means s is a prefix of t .

Let $\hat{R}_u$ denote the set of all triples (r,q,p) that satisfy :

$$\begin{array}{l} r \in trG(u) \\ q \le r \ ; \ q \in trG(u') \\ p \in trS(u') \ ; \ p\downarrow_Y = q \end{array} \qquad \text{(A6)}$$

and define the prefix closed set $\hat{\rho}_u(r,q,p)$ for each $(r,q,p) \in \hat{R}_u$ as

$$\hat{\rho}_u(r,q,p) := (z \in A^* \mid z \le p\hat{}w\hat{}<a>\hat{}\tau \ ; \ p\downarrow_U \hat{}w = s \ ; \ q\hat{}\tau = r\ ) \qquad \text{(A7)}$$

and let

$$\rho_u := \bigcup_{(r,q,p)\in R_u} \hat{\rho}_u(r,q,p) \qquad \text{(A8)}$$

where R_u is <u>any</u> subset of $\hat{R}_u$ which satisfies the condition

$$[r \in trG(u) \ ; \ r \notin trG(u')] \Rightarrow (r,q,p) \in R_u \ \text{ for some feasible } q,p \qquad \text{(A9)}$$

We define the pre-state function on u by defining its traces as below

$$tr\hat{S}(u) := \rho_u \cup tr\hat{S}(u') \qquad \text{(A10)}$$

Observe that the choice of R_u is left arbitrary except for the condition (A9) . For example the set $\hat{R}_u$ satifies (A9) and therefore can be taken as a possible R_u [14] .

We claim that with the extension of $\hat{S}$ given by (A10) the induction hypotheses (1) to (4) above hold for n . The proof of hypothesis (3) is obvious by the definition (A10). We prove the remaining below. Throughout we shall use the assumption that the input and output signals are event decoupled (*ed*).

(1) Let $t' \in tr(\hat{S}(u).U)$, then there is $t \in tr\hat{S}(u)$ such that $t' = t\downarrow_U$. If $t \in \hat{S}(u')$ we are done by hypothesis (1) since

$$t' = t\downarrow_U \in tru' \subseteq tr(u)$$

Otherwise $t \in \hat{\rho}_u(r,q,p)$ for some $(r,q,p) \in R_u$. Therefore $t \le p\hat{}w\hat{}<a>\hat{}\tau$ and by definitions (A6) and (A7) using the *ed* assumption

$$t\downarrow_U \le s\hat{}<a>$$

and therefore $t' \in tru$.

Conversely let $t \in tru$. If $t \in tru'$ we are done by hypothesis (1) and (3) since

$$t \in tr(\hat{S}(u').U) \subseteq tr(\hat{S}(u).U)$$

Otherwise $t = s\hat{}<a>$. Let $r \in trG(u)$ and $r \notin trG(u')$ then by definition of choice of R_u constrained by (A9) there exists (r,p,q) that satisfies (A6). The fact that such p and q exist is guaranteed by induction hypotheses (1) and (2). Construct the trace v as

$$v := p\hat{}w\hat{}<a>\hat{}\tau$$

where w is defined to complement $p\downarrow_U$ to s as in (A7) and τ satisfies $q\hat{}\tau = r$. We have $v \in \hat{\rho}_u(r,q,p)$ and therefore $v \in tr\hat{S}(u)$ by (A10) and the fact that $(r,q,p) \in R_u$. Therefore since

$$v\downarrow_U = p\downarrow_U\hat{}w\hat{}<a> = s\hat{}<a> = t$$

that follows from *ed* assumption we have $t = v\downarrow_U \in tr(\hat{S}(u).U)$.

[14] Different choices of $\hat{R}(u)$ lead to different buffering strategies . Avoidance of nondeterministic behaviour is also related to this choice (see the main text and the proof of Theorem 3.2 for details).

(2) Let $t' \in tr(\hat{S}(u).Y)$ then for some $t \in tr\hat{S}(u)$, $t'=t\downarrow_Y$. If $t \in \hat{S}(u')$ then we are done by hypothesis (2) since

$$t'=t\downarrow_Y \in trG(u') \subseteq trG(u)$$

using the continuity of G. Otherwise $t \in \rho_u$ and for some $(r,q,p) \in R_u$ we have $t \in \hat{\rho}_u(r,q,p)$. Hence $t \leq p\hat{}w\hat{}<a>\hat{}\tau$ and $t\downarrow_Y \leq q\hat{}\tau = r$ by definitions (A6) and (A7) and therefore $t' \in trG(u)$. This proves that $\hat{S}(u).Y \leq G(u)$.

Conversely let $r \in trG(u)$. If $r \in trG(u')$ we are done by induction hypothesis , else $r \notin trG(u')$ and by using the fact that $(r,q,p) \in R_u$ for some q and p by constraint (A9) - again existence of such q and p are guaranteed by the induction hypothesis - we utilize the same construction of v as in the proof of (1) above and show that $r \in tr(\hat{S}(u).Y)$.

(4) Let $v \in tr\hat{S}(\hat{u})$ then by induction hypothesis (3) $v \in tr\hat{S}(u)$. But by construction of $\hat{S}(\hat{u})$, $v\downarrow_U \in tr\hat{u}$ and so by definition $v \in trS_D(\hat{u})$.

Let $p \in trS_D(\hat{u})$ then by definition of D , $p \in tr\hat{S}(u)$ and $p\downarrow_U \in tr\hat{u} \subseteq tru$. If $p \in tr\hat{S}(u')$ we are done by induction hypothesis since then $p \in S_{D'}(\hat{u})$ where $D' := (U,\hat{S}(u'),Y)$. Otherwise unless $\hat{u} = u$ which is a trivial case $p \leq \hat{p}\hat{}w$ by (A7) where $\hat{p} \in tr\hat{S}(u')$. If $\hat{u} = u'$ we are done , else apply the induction hypothesis (4) to $\hat{p}$ using D' as above and the result follows.

We have proved by induction that (1) to (4) are valid for all n . Now for any $v \in [V\uparrow n]$ we can write

$$v := \bigsqcup_{T \leq v} T$$

where each T is a trajectory . Define

$$\hat{S}(v) = \hat{S}(\bigsqcup_{T \leq v} T) := \bigsqcup_{T \leq v} \hat{S}(T)$$

which defines $\hat{S}$ on $[V\uparrow n]$. It is straightforward to show that hypotheses (1)-(4) apply to this extended domain $[V\uparrow n]$ by using the *lub* property of the G function . We define $\hat{S}(v)$ for arbitrary v as the chain limit of $\hat{S}(v\uparrow n)$ and choose $X := \hat{S}(V)$.

We next prove that hypothesis (2) holds when $\hat{S}$ is replaced by S_D where $D := (U,X,Y)$. We use hypothesis (4) in the limit case to prove this. By (4) we have for any $v \in [V]$, $\hat{S}(v) \leq S_D(v)$, therefore it is sufficient to show that

$$S_D(v).Y \leq \hat{S}(v).Y$$

Let $s \in tr(S_D(v).Y)$ then there exists $t \in trS_D(v)$ such that $t\downarrow_Y = s$ .Using (4) there exists $p \in tr(\hat{S}(v))$ such that $t = p\hat{}w$ and $t\downarrow_U = p\downarrow_U\hat{}w$. But this implies that all the transitions in w are input ones by *ed* assumption and $t\downarrow_Y = p\downarrow_Y = s$ and we are done.

Finally applying (1) in the limit we have $\hat{S}(V).U = X.U = V$.

Proof of Theorem 3.2

($\Rightarrow$) We show using the existence of R with the given properties that G satisfies the requirements stated in the theorem . The fact that G behaves deterministically on T follows from conclusions (2) and (3) of Theorem 3.2 . By the definition of a realization of G we have $\delta(X.U) = \delta V$ and $\delta(H_R(v)) = \delta G(v)$ for all $v \in [V]$ and hence using Fact 2.5 and the definition of δG , δR is a realization of δG. Therefore by (3.2) δG must be continuous and must satisfy the *lub* property.

We show that G is stable relative to $(T,G(V))$ by contradiction. First suppose that G is not bounded from above. Then there is a trajectory $v \in V^\delta$ and integer n such that the signal $G(v\uparrow n)$ has traces of unbounded length. More precisely there exist an infinite sequence of traces $\{s_i\}$ such that for all integers i :

$$s_i \in tr(G(v \uparrow n))$$
$$\# s_i \geq i$$

But since by hypothesis

$$G(v \uparrow n) = H_R(v \uparrow n) = S_R(v \uparrow n).Y$$

there exists another infinite sequence of traces $\{t_i\}$ such that

$$t_i \in tr(S_R(v \uparrow n))$$
$$\# t_i \geq i$$
$$t_i \downarrow_Y = s_i$$
$$t_i \downarrow_U \in tr(v \uparrow n)$$

On the other hand the signal $v \uparrow n$ has finite number of traces and hence there is a subsequence $\{t_{k_i}\}$ and a trace $\tau \in tr(v \uparrow n)$ such that

$$t_{k_i} \downarrow_U = \tau$$

for all i. But this implies that [15]

$$\mu_d(X.U)(\tau) = 1$$

which clearly contradicts the conclusion (1) that $X.U = V$ since V is assumed to behave deterministically.

The proof that G must be bounded from below follows similar arguments , therefore it suffices to summarize an outline. Assume that it is not bounded from below then there exists in $G(V)$ a $y \in Y^\delta$, an integer n and $v \in V^\delta$ such that v has unbounded traces where

$$G(v) = y \uparrow n$$

We can take v to be an infinite trajectory without loss of generality using the construction given by the proof of Lemma 2.1 . By hypothesis $G(v)= S_R(v).Y$ and therefore there exists infinite sequence of traces s_i of v and another sequence t_i of $S_R(v)$ that satisfy similar requirements to the previous case . This argument similarly leads to the conclusion that for some $\tau \in trY$ the process $S_R(v).Y$ diverges at trace τ , in other words

$$\mu_d H_R(v)(\tau) = 1$$

which violates the condition that $H_r(v) \in Y^\delta$ for $v \in V^\delta \cap T$. This completes the argument that G is stable .

($\Leftarrow$) We use the construction given in the proof of Theorem 3.1 for a deterministic realization of δG with domain δV. Let $\hat{R} := (\delta U, \hat{X}, \delta Y)$ denote the deterministic realization of δG constructed according to the proof of Theorem 3.1 then we choose $R = (U, X, Y)$ as the nondeterministic image of $\hat{R}$, namely $R := \eta \hat{R} = (U, \eta \hat{X}, Y)$. Considering the reverse inequalities that follow from (2.20) we have to show under the hypotheses on G that :

$$\eta(\hat{X}.\delta U) \leq X.U$$

[15] For a construction of an infinite hidden trajectory see the statement and proof of Lemma 2.1 above.

and

$$\eta(H_{\hat{R}}(\delta v)) \leq H_R(v)$$

whenever v is a trajectory for the latter inequality . In other words it is enough to show that $X.U$ behaves deterministically and H_R behaves deterministically on $V^\delta \cap T$. We do this in two steps . In the first step we show that the refusals behave deterministically. By this we mean that at any stage the refusal sets cannot contain events that can generate transitions. In the second step we show that the processes in question do not diverge .

Before proceeding with the details of the proof we first make explicit our choice of buffering for the specific realization we use for the proof. We choose $R_u = \hat{R}_u$, i.e., the realization that corresponds to infinite buffering in using the procedure of Theorem 3.1 to synthesize δG . Note that under this choice hypothesis (4) for $u = V$ in the proof of Theorem 3.1 can be replaced by the equality $S_D(\hat{u}) = \hat{S}(\hat{u})$, namely the pre-state function coincides with the actual state function.

Now let $s \in tr(X.U)$. We demonstrate that any event a such that $<a>$ belongs to $tr(X.U)/s$ cannot belong to any of the refusal sets in $\mu_r(X.U)(s)$. Noting that X is a process that behaves deterministically ($X := \eta\hat{X}$) it is enough to prove the following statement :

> Suppose that $s \hat{} <a> \in tr(X.U)$ then for each $t \in trX$ such that $t\downarrow_U = s$ there exists $t \hat{}\ t' \in trX$ such that
>
> $$(t \hat{}\ t')\downarrow_U = s \hat{}\ <a>$$

Let u denote the input trajectory with the maximal trace $s \hat{} <a>$ and let $\gamma := t\downarrow_Y$. By definition $t \in trS_R(u')$ and by hypothesis (4) in the proof of Theorem 3.1 under infinite buffering $t \in tr(\hat{S}(u'))$, where u' denotes the input trajectory with the maximal trace s . It also follows that $\gamma = t\downarrow_Y \in trG(u')$. Now let r be any trace of $G(u)$ such that γ is a prefix of r. Then $R_u = \hat{R}_u$ by the infinite buffer selection as stated above and we have $(r,\gamma,t) \in R_u$. Therefore $t\hat{}< >\hat{}<a>\hat{}\tau \in tr\hat{S}(u)$ by definition (A9) and (A10) where $\gamma \hat{}\ \tau = r$ and the result follows with $t' := <a>$

Next we prove that for each trajectory $v \in V^\delta$, refusals of $H_R(v)$ behave deterministically . In other words we show that if $<a>$ belongs to $tr(H_R(v)/s)$ then a cannot be a member of any refusal set in $\mu_r(H_R(v))(s)$. We prove this by using induction on the length of the input trace v. When v is of length zero result follows trivially by definition of $\hat{S}(HALT)$ in the proof of Theorem 3.1 . Assume it holds for $n = \#s$ where $s \hat{} <a>$ denotes the maximal trace of trajectory v. It suffices to prove the following statement also using the fact that $S_R(u)$ behaves deterministically by (2.20) .

> Suppose that $q\hat{}<b> \in trH_R(v)$ then for each $t \in trS_R(v)$ such that $t\downarrow_Y = q$ there exists $t \hat{}\ t' \in trS_R(v)$ such that
>
> $$(t \hat{}\ t')\downarrow_Y = q \hat{}\ <b>$$

Since $t \in trS_R(v) = tr\hat{S}(v)$ it can be written as

$$t \leq p \hat{}\ w \hat{}\ <a> \hat{}\ r$$

where $p \in tr\hat{S}(v')$, $p\downarrow_Y \hat{}\ r = q$ and $p\downarrow_U \hat{}\ w = s$ and v' is the trajectory with the maximal trace s . But $q\hat{}<b> \in trG(v)$ is given , hence by definition

$$(q\hat{}<b>, p\downarrow_Y, p) \in \hat{R}_v = R_v$$

since $p\downarrow_Y$ is a prefix of q and $p \in tr\hat{S}(v')$. This implies that

$$t \leq p \hat{}\ w \hat{}\ <a> \hat{}\ r \hat{}<b> \in tr\hat{S}(v) = trS_R(v)$$

and construction of t' is obvious.

Finally we show that $X.U$ and $H_R(v)$ both do not diverge , the latter evaluated at any $v \in V^{\delta} \cap T$. We prove this by contradiction for each case.

Suppose $X.U$ diverges at s then since X behaves deterministically by definition there exists $q \in trX$ such that $q \downarrow_U = s$ and X/q has infinitely many traces with null projection on U/s. But this violates the condition that G is bounded from above on T since for $v = u$ where u is the deterministic behaving trajectory with the maximal trace s , $G(u) = H_R(u) = S_R(u).Y$ has traces of unbounded length by the event decoupling assumption and the fact that $X \in [U \oplus Y]$.

Now assume that $H_R(v)$ diverges at some s for some $v \in V^{\delta} \cap T$. Then because $S_R(v)$ behaves deterministically we have for some $q \in trS_R(v)$ such that $q \downarrow_Y = s$, $S_r(v)/q$ has infinitely many traces that have null projections on Y/s. But this violates the condition that G is bounded from below on $G(V)$ by taking y as the trajectory with the maximal trace s , since projecting the unbounded traces on $S_R(v)/q$ on $U/(q \downarrow_U)$ we again get unbounded set of traces on the input $v/(q \downarrow_U)$ by the event decoupling assumption on the input and output transitions. This completes the proof of Theorem 3.2 .

Proof of Fact 3.1

Proof of (1) is trivial. To prove (2) we shall be content with proving that if R is not strongly nondeterministically equivalent to $\hat{R}$ then they cannot be bisimulation equivalent. The remaining details are routine.

We assume without loss of generality that R and $\hat{R}$ are strongly equivalent , otherwise condition (1) of bisimulation is violated by choosing a trace in $X.(U \oplus Y)$ which is not in $\hat{X}.(U \oplus Y)$ and we are done. Then there are two ways in which strong nondeterministic equivalence can be violated as explained below (each process is assumed to be replaced by its image under η below without a change in the notation) .

(1) For some s , $\mu_d(X.(U \oplus Y))(s) = 1$ whereas $\mu_d(\hat{X}.(U \oplus Y))(s) = 0$. Then since U , X and Y are assumed to behave deterministically there exist $t \in trX$ where $t \downarrow_{(U \oplus Y)} = s$ and X/t has infinitely many increasing traces t_j invisible to $(U \oplus Y)/s$. Now suppose that q_j are the corresponding increasing traces in $\hat{X}$ according to the definition of bisimulation. Then it must be true that $q_j \downarrow_{(U \oplus Y)} = s$ for all j and $\#(q_j)$ must be unbounded by conditions (1) and (2) of bisimulation . But this violates the hypothesis that the latter process is non-divergent at s .

(2) For some s , $B \in \mu_r(\tilde{X}.(U \oplus Y))(s)$ whereas $B \notin \mu_r(\hat{X}.(U \oplus Y))(s)$. We can assume , without loss of generality , both processes to be nondivergent at s by using (1) above . Then there exists $t \in trX$ such that $t \downarrow_{(U \oplus Y)} = s$ and $t\hat{\ }r\hat{\ }<a> \notin trX$ for all $a \in B$ and all r where $(t\hat{\ }r)\downarrow_{(U \oplus Y)} = s$. On the other hand for all $p \in tr\hat{X}$ such that $p \downarrow_{(U \oplus Y)} = s$ there exists at least one event $b \in B$ such that $p\hat{\ }q\hat{\ }<b> \in tr\hat{X}$ for some q with the property $(p\hat{\ }q)\downarrow_{(U \oplus Y)} = s$. We claim that R/t cannot be bisimulation equivalent to $\hat{R}/p$ for any choice of p above . We demonstrate this by choosing the trace $q\hat{\ }<b> \in \hat{X}/p$. Clearly X/t cannot have a corresponding trace satisfying condition (1) of bisimulation because of the above property of t .

REFERENCES

[1] K. Inan " Asynchronous Dynamical Systems I ", ERL Mem. No. UCB/ERL M89/59 , 17 May 1989 .

[2] K. Inan , P. Varaiya , " Finitely Recursive Process Models for Discrete Event Systems " , *IEEE Trans. Automat. Contr.* , vol.AC-33, No.7, pp. 626-639 , July 1988.

[3] K. Inan , P. Varaiya , "Algebras of Discrete Event Models " ,*IEEE Proc.* vol.77 , No.1 , pp. 24-38 , Jan. 1989 .

[4] C.A.R. Hoare , *Communicating Sequential Processes* , Herts, England: Prentice-Hall International, 1985 .

[5] R. Milner , " Lectures on a Calculus for Communicating Systems " , in *Control Flow and Data Flow: Concepts of Distributed Programming* , (M. Broy ed.) , NATO ASI Series F : Vol. 14 Springer Verlag , 1985 , pp. 205-228 .

[6] P. De Nicola , M. Hennessy , "Testing Equivalences for Processes ", *Theoretical Computer Science 34* (1984) , pp. 83-133 .

[7] J.B. Dennis , "The Scenario Theory for Non-Determinate Computation" , in *Control Flow and Data Flow: Concepts of Distributed Programming* ,(M. Broy ed.) , NATO ASI Series F : Vol. 14 Springer Verlag , 1985 , pp. 382-398 .

Figure Captions

Figure 2.1 Example for nondeterministic behaviour

Figure 2.2 Reduced job-shop example for a nondeterministic ADS

Figure 3.1 Input and output spaces for the realization example

Figure 3.2 Unbuffered and buffered realizations

Figure 3.3 Example for complexity index

Figure 3.4 Input and output spaces for the nondeterministic realization example

Figure 3.5 Nondeterministically behaving realizations

Figure 3.6 A deterministically behaving realization

Figure 3.7 Example for different equivalence relations on ADS

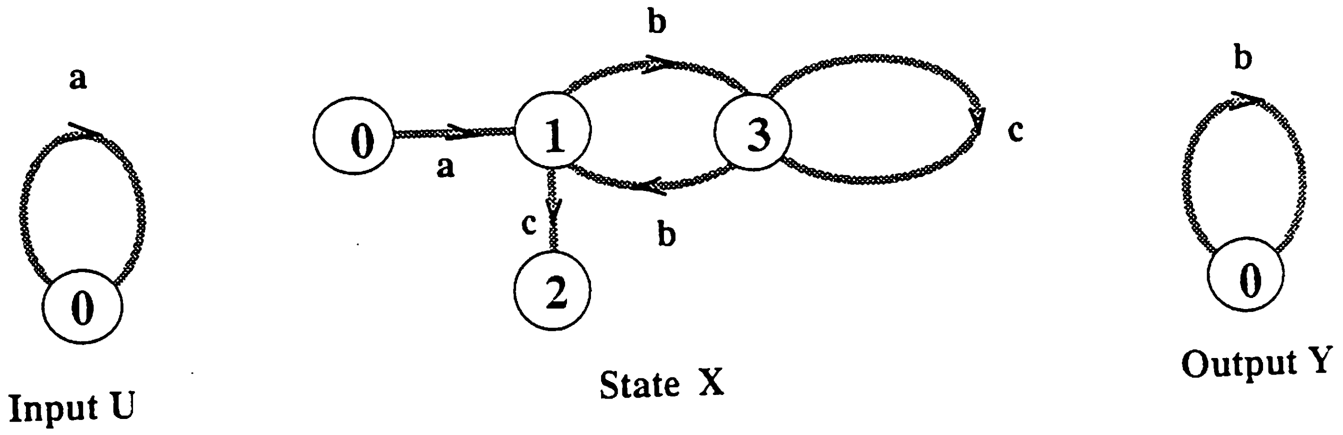
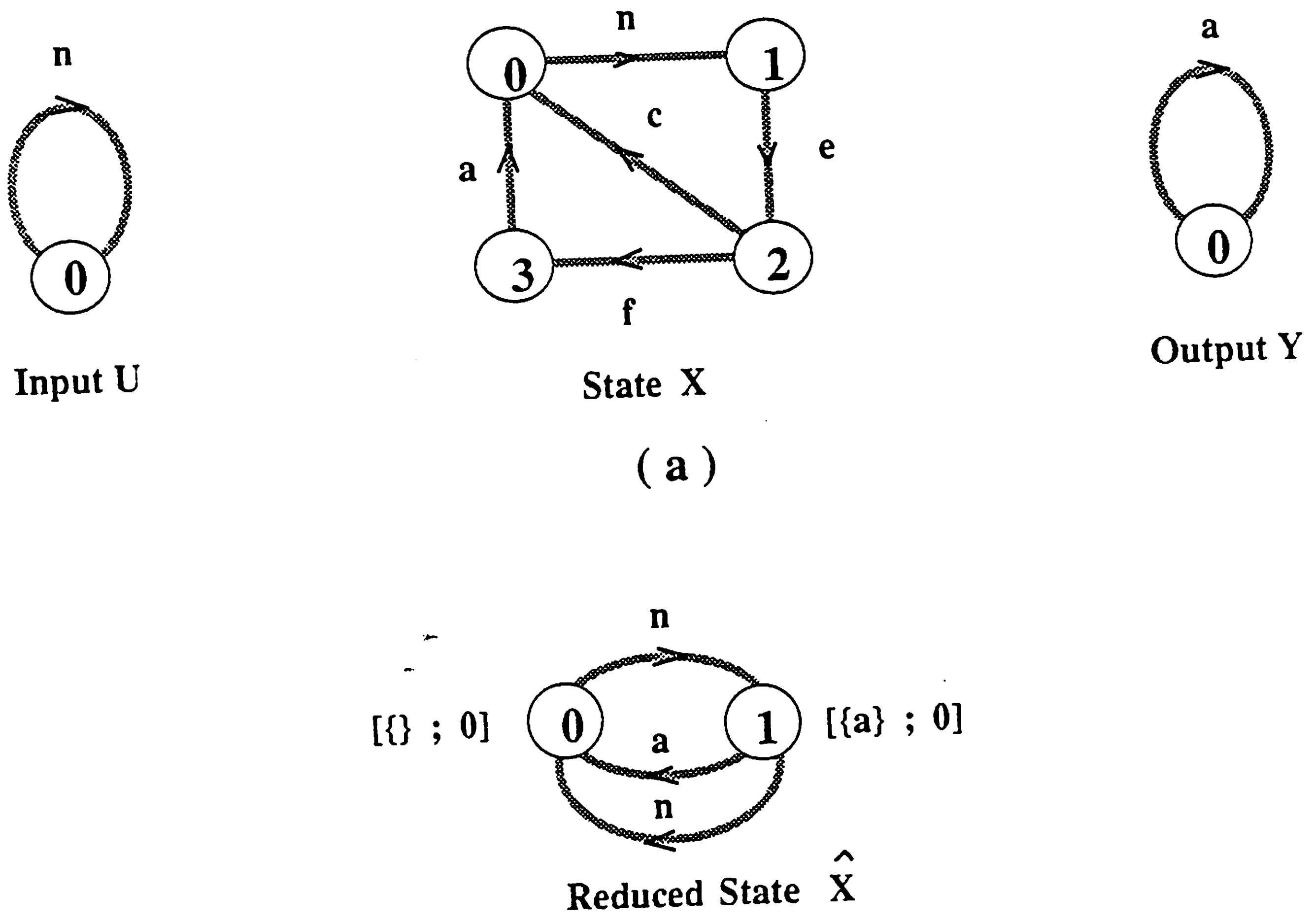


Figure 2.1



(a)

(b)

Figure 2.2 (a) & (b)

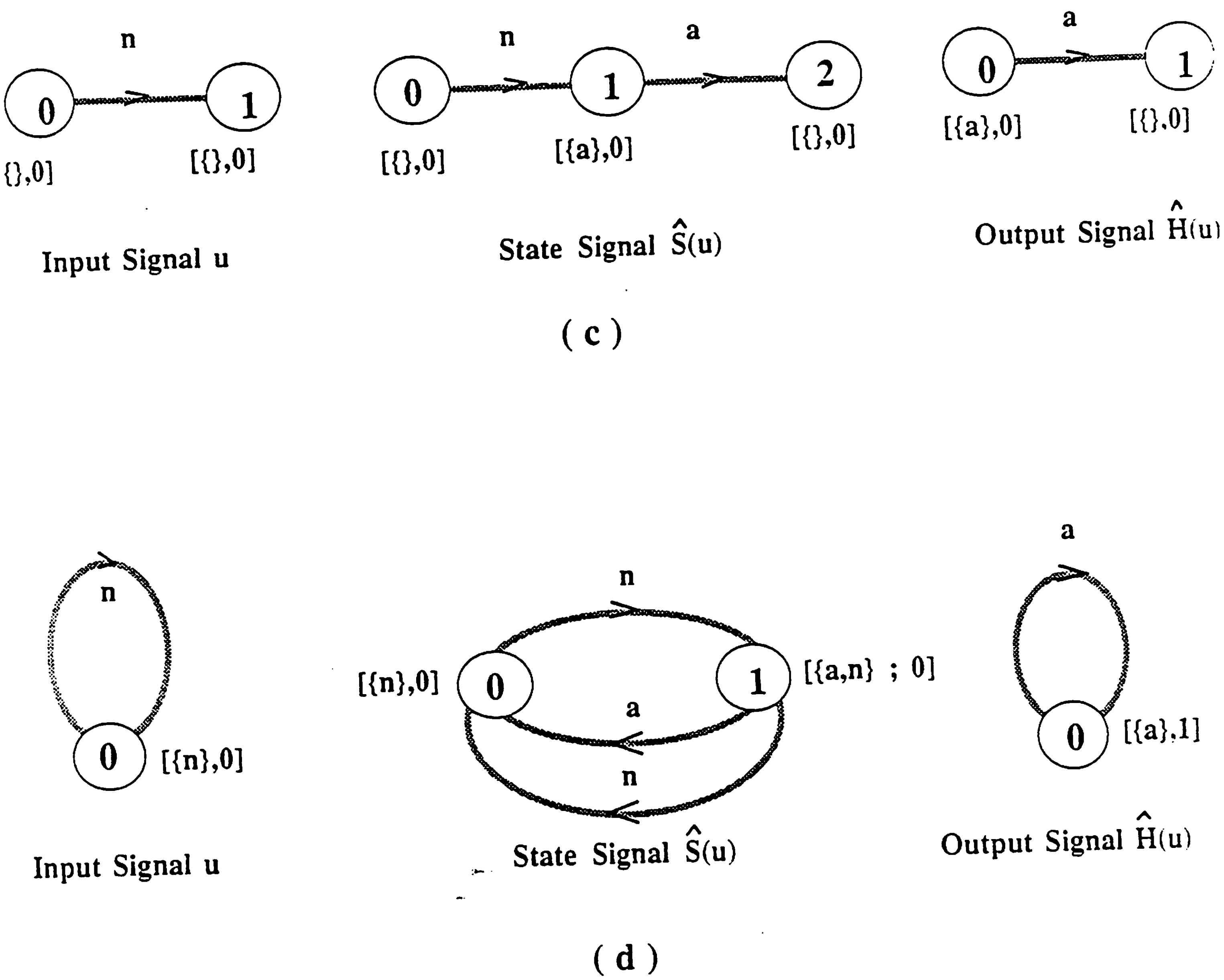
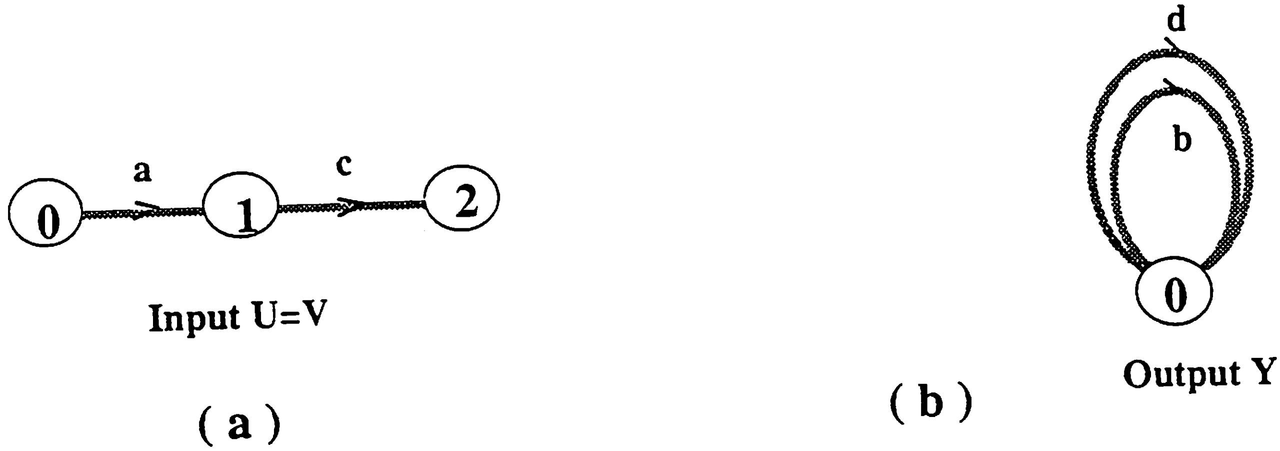

n
0
1
{},0]
[{},0]
Input Signal u
n
a
0
1
2
[{},0]
[{a},0]
[{},0]
State Signal $\hat{S}(u)$
a
0
1
[{a},0]
[{},0]
Output Signal $\hat{H}(u)$
(c)
n
0
[{n},0]
Input Signal u
n
a
n
[{n},0]
0
1
[{a,n} ; 0]
State Signal $\hat{S}(u)$
a
0
[{a},1]
Output Signal $\hat{H}(u)$
(d)

Figure 2.2 (c) & (d)



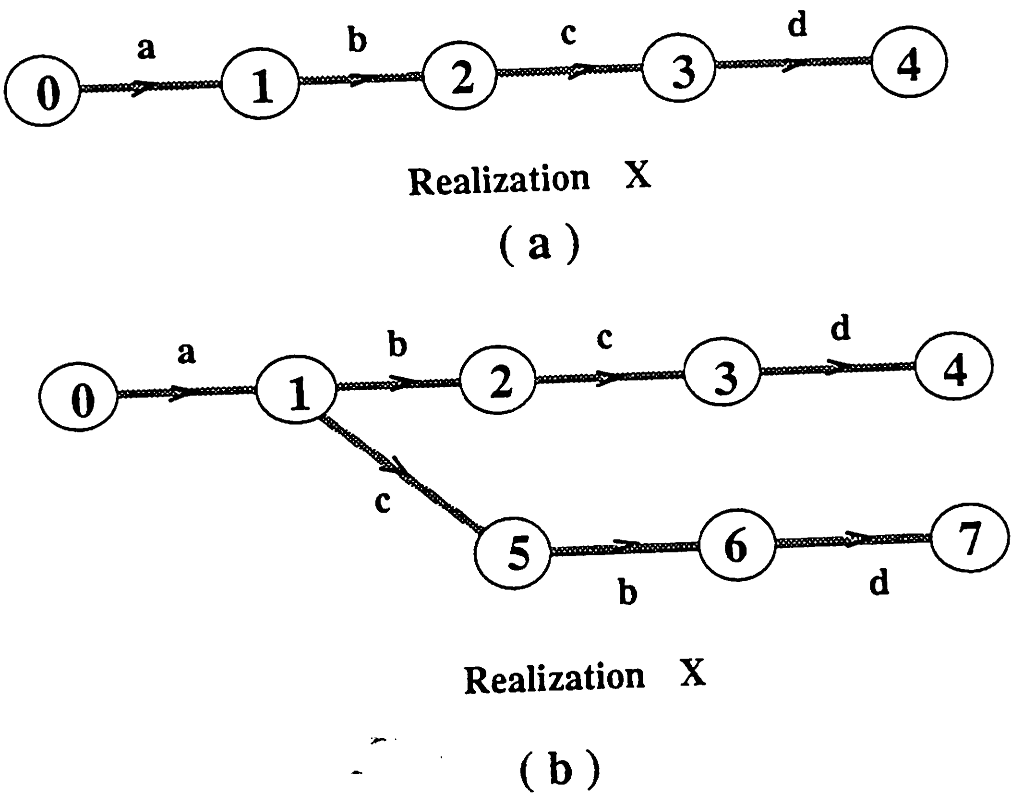
Input U=V

(a)

(b)

Output Y

Figure 3.1



Realization X

(a)

Realization X

(b)

Figure 3.2

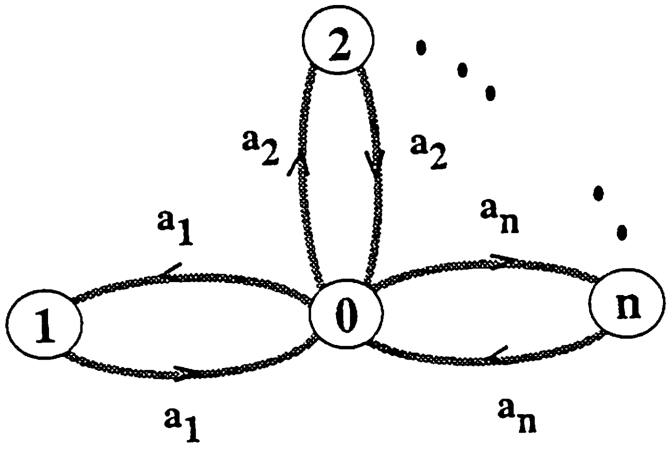
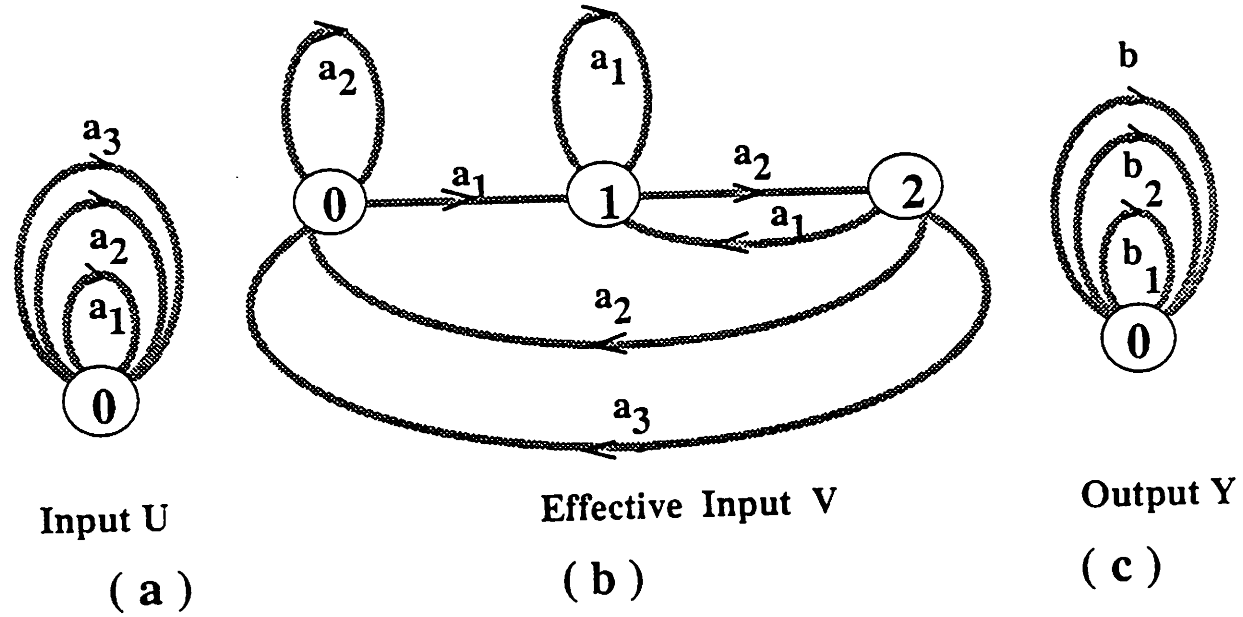


Figure 3.3



Input U

(a)

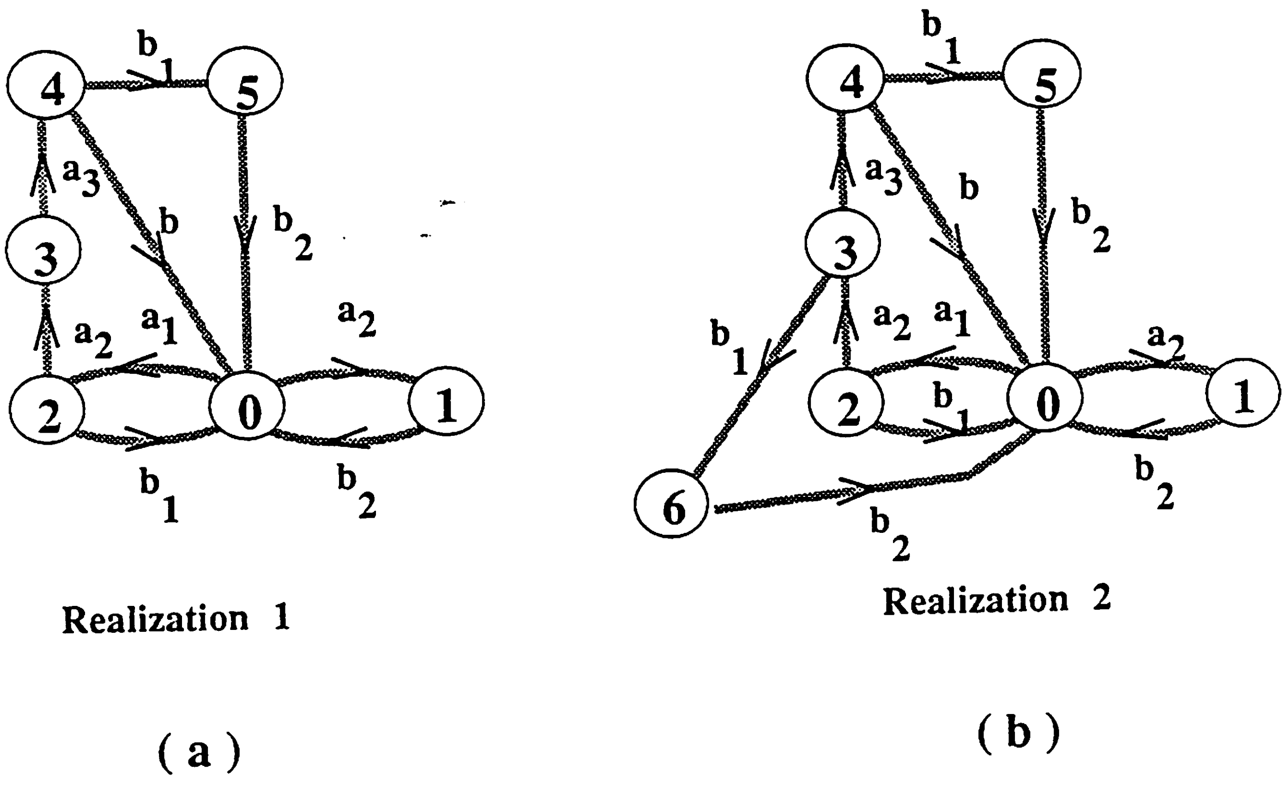
Effective Input V

(b)

Output Y

(c)

Figure 3.4



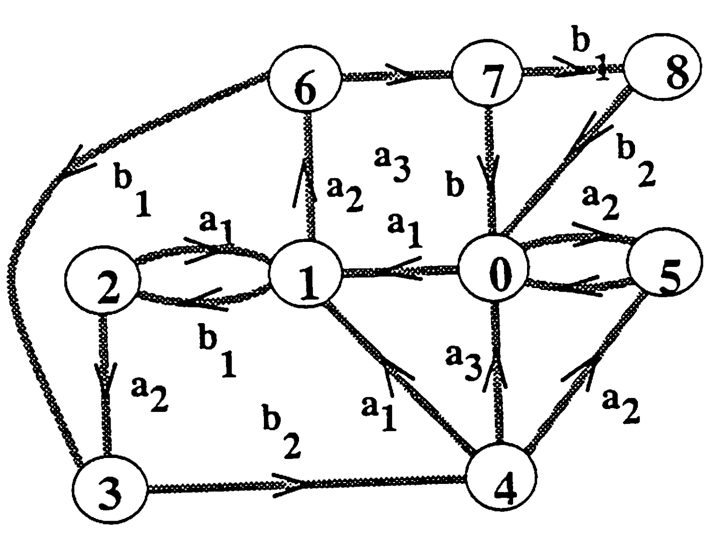
Realization 1

(a)

Realization 2

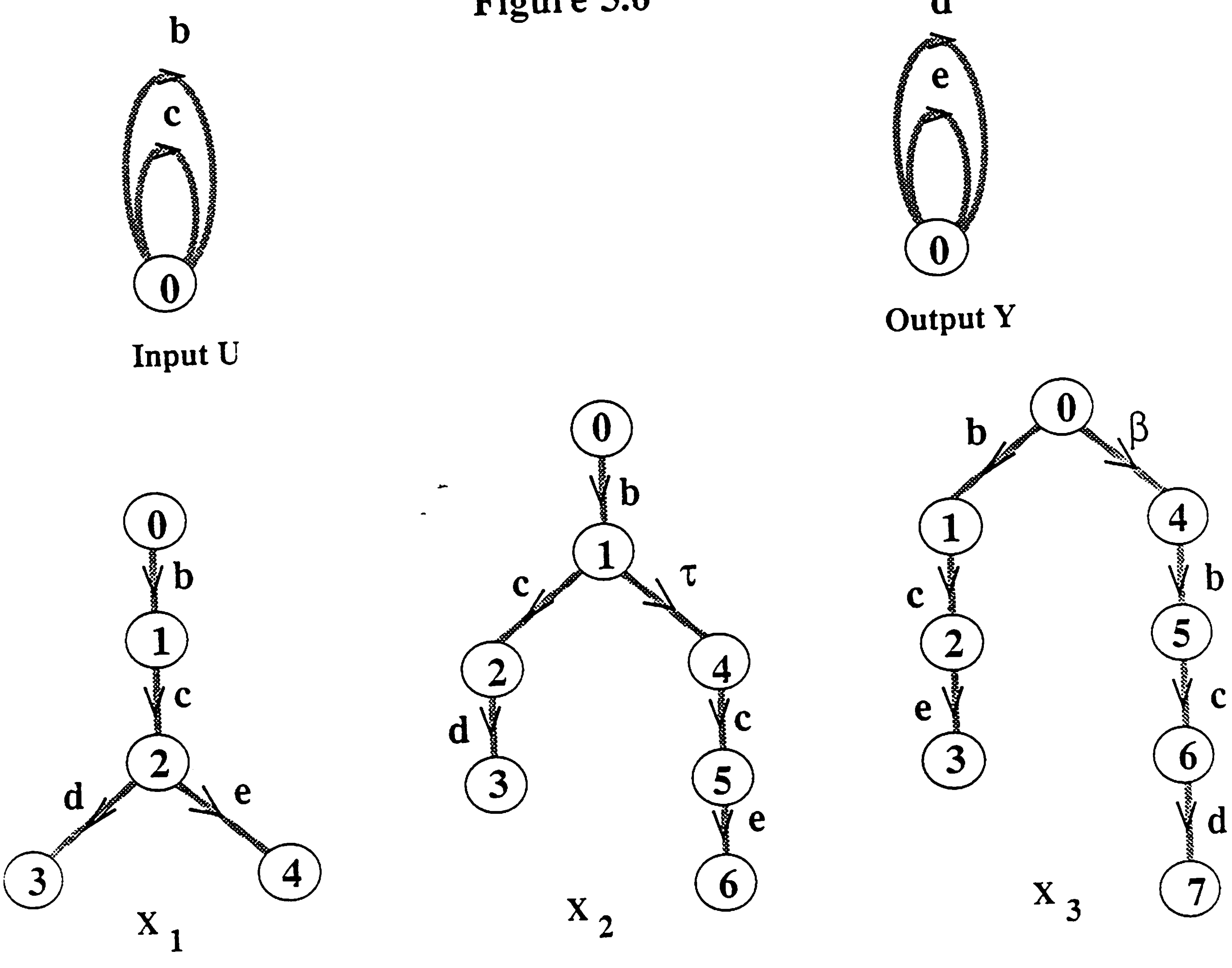
(b)

Figure 3.5



Deterministically Behaving Realization

Figure 3.6



Input U

Output Y

X 1

X 2

X 3

Figure 3.7