Chapter 8 :- Assigning Meanings to Models

So far in this thesis we have studied in some depth the mathematical structure of two models for computation. We have derived many interesting results which are in many cases correctness results for the model processes which exist within our two models. It is natural to ask how these models relate to any more "real" space of processes and how our correctness proofs translate into theorems of the more concrete processes.

Before examining our particular models it is interesting to develop as general a calculus as possible for this type of relative study. This should give us a better general understanding of the problem of modelling processes, and suggest alternative models with chosen properties. Let us therefore suppose that C is some space of concrete processes and that M is intended to be a mathematical model for C. We will for the moment make no assumptions about the natures of C and M. Since M is a model for C we may suppose that there is some function $\Phi: C \to M$ such that for any $c \in C$ $\Phi(c)$ is the representation in M of c.

Our particular interest is in the translation of proofs in M to results in C (i.e deducing results about real processes from the model). It is therefore very important to us to know how predicates translate. Suppose that Ψ is some predicate we want to prove of an element of C (Ψ is thus a one-place relation on C). It is clearly important that we should know just how much we need to prove of $\Phi(c)$ $(c \in C)$ in order to know for certain that $\Psi(c)$ holds. Define the predicate Ψ' on M by $\Psi'(A) \equiv \forall c.(\Phi(c) = A) \Rightarrow \Psi(c)$. By construction Ψ' is the weakest predicate on M which ensures this (i.e. $\Psi'(\Phi(c)) \Rightarrow \Psi(c)$). Correspondingly if Π is a predicate on M we can define the predicate Π^* on C by $\Pi^*(c) \equiv \Pi(\Phi(c))$. Π^* is the strongest predicate on C which is proved by $\Pi(\Phi(c))$. These two constructs relate in the following way.

8.1 Theorem

If $\Phi: C \to M$, Ψ is a predicate of C, Π is a predicate of M and the constructs "'" and "*" are as described above then we have the following.

a) $\Pi(A) \Rightarrow (\Pi^*)'(A)$ ($\Leftrightarrow$ if Φ is onto) for each $A \in M$

b) $\Psi(c) \Leftarrow (\Psi')^*(c)$ ($\Leftrightarrow$ if Φ is one-one) for each $c \in C$

c) $\Pi^* \equiv ((\Pi^*)')^*$

d) $\Psi' \equiv ((\Psi')^*)'$

proof

The following lemma is useful in proving the above (amongst other things).

8.1.1 lemma

a) If H and Θ are two predicates over M such that H is weaker than Θ (i.e. $\Theta(A) \Rightarrow H(A)$ for all $A \in M$) then H^* is weaker than Θ^* (i.e. $\Theta^*(c) \Rightarrow H^*(c)$ for all $c \in C$).

b) If X and Ω are two predicates over C such that Ω is weaker than X then Ω' is weaker than X'.

proof

Both these results follow immediately from the definitions of Ψ' and Π^*.

We can now prove a) - d) above.

a) $\Pi(A) \Rightarrow \Pi^*(c)$ for each c s.t. $\Phi(c) = A$ (by defn of Π^*)

$\Rightarrow (\Pi^*)'(A)$ (by defn of Ψ')

If Φ is onto then $A = \Phi(c)$ for some $c \in C$, so

$(\Pi^*)'(A) \Rightarrow (\Pi^*)'(\Phi(c))$

$\Rightarrow \Pi^*(d)$ for each d s.t. $\Phi(d) = \Phi(c)$ (by defn of Ψ')

$\Rightarrow \Pi^*(c)$ (as $\Phi(c) = \Phi(c)$)

b) $(\Psi')^*(c) \Rightarrow \Psi'(\Phi(c))$ (by defn of Π^*)

$\Rightarrow \Psi(d)$ for each d s.t. $\Phi(d) = \Phi(c)$ (defn of Ψ')

$\Rightarrow \Psi(c)$ (as $\Phi(c) = \Phi(c)$)

If Φ is one-one then

$\Psi(c) \Rightarrow \Psi(d)$ for each d s.t. $\Phi(c) = \Phi(d)$ (as there is only one such d (i.e. c))

$\Rightarrow \Psi'(\Phi(c))$ (by defn of Ψ')

$\Rightarrow (\Psi')^*(c)$ (by defn of Π^*)

c) $(\Pi^*)'$ is weaker than Π by (a) above. Thus $((\Pi^*)')^*$ is weaker than Π^* by 8.1.1(a). Π^* is weaker than $((\Pi^*)')^*$ by (b) above. These two clauses together give us the desired result.

d) Ψ is weaker than $(\Psi')^*$ by (b) above. This tells us that Ψ' is weaker than $((\Psi')^*)'$ by 8.1.1(b). Also $((\Psi')^*)'$ is weaker than Ψ' by (a) above. Again these two clauses

combine to give the desired result.

Note that 8.1 (a) and (b) tell us (as we might have expected) that if Φ is a bijection then also the predicate spaces have "'" = "*"$^{-1}$ as a bijection between them. We will however find that it is unusual for Φ to be either one-one or onto.

From the nature of the mapping $\Psi \to \Psi'$ we see that the more discriminating the map Φ the more likely we are to get a reasonable translation into M of a predicate Ψ. This is because there are less additional $d \in C$ which need to satisfy Ψ in order to make $\Psi'(\Phi(c))$ true for any c. It is clearly more important that a useful predicate should translate well than one which we are unlikely to want to prove of a process. We would like (for as many and as useful Ψ as possible) that there should be predicates Π of M s.t.

(i) $\Pi \Rightarrow \Psi'$

(ii) Π is reasonable to prove (is continuous, perhaps)

(iii) Π is not ridiculously strong, so that generally $\Pi(\Phi(c))$ does in fact hold of a process c s.t. $\Psi(c)$.

Note that everything we have said so far is equally true of product spaces and their predicates because of the extremely general nature of the spaces we have considered. Though we treat only single predicates in the following, for simplicity, almost the whole of the rest of what follows can be translated to accommodate predicates of several (or many) variables.

We must now be more specific about the nature of the objects we are studying. Let us suppose that elements of C have things called <u>behaviours</u>, which we will think of as representing possible sequences of actions carried out by processes (relative to any control exerted by outside agents). We will suppose that the behaviours of a process are a powerful language for specifying correctness; we will restrict ourselves to the study of predicates of the form $\underset{\sim}{\chi}(c) \Leftrightarrow \forall b.\ b$ is a behaviour of $c \Rightarrow \chi(b)$ (χ any predicate of behaviours). For $c \in C$ we will suppose that B(c) is the set of possible behaviours of c. Note that the type of predicate specified above corresponds quite

closely to normal ideas of what a correctness predicate of a process should be: a process is "correct" if and only if it must inevitably behave correctly.

We will further suppose that the map $\Phi: C \to M$ is closely based on the possible behaviours of a process. This assumption is prompted by the nature of the models we already possess, which are both intended to reflect some aspect of a process' behaviour. Specifically we will suppose that $\Phi(c) = \{ \theta(b) \mid b \in B(c) \}$ for some function θ. (For example in the "traces" model P of chapters 1-3 we would expect that $\theta(b)$ = the sequence of external communications occurring in b.) This assumption clearly carries with it the requirement that M should be (at least up to isomorphism) a subset of some powerset (a requirement which is met by both our existing models).

Clearly the more discriminating the function θ, the more expressive becomes the model.

For the predicate $\underset{\sim}{X}$ and function Φ described above we have

$$\begin{aligned}
\underset{\sim}{X}'(A) &\equiv \forall c.(\Phi(c) = A) \Rightarrow \underset{\sim}{X}(c) \\
&\equiv \forall c.(\{ \theta(b) \mid b \in B(c) \} = A) \Rightarrow (\forall b \in B(c).\ X(b)) \\
&\Leftarrow \forall b \in \theta^{-1}(A).\ X(b) \\
&\equiv \forall d \in A.(\forall b \in \theta^{-1}(d).\ X(b)) \\
&\equiv X^+, \text{ say.}
\end{aligned}$$

X^+ is the predicate (of M) saying that all behaviours which might occur in a process mapping to A are correct, the phrase "might occur" being interpreted as element-wise possibility. (We are thus considering the behaviours which appear to be possibilities, even though it might be that we could eliminate them by consideration of the gross structure of A. For an example of the influence this has see the later examination of the traces model.)

Since $X^+ \Rightarrow \underset{\sim}{X}'$ we have $X^+(\Phi(c)) \Rightarrow \underset{\sim}{X}(c)$ for any $c \in C$.

We will also reserve the right to make θ a <u>relation</u>, in which case $\Phi(c) = \bigcup \{\theta(b) \mid b \in B(c)\}$. Behaviours then have any number of representatives in the image. To simplify matters, from here on we will always assume that θ is a function unless we clearly state the contrary. Except where

we indicate otherwise all results stated or proved for "Φ"s defined using functions are also true for those defined using relations. We will have to wait until the last two sections of this chapter for examples of the use of relations in defining maps to models.

The question of the expressiveness of "Φ"s defined using relations is less clear-cut than with functions, though it is still usually true that a "θ" which makes more distinctions will give rise to a more expressive model. It is now possible, however, for a behaviour which has a large image under "θ" to obscure much detail in the image (it is precisely for this reason that we will later use relations in some cases).

It is possible to define the "$^+$" operator on predicates of behaviour in much the same way as before. Suppose that "Φ" is a modelling function defined using a relation "θ". Then we have

$$\begin{aligned} \underset{\sim}{X}'(A) &\equiv \forall c.(\Phi(c) = A) \Rightarrow X(c) \\ &\equiv \forall c.(\bigcup\{\theta(b) \mid b \in B(c)\} = A) \Rightarrow (\forall b \in B(c).\ X(b)) \\ &\Leftarrow \forall b.\ \theta(b) \subseteq A \Rightarrow X(b) \\ &\equiv X^+, \text{ say.} \end{aligned}$$

Once again X^+ is the predicate of M saying that all behaviours which might occur in a process mapping to A are correct. This time we do not consider A element by element, since the result of doing this, $\forall d \in A.\forall b.((\theta(b) = \emptyset \vee d \in \theta(b)) \Rightarrow X(b))$, is in practice too strong to be of any use (or at least this so in the two examples of relations which we meet later). The penalty we pay for this is that the predicate X^+ defined relative to a relation may not have such a workable form. It is, for example, easy to see that any X^+ defined relative to a function must be strongly continuous when the map Φ is to either of our existing models, but that this need not always be so for relations. Also the results 8.2 and 8.3 proved below only hold in general when θ is a function. The most important single result about the predicates X^+, namely 8.4 does hold in general, however.

8.2 Theorem (holds only when θ is a function)

If X is any predicate of behaviour then there is another one ψ, such that $(X^+)^* = \underset{\sim}{\psi}$ and $X^+ = \psi^+$.

proof

By our earlier definitions

$$(\chi^+)^*(c) \Leftrightarrow \chi^+(\Phi(c))$$
$$\Leftrightarrow \forall d \in \Phi(c).(\forall b \in \theta^{-1}(d).\ \chi(b))$$
$$\Leftrightarrow \forall b \in B(c).(\forall b' \in \theta^{-1}(\theta(b)).\ \chi(b'))$$
$$\Leftrightarrow \underset{\sim}{\psi}(c), \text{ where } \psi(b) \Leftrightarrow \forall b' \in \theta^{-1}(\theta(b)).\ \chi(b')$$

For this same ψ we have

$$\psi^+(A) \Leftrightarrow \forall d \in A.(\forall b \in \theta^{-1}(d).\ \psi(b))$$
$$\Leftrightarrow \forall d \in A.(\forall b \in \theta^{-1}(d).(\forall b' \in \theta^{-1}(\theta(b)).\ \chi(b')))$$
$$\Leftrightarrow \forall d \in A.(\forall b \in \theta^{-1}(d).(\forall b' \in \theta^{-1}(d).\ \chi(b')))$$
$$\Leftrightarrow \forall d \in A.(\forall b' \in \theta^{-1}(d).\ \chi(b'))$$
$$\Leftrightarrow \chi^+(A)$$

If more restrictive conditions are placed upon the nature of the space of behaviours and upon M we can prove stronger results than the above.

8.3 Theorem

Suppose that for every behaviour b and $A \in M$ we have $\theta(b) \in A \Rightarrow \exists c \in C.b \in B(c)$ & $\Phi(c) \subseteq A$. Then for all predicates χ & ψ of behaviours we have $(\underset{\sim}{\chi} \equiv \underset{\sim}{\psi}) \Rightarrow (\chi^+ \equiv \psi^+)$. Furthermore if $\underset{\sim}{\chi} \equiv (\psi^+)^*$ then $\chi^+ \equiv \psi^+$.

proof

Suppose χ and ψ are such that $\underset{\sim}{\chi} \equiv \underset{\sim}{\psi}$. By symmetry, to prove $\chi^+ \equiv \psi^+$ it is enough to show $(\chi^+)(A) \Rightarrow (\psi^+)(A)$ for all $A \in M$. Suppose $(\chi^+)(A)$ holds and that $d \in A$ & $b \in \theta^{-1}(d)$.

$$(\chi^+)(A) \Rightarrow \forall d \in A.(\forall b \in \theta^{-1}(d).\ \chi(b)) \qquad (*)$$

By assumption there exists some $c \in C$ such that $b \in B(c)$ and $\Phi(c) \subseteq A$. If $b' \in B(c)$ then by construction $\theta(b') \in A$, so that $b' \in \theta^{-1}(d)$ for some $d \in A$. Thus $\chi(b')$ holds by (*). Hence $\forall b' \in B(c).\chi(b')$ $(\Leftrightarrow \underset{\sim}{\chi}(c))$ holds.

Since by assumption $\underset{\sim}{\chi} \equiv \underset{\sim}{\psi}$ we have $\underset{\sim}{\psi}(c)$.

Therefore $\psi(b)$ holds by definition of $\underset{\sim}{\psi}$ since $b \in B(c)$.

Thus on the assumptions $(\chi^+)(A)$, $d \in A$ and $b \in \theta^{-1}(d)$ we have proved $\psi(b)$. Hence

$$(\chi^+)(A) \Rightarrow \forall d \in A.(\forall b \in \theta^{-1}(d).\ \psi(b))\ (\Leftrightarrow (\psi^+)(A)),$$

which was what we wanted to prove.

This result shows that (under the stated conditions) the operator "$^+$" can be regarded as an operator on predicates of C (rather than on predicates of behaviour) without ambiguity (as long as the predicate has the form $\underset{\sim}{\chi}$ for some χ).

To prove the second half of 8.3 we need merely observe that if $\underset{\sim}{\chi} \equiv (\phi^+)^*$ then by 8.2 there is some τ such that $\underset{\sim}{\tau} \equiv (\phi^+)^*$ and $\tau^+ \equiv \phi^+$. But then $\underset{\sim}{\chi} \equiv \underset{\sim}{\tau}$, so $\chi^+ \equiv \tau^+$ by the first part of 8.3. Hence $\chi^+ \equiv \phi^+$ as desired.

This shows that when the conditions of 8.3 hold so that "$^+$" is a well-defined operator on predicates on C we have $((Y^+)^*)^+ \equiv Y^+$ for all predicates Y of the form $\underset{\sim}{\chi}$.

The condition stated in 8.3 is one which will tend to be met in all examples. It simply states that if a behaviour appears to be possible for processes mapping to $A \in M$ by elementwise consideration of A then there is some $c \in C$ which has this behaviour and whose behaviour does not exceed the bounds defined by A. See below for further clarification of the use of "subset" rather than "equality" in the condition.

Define a <u>monotonic</u> predicate on M to be one which satisfies the condition $A \subseteq B \;\&\; \Pi(B) \Rightarrow \Pi(A)$. The following result is obvious (from the definition of φ^+).

8.4 <u>Lemma</u>
For any predicate φ of behaviours the predicate φ^+ is monotonic, and furthermore if $N \in M$ and $\forall A \in N.\ \varphi^+(A)$ and $B \in M$ is such that $B \subseteq \bigcup N$ then also $\varphi^+(B)$.

Note that the simple structure of the predicate φ^+ ($\varphi^+(A) \equiv \forall d \in A.\ \xi(d)$) ensures that it will be pleasant in other ways. It will for example be strongly continuous in both our existing models (see 2.44, 5.10).

Suppose we were to restrict ourselves to the use of monotonic predicates in directly proving things about elements of C (from their values in M). This would have several advantages and several disadvantages. The advantages we will meet shortly. The chief disadvantage of this approach is that we lose a certain amount of expressive power by our abandonment of a large class of possible predicates to prove of $\Phi(c)$, in particular the predicates of the form $\underset{\sim}{\chi}'$ (not in general monotonic). The natural predicate to prove of $\Phi(c)$ in order to prove $\underset{\sim}{\chi}(c)$ is then χ^+, which may well not be true of $\Phi(c)$ when $\underset{\sim}{\chi}'$ is. In some cases χ^+ is very much stronger than $\underset{\sim}{\chi}'$, and can even be "false" when $\underset{\sim}{\chi}'$ is quite reasonable. The question of

whether a particular triple (C,M,Φ) is suitable for use with monotonic predicates will vary both with the structure of the triple and with the things we wish to prove. As we will see there are some cases where monotonic predicates are usually adequate and others where they are often not. We will also see that there are considerable advantages in the monotonic case in the modelling and implementation of operators, inasmuch as there is a large class of implementations which can be considered correct relative to the proving of monotonic predicates but not of general predicates. Say we are treating (M,Φ) as a class 1 model for C if we only seek to prove monotonic predicates and as a class 2 model if we seek to be able to prove general predicates.

Class 1 Models

We are likely to want to model in M the operators we wish to use on the space C. The purpose of this will be to prove things about the result of applying the operator in C from the value(s) in M of its operand(s). Modelling of operators is a two-way process: an operator in M can model in C or an operator in C can implement an operator in M. We must seek workable definitions of these terms. Define an operator "op" over M to be reasonable if it is both monotonic (in the sense $op(A,*) \subseteq op(B,*)$ if $A \subseteq B$) and there is a possible correct implementation of it over C (in the sense to follow). The reason that a "reasonable" operator should be implementable is obvious. The reason that it should be monotonic is that the more possible behaviours of its operands, the more behaviours might be expected possible of the result of applying it.

Say that $op^*:C^n \to C$ is an operator implementing $op:M^n \to M$ (or alternatively op models op^*) if for all $c_1,..,c_n \in C$

$$\Phi(op^*(c_1,...,c_n)) \subseteq op(\Phi(c_1),...,\Phi(c_n)) \ .$$

This means that the result of applying op^* has behaviours contained within the bounds predicted by applying op to the images in M of its operands.

This implies that if Π is any (monotonic) predicate and op correctly models op^* then

$$\Pi(op(\Phi(c_1),...,\Phi(c_n))) \Rightarrow \Pi^*(op^*(c_1,...,c_n))$$

so that for any predicate χ of behaviour

$$\chi^{+}(op(\Phi(c_1),..,\Phi(c_n))) \Rrightarrow (\chi^{+})^{*}(op^{*}(c_1,..,c_n)) \Rrightarrow \underset{\sim}{\chi}(op^{*}(c_1,..c_n))$$

8.5 Lemma

If each of a set of "basic" operators over M is correctly implemented then any composition of them is correctly implemented by the operator over C which results by composing the implementations of the basic operators in the natural way.

The proof of this result is an easy induction using the monotonicity of the basic operators over M.

For example, if f and g are two and one place operators on M respectively, and they are correctly implemented by f* and g* respectively, then the compound operator $h(A,B) = f(f(A,B),g(B))$ is correctly implemented by $h^{*}(c,d) = f^{*}(f^{*}(c,d),g^{*}(d))$.

As we have previously discovered, the simple notion of operators over M alone is unlikely to be sufficient to give a semantics to a language. We need to be able to cope with variables for such things as recursion and input to/assignment to variables. The obvious way to do this is to bring some kind of "state" into our calculations. Let us assume that there is some set Θ of formal parameters taking "process" values and a set Ξ of parameters taking values of other sorts. For the sake of simplicity we will assume in what follows that the sets of parameters and the values taken by elements of Ξ are the same in the two models. One could doubtless extend what follows to the case where there are "real" and "model" values taken by non process parameters and "real" parameters become locations. We will suppose that the constructs of our language are given values in the set M' of monotonic functions in $M^{\Theta}\times S \to M$, where S is the set containing all possible values of that component of the state which maps elements of Ξ to their possible values in T, a set of "tokens". Constructs in the space C will have the form $C^{\Theta}\times S \to C$ (= C'). We will shortly see examples of what these constructs might look like when we examine the semantics of our usual language over various models.

We will assume that the semantics of our language are

built up from a set B of basic monotonic operators on M'. Say that $A \in M''$, the set of expressible elements of M', if and only if A is the finite composition of a number of elements of B (there will always be several elements of B which are constants (no arguments), so this definition is not vacuous). Assume also that there is a set B* of basic operators over C' from which all constructible elements of C' are composed. We do not assume that there is any sense in which elements of B* attempt to implement elements of B.

If $e \in M'$ and $e^* \in C'$ say that e^* is a correct implementation of e if for all $\underline{c} \in C^{\Theta}$ and $s \in S$ we have $e(\Phi(\underline{c}),s) \supseteq \Phi(e^*(\underline{c},s))$, where Φ is extended in the natural (componentwise) way to C^{Θ}. This definition clearly has a similar effect to the earlier one (op* implementing op) but is different in that here the objects we are discussing are effectively at a lower level: we still need to study the implementations of operators over M'. Suppose that $f:M'^{k} \to M'$ is monotonic and $f^*:C'^{k} \to C'$. Say that $f^* \underline{imp}\ f$ if for all $e_1,..,e_k \in M'$ & $e_1^*,..,e_k^* \in C'$ such that e_i^* implements e_i correctly $f^*(e_1^*,..,e_k^*)$ is a correct implementation of $f(e_1,..,e_k)$. It is quite easy to check that this definition corresponds exactly to the old definition of correctness of operators in the degenerate cases covered by that definition.

Say that an element g of B is <u>reasonable</u> if there is some finite composition f_g^* of elements of B* which satisfies $f_g^* \underline{imp}\ g$. The justification for this definition is that g is only a reasonable operator if it is possible to implement it; if we construct elements of M' using non-reasonable operators we can have no guarantee that we will be able to construct real processes to implement them. The following result is obvious, but is nevertheless important.

8.6 <u>Theorem</u>

If for each $g \in B$ there corresponds an f_g^* with the above properties then every expressible element of M' is correctly implemented by a constructible element of C', the construct which results from composing the f_g^*s in the natural way modelling the composition of the expressible element.

For example, if f (0-place), g (1-place) and k (2-place) are all elements of B implemented by f*, g* and k* respectively then k*(g*(f*),f*) is a correct implementation of k(g(f),f).

From the above work we see that if Π is any monotonic predicate of M we get, for any $e^* \in C'$ correctly implementing $e \in M'$, that if $\underline{A} \in M^{\Theta}$, $\underline{c} \in C^{\Theta}$ and $s \in S$ are such that $\Phi(\underline{c}) \subseteq \underline{A}$ then $\Pi(e(\underline{A},s)) \Rightarrow \Pi^*(e^*(\underline{c},s))$. It is worth noting the common special case in which e is independent of $\underline{A}$, s or both, in that then $\Pi^*(e^*(\underline{c},s))$ is true independently of one or both components of the state. In the case of our usual language we would expect an expression with no free variables of any sort to give rise to an "e" independent of both $\underline{A}$ and s.

The technicalities of the above and lack of examples obscure the advantages gained from the assumption that our model is class 1. The fundamental advantage is the possibility of using "$\subseteq$" in our definitions of correct implementation, where otherwise we would have had to use "=". This informally means that an implementor merely has to restrict the behaviours of his resultant processes to be within the bounds specified by an operator over M, and need not make sure that in every case of applying his implementation of the operator there is a possible behaviour mapping under θ to every element of the predicted element of M. This corresponds to allowing the implementor to resolve non-determinism inherent in an operator. It also helps in that the map Φ may well identify large classes of structurally different elements of C. When defining an operator op it may be necessary to allow for this by including in op(A) ($A \in M$, for simplicity) values which arise from applying a natural implementation of op to some c mapping to A but not to others. We will see examples of both these points later.

The most important feature of the above work is that it provides an exact calculus by which we can prove results about the value in C of a process by consideration only of the language used to define the process.

As indicated above, in a class 2 model, where monotonic predicates are not considered sufficiently expressive, one

can derive a very similar calculus for proving predicates true of implemented processes. It is necessary to demand that all implementations are exact: $e^* \in C'$ will correctly implement $e \in M'$ only if for each $\underset{\sim}{c} \in C$ and $s \in S$ we have $\Phi(e^*(\underset{\sim}{c},s)) = e(\Phi(\underset{\sim}{c}),s)$.

Compositions of Mappings

Suppose that M is a class 1 model for a space C and that C is in turn a model for some space D. It is convenient to think of D as a space of processes and C as a space of "idealized" processes. We will suppose that we have the usual map $\Phi: C \to M$ and in addition a map $\Upsilon: D \to C$. Suppose that the set of behaviours of elements of C and D are denoted by B(c) and B'(d) respectively, which are subsets of U and V respectively (types of "behaviour"). It is, as we will later find, useful to know to what extent and under what conditions the composite map $T = \Phi \bullet \Upsilon$ is useful in modelling D by M. Let us assume that there is some translation function $\eta: V \to U$ (the identity if U = V) such that $B(\Upsilon(d)) \supseteq \{\eta(b') \mid b' \in B'(d)\}$ for each $d \in D$. This condition effectively says that the value $\Upsilon(d)$ assigned to each $d \in D$ (as its idealization) must not ignore possible behaviours of d.

It is clear from the above definitions that for all $d \in D$ we have $T(d) \supseteq \{\theta(\eta(b')) \mid b' \in B'(d)\}$. This allows us to place a bound on the behaviour of elements of D from a knowledge of T(d). In the case where U = V and η is the identity this bound is the same one as we had before. In this U = V case the predicates of C which have the form $\underset{\sim}{\chi}$ (for a predicate χ of U) clearly extend in a natural way to the space D, with the result that $\chi^+(T(d)) \Rightarrow \underset{\sim}{\chi}(d)$. In cases where the function η is many-one there is the possibility that a predicate of V might not translate reasonably to a predicate of U, but we always have that for any predicate χ of V $(\chi'')^+(T(d)) \Rightarrow \underset{\sim}{\chi}(d)$, where $\chi''(b) \Leftrightarrow \forall b'.(\eta(b')=b) \Rightarrow \chi(b')$. Note that the predicate χ'' of U is to χ just what ψ' is to ψ for a predicate ψ of C, with the same result that the more discriminating the function η the more likely χ'' is to be reasonable.

Note that because of the decisions made above the space M is not a model for D in the sense introduced earlier

since we do not in general have that $T(d) = \{\theta^*(b') \mid b' \in B'(d)\}$ for any θ^*. It is clear that as long as we restrict ourselves to monotonic predicates this is of little consequence, if we are reasonably careful.

It is desirable to find a criterion by which to judge the implementations over D of operators over C which is "transitive" in the sense that a correct implementation op** (over D) of op* (over C) which in turn is a correct implementation of op (over M) is a correct implementation of op.

Define $D' = D^{\Theta} \times S$ to be the space of constructs over D (analogous to M' and C'). Say that $e^{**} \in D'$ implements $e^* \in C'$ correctly if for all $(\underset{\sim}{d},s) \in D^{\Theta} \times S$ we have $B(Y(e^{**}(\underset{\sim}{d},s))) \subseteq B(e^*(Y(\underset{\sim}{d}),s))$ (Y extended in the natural way to the product space). This definition (similar in form to our earlier ones) is justified by the following result.

8.7 Lemma

If $e^{**} \in D'$, $e^* \in C'$ and $e \in M'$ are such that e^{**} correctly implements e^* and e^* correctly implements e then for all $(\underset{\sim}{d},s) \in D^{\Theta} \times S$ we have $T(e^{**}(\underset{\sim}{d},s)) \subseteq e(T(\underset{\sim}{d}),s)$, (where T is extended in the natural way to D).

proof

$$
\begin{aligned}
T(e^{**}(\underset{\sim}{d},s)) &= \Phi(Y(e^{**}(\underset{\sim}{d},s))) \\
&= \{\theta(b) \mid b \in B(Y(e^{**}(\underset{\sim}{d},s)))\} \\
&\subseteq \{\theta(b) \mid b \in B(e^*(Y(\underset{\sim}{d}),s))\} \qquad (+) \\
&\subseteq \Phi(e^*(Y(\underset{\sim}{d}),s)) \\
&\subseteq e(\Phi(Y(\underset{\sim}{d})),s) = e(T(\underset{\sim}{d}),s) \qquad (++)
\end{aligned}
$$

(+) follows by definition of e** correctly implementing e*, and (++) follows by definition of e* correctly implementing e.

Suppose op** is a k-place operator over D' and op* is a k-place operator over C'. Say that op** imp' op* if whenever $e_1^{**},..,e_k^{**} \in D'$ and $e_1^*,..,e_k^*$ are such that each e_i^{**} correctly implements e_i^* then $op^{**}(e_1^{**},..,e_k^{**})$ correctly implements $op^*(e_1^*,..,e_k^*)$.

The corresponding result to 8.6 clearly holds for a set of basic operators B** over D' which can be combined to implement each basic operator (in B*) over C'.

This result, together with 8.6, 8.7 and the fact that for any $d \in D$ we have $T(d) \subseteq \{\theta(\eta(b')) \mid b' \in B'(d)\}$ allows us to form a linked system of implementations with respect to which it is possible to prove predicates of the form $\underset{\sim}{\chi}$ (χ a predicate of V) by consideration of the value taken by a program in M.

There is of course no reason why we must restrict ourselves to three levels of abstraction (D,C,M). We can introduce as many extra levels between D and C as we like, so long as the relationship between consecutive spaces has the same form as that between the old D and C. We would then have the following:

$$\begin{array}{lll} D=C_n \xrightarrow{\gamma_n} C_{n-1} \xrightarrow{\gamma_{n-1}} \dots \xrightarrow{\gamma_1} C_0 \xrightarrow{\Phi} M & & (C_0 = C) \\ U_n \xrightarrow{\eta_n} U_{n-1} \xrightarrow{\eta_{n-1}} \dots \xrightarrow{\eta_1} U_0 & & (U_n = V,\ U_0 = U) \end{array}$$

for spaces of processes C_i with associated spaces of behaviours U_i. We would demand that they satisfy (for all $1 \leq i \leq n$) $B_{i-1}(\gamma_i(d_i)) \supseteq \{\eta_i(b) \mid b \in B_i(d_i)\}$ for each $d_i \in C_i$ (where $B_i(d_i)$ represents the set of behaviours associated with d_i). Having done this we could prove very much the same sort of result about this system and its implementations as before.

The work of this section allows us to break down our analysis of the relationships between systems into easier steps. We can construct one or more idealized versions of a system of "real" processes for use as stepping-stone(s) between our model M and the real world. We are allowed to break up the problems of proving correctness of implementations into two or more distinct parts. This work also has the advantage that once we have established the relationship between M and C we can use our knowledge of this for many different "D"s (and vice-versa).

The application of our theory

The rest of this chapter is devoted to analyzing our existing models and seeing how they might be improved. The first thing we must do is to establish the space C of "real" processes by which we are to judge them. We will first form an informal picture of what we expect a process to look like. We will then formalize these ideas into postulates on the behaviour of processes, and as a result produce a fairly

abstract system C which is in a sense a unique model for them.

It should of course be noted that the following is only one possible system by which to judge our models. It does however have the advantages of its very general nature and the fact that it corresponds closely to our intuitive idea of what a process should look like.

Let us therefore suppose the following of the processes which we seek to model:
(i) That at all times a process has an internal state, and that this state and the process' environment are the only factors influencing its behaviour. The state can change only through discrete actions (or transitions). An action is instantaneous, and can either be internal or external. Internal actions are uncontrollable by the environment and indistinguishable to it (if it can observe them at all). External actions, or communications, can take place only with the co-operation of the environment. Only finitely many actions can occur in any given finite interval.

(ii) External actions are named by elements of some alphabet Σ; this name is the only feature of an external action visible to the environment. The only device by which the environment can influence the behaviour of a process is the subset of Σ in which it is willing to co-operate. This influence is restricted to what communications actually take place, not any other feature of the set.

(iii) There is some finite uniform time for which it is not possible for an action to be possible for a process without some (possibly different) action occurring. Apart from this the behaviour of a process is independent of the length of time it has been in a state, subject to the condition that only finitely many actions can occur in a finite time.

Our next step will be to reinforce the above postulates with some more formal ones. To do this we will need a notation for describing behaviour. Denote by $(\rho, X) \stackrel{\cdot}{\rightarrow} \tau$ the fact that a process in state ρ, while the environment offers set X, might perform some internal action which results in it coming

into state τ. Similarly denote by $(\rho,X) \xrightarrow{a} \tau$ the fact that under the same conditions a process in state ρ might perform some external action named "a" and come into state τ. Postulates (i), (ii) and (iii) essentially imply that once we know both the initial state of a process and the possible transitions in the above form of all states, we know exactly which sequences of actions are possible for the process (relative to its environment). This will be made more precise below. The following three postulates formalize and extend another set of ideas introduced in (i), (ii) and (iii).

(P1) $(\sigma,X) \xrightarrow{a} \rho \Rightarrow a \in X$

(P2) $(\sigma,X) \dashrightarrow \rho \Leftrightarrow (\sigma,\emptyset) \dashrightarrow \rho$

(P3) $(\sigma,X \cup \{a\}) \xrightarrow{a} \rho \Leftrightarrow (\sigma,\{a\}) \xrightarrow{a} \rho$

The first of these simply says that a process can only carry out an external action with the co-operation of the environment. The second formalizes the notion that the environment cannot influence the possibility of an internal action occurring. The third formalizes the notion that the only influence which the environment has is through what actually occurs, not what might have occurred.

Our next step will be to bring in a set of postulates which specify how a process will behave in any environment, this behaviour being a function of the possible state transitions. To this end we will introduce the "behaviours" which we will use to describe individual execution sequences of a process. We must be careful in our choice of which type of "behaviour" to use since the sets of behaviours of processes play a central role in the first sections of this chapter. One type of "behaviour" which one might suggest is the observed response of the process to some type of experiment carried out by the environment: this would correspond well to our intuition about the meanings of our existing models, and such behaviours would lend themselves easily to the construction of correctness predicates. It is however rather early to have to decide either the nature of the experiments to be carried out by the environment or exactly what is observable by the environment (e.g. whether or not internal actions are discernable in any way by the environment).

To avoid having to make these decisions, and so as to make our space better able to model less abstract ones, we will choose a type of behaviour which attempts to record as much relevant detail as possible about an execution sequence without attempting to extract the "observable" aspects. The extraction of "observable" aspects of a behaviour is a task better left to the function "θ" which maps behaviours to their representatives in the image in the model of a process. It is necessary however to decide a few general points about our environments. Let us therefore decide that an environment has the ability to apply only a finite number of sets to a process in a finite time, but that there is no necessity for these sets to be themselves finite. Both of these decisions are consistent with the thought that the environment might itself be a process; this idea would not be so easy to entertain if the sets were always finite.

The obvious choice of what a behaviour is (bearing the above in mind) is a finite or infinite sequence of pairs (state and environment) linked by any state transitions which occur between them. In a more general space of processes it might be desirable to include a third component representing the time for which the state/environment pair persists, but because of the idealized nature of our space and the independence of our models from time there seems little need to include this extra component here. It is important however to be able to distinguish finite from infinite behaviours: there is no problem in the case of infinite sequences, since by our assumptions about the nature of processes and environments these can only represent infinite behaviours (i.e. behaviours which occupy an infinite amount of time). There is however the possibility that a finite sequence might represent an infinite experiment, for a stubborn environment might encounter a state which was unwilling either to accept any of the environment's symbols or to perform any internal action. For obvious reasons this can only occur at the end of a behaviour.

The most convenient form of the behaviours described above is finite or infinite sequences of triples of the form (σ,X,δ), for σ a state, X a set offered by the environment and δ either an element of Σ (representing a communication),

"·" (representing an internal action), "*" (representing a finite or infinite fruitless wait which is longer than the bound on the inactivity of a state which can do something), or "-" (representing a finite fruitless wait which is shorter than this bound). The only possible "δ"s for the final triple of a finite sequence are "*" and "-".

In the following we will typically use $\underline{a}$, $\underline{b}$,... to denote sequences of triples. If c is a process we assume that it is endowed with a set $B(c)$ of behaviours, the set of behaviours which can actually occur for c. Thus when we define a system of processes it is necessary to define the possible transitions of the processes' states and the sets of behaviours of the processes. It is natural to expect transitions and behaviours to be closely related; the following postulates describe this relation, and also complete the formalization of the "informal" postulates (i), (ii) and (iii).

(Q1) $\langle\rangle \in B(c)$

(Q2) $\langle\!\langle \sigma, X, - \rangle\!\rangle \in B(c)$ iff σ is the initial state of c.

(Q3) $\underline{a}\,\langle(\sigma,X,-)\rangle \in B(c)$ & $(\sigma,X) \xrightarrow{\delta} \rho \Leftrightarrow \underline{a}\,\langle\!\langle(\sigma,X,\delta)(\rho,Y,-)\rangle\!\rangle \in B(c)$

(Q4) $\underline{a}\,\langle(\sigma,X,-)(\rho,Y,\delta)\rangle\,\underline{b} \in B(c) \Leftrightarrow \sigma=\rho$ & $\underline{a}\,\langle(\sigma,Y,\delta)\rangle\,\underline{b} \in B(c)$

(Q5) $\underline{a}\,\langle(\sigma,X,-)\rangle\,\underline{b} \in B(c)$ & $(\neg\exists \rho,\delta.\delta \in X \cup \{\cdot\}$ & $(\sigma,X) \xrightarrow{\delta} \rho)$
$\Leftrightarrow \underline{a}\,\langle(\sigma,X,*)\rangle\,\underline{b} \in B(c)$

(Q6) $\underline{a}\,\langle(\sigma,X,-)\rangle\,\underline{b} \in B(c) \Rightarrow \underline{a}\,\langle(\sigma,X,-)\rangle \in B(c)$

(Q7)* If $\underline{a} = \langle(\sigma_o,X_o,\delta_o)\ldots(\sigma_i,X_i,\delta_i)\ldots\rangle$ is an infinite behaviour sequence such that infinitely many of the δ_i are elements of $\Sigma \cup \{\cdot\}$ and such that $\langle(\sigma_o,X_o,\delta_o)\ldots(\sigma_i,X_i,-)\rangle$ is an element of $B(c)$ for all i, then $\underline{a} \in B(c)$.

(Q7) If $\underline{a} = \langle\!\langle(\sigma_o,X_o,\delta_o)\ldots(\sigma_i,X_i,\delta_i)\ldots\rangle\!\rangle$ is an infinite behaviour sequence such that δ_i = "-" for all $i \geqslant k$ and such that $\langle(\sigma_o,X_o,\delta_o)\ldots(\sigma_i,X_i,-)\rangle \in B(c)$ for all i, then $\underline{a} \in B(A)$ iff $\langle\!\langle(\sigma_o,X_o,\delta_o)\ldots(\sigma_k,Z,*)\rangle\!\rangle \in B(c)$, where $Z = \bigcup_{j=k}^{\infty}(\bigcap_{i=j}^{\infty} X_i)$.

(*) On certain occasions it will be necessary to replace Q7 with a less severe postulate.

The first two of these are easy to interpret. Q3 says that a transition can occur in behaviours exactly when it is

implied possible by the transition relations. Q4 says both that a finite application of a set without response by the environment cannot influence the behaviour of a process and (together with Q2 and Q3) that it is always possible that the state may fail to respond to any set only applied for a short time. (This can be regarded as a convenient fiction which makes the structure of behaviours more tractable, and therefore makes these postulates simpler. For more about this assumption see later.) Q5 says that a fruitless wait which is longer than our bound on the bound on inactivity is possible when, and only when, there is no internal or external transition possible for the state in the environment which is offered. Q6 says that any initial part of a possible behaviour is possible. Q7 essentially postulates the absence of fairness, saying that an infinite sequence of transitions is possible if each of its finite approximations is possible. If we are able to assume this it removes much complexity from our arguments; we will find however that it is sometimes impossible to define reasonable operators without an element of fairness. Q8 reflects the fact that the environment can only apply a finite number of sets in a finite time: thus no transition can be possible during the whole of a final portion of an infinite sequence of fruitless waits, as this would contradict our assumption that no transition can be possible infinitely without something occurring.

It is possible to simplify the form of some of the Q-postulates if we assume the P-postulates. It is however desirable to keep them separate so that we can change the form of the P-postulates without having to alter all the Q-postulates as well.

One useful feature of the system of postulates set out above (Q1 - Q8) is that whether or not the system satisfies P1 - P3 the behaviour set $B(c)$ of a process is always exactly determined by its initial state and the transitions $(\sigma,X) \dashrightarrow \rho$ and $(\sigma,X) \xrightarrow{a} \rho$ which are possible for its states. Thus we can exactly specify the nature of a system of processes by defining which transitions are possible for its states and saying that it satisfies Q1 - Q8. This is a freedom which we will often make use of later in our abstract discussion of spaces of processes. If one were to

decide to alter one of Q1 - Q8 (for example Q7) it would be useful if this "exact determination of behaviour" could in some sense be preserved.

An immediate consequence of the postulates P1 - P3 is that we can simplify the structure of our transition relations. If we write $\sigma \stackrel{\cdot}{\rightarrow} \rho$ for $(\sigma,\emptyset) \stackrel{\cdot}{\rightarrow} \rho$ and $\sigma \stackrel{a}{\rightarrow} \rho$ for $(\sigma,\Sigma) \stackrel{a}{\rightarrow} \rho$ then it is easy to see that the possible old-style transitions are exactly determined by a knowledge of which new-style ones are possible. Thus for all states σ & ρ, environments X and $a \in \Sigma$ we have:

$$(\sigma,X) \stackrel{\cdot}{\rightarrow} \rho \Leftrightarrow \sigma \stackrel{\cdot}{\rightarrow} \rho$$
$$(\sigma,X) \stackrel{a}{\rightarrow} \rho \Leftrightarrow \sigma \stackrel{a}{\rightarrow} \rho \ \& \ a \in X.$$

Before we continue with our study of the systems which satisfy these postulates it is perhaps desirable to consider briefly just how valid and useful the postulates are. The first thing which we should remark on is the fact that our postulates do indeed seem to paint a very idealized picture of what a process is likely to be. It is certainly tempting, and also justifiable, to relax some of them somewhat. It is important to note however that exception of our assumption of the existence of a uniform bound on the "idle time" of a state (a postulate which is by no means critical), each of the postulates which might be regarded as controversial has the effect of assuming maximal unpredictability on the part of our processes. This is because of our emphasis on possible, rather than certain, behaviours, which means that the absence of a behaviour is never checkable by any experimenter. The following paragraphs are brief analyses of some of the points at which this is true.

a) One might regard our postulate that actions occur instantaneously as suspect, and also perhaps our assumption that the behaviour of a state is independent of time. One might prefer to regard actions as events which take a non-zero time to complete, and to believe that as a state develops it can acquire more possible actions (reductions can be modelled by internal actions). In many ways the best way of dealing with the first of these points is to identify the start of each action with the action itself, so that "actions" are again thought of as instantaneous. It is interesting that this identification is also inconsistent with

the postulate that a state's possible transitions remain constant throughout its life. We are forced to the conclusion that the possible transitions of a state may increase gradually during the first part of its life (on the completion of the various actions which may be in progress by the possibly distributed state).

To take account of this type of time dependance we would have to modify our transition relations to include a time component, so that for example $(\sigma,X,t) \dot{\rightarrow} \rho$ would mean that after spending time t in state σ a process might, in environment X, perform some internal action which transforms it into state ρ. We would have to modify some of our postulates. Firstly we would assume that the bound on "idle time" was increased to take account of the possible "warming up" time of states. Most of the necessary modifications to the P and Q postulates are fairly obvious. The only major changes would be in the form of Q5 and the addition of a P4.

(P4) $\quad (\sigma,X,t) \xrightarrow{\delta} \rho \;\&\; t < t' \;\Rightarrow\; (\sigma,X,t') \xrightarrow{\delta} \rho$

(Q5') $\quad \underline{a}\langle(\sigma,X,-)\rangle \underline{b} \in B(c) \;\&\; \neg(\exists t,\delta,\rho.\, \delta \in X \cup \{\cdot\} \;\&\; (\sigma,X,t) \xrightarrow{\delta} \rho)$
$\qquad \Leftrightarrow \; \underline{a}\langle(\sigma,X,*)\rangle \underline{b} \in B(c)$

It should not be too hard to prove that if we took a process whose states satisfied the modified postulates, and mapped it to another with the same transitions but independent of time, then the two would have identical sets of behaviours. While P4 would probably not be necessary for such a proof, its assumption is crucial in making Q5' reasonable.

b) The "$\Leftarrow$" halves of P2 and P3 might be regarded as being slightly suspect. This is because we might like to believe that a process might prefer one (external) action to another (internal or external). If this were possible then it might occur that some transition $(\rho,X) \dot{\rightarrow} \tau$ or $(\rho,X) \xrightarrow{a} \tau$ be possible only when some symbol b is not an element of the set X. It would be reasonable to expect that in this case there is some state v such that $(\rho,X\cup\{b\}) \xrightarrow{b} v$ for all X such that the original transition $(\rho,X) \xrightarrow{a} \tau$ or $(\rho,X) \dot{\rightarrow} \tau$ is possible. We would need a single extra postulate to replace the "$\Leftarrow$" halves of P2 and P3, for example P4' below.

(P4') There is some partial order ">" on the set $\{(a,\tau) \mid (\rho,\{a\}) \xrightarrow{a} \tau\} \cup \{(\cdot,\tau) \mid (\rho,\emptyset) \not\rightarrow \tau\}$ with the following properties:

a) The partial order has no infinite ascending chains (so that all its non-empty subsets have maximal elements).

b) Every pair of the form $(\cdot,\tau)$ is minimal; there can be no $a,\tau\&\upsilon$ such that $(a,\tau) > (a,\upsilon)$.

c) For each set X we have $(\rho,X) \xrightarrow{\delta} \tau$ if and only if (δ,τ) is maximal in $\{(a,\upsilon) \mid a \in X \ \& \ (\rho,\{a\}) \xrightarrow{a} \upsilon\} \cup \{(\cdot,\upsilon) \mid (\rho,\emptyset) \not\rightarrow \upsilon\}$ (with respect to the partial order).

Note that (c) above actually implies (b), and also the "$\Rightarrow$" clauses of P2 and P3. The P4' which results from the vacuous partial order is equivalent to P2 and P3. The necessity for condition (a) above arises from the ridiculous situations which would arise if it did not hold: the state having a sequence of possible actions each of which is made impossible by the presence of another, so that in fact no action actually takes place.

One should be able to prove that if we took a system whose states satisfied the above postulate and P1 and Q1-Q8 and mapped it to another which satisfied our original postulates, each state being mapped to one with exactly the same transitions of the form $(\rho,\emptyset) \not\rightarrow \tau$ and $(\rho,\{a\}) \xrightarrow{a} \tau$, then the set of behaviours of each process is contained in the set of behaviours of its image. The critical feature of a proof of this will be (for Q5) the fact that for each state ρ (with image ρ') and set X we have $\neg\exists\tau,\delta.(\rho,X) \xrightarrow{\delta} \tau \ \Rightarrow \neg\exists\tau,\delta.(\rho',X) \xrightarrow{\delta} \tau$.

c) When it is assumed the absence of fairness (Q7) might be regarded a being a little restrictive. It is easy to see however that the set of behaviours of a process in a space not satisfying Q7 is contained in that of the process' image under the obvious map to the corresponding space which does satisfy it. If we were to drop the "convenient fiction" of the "$\Leftarrow$" half of Q4 and bring in weaker postulates the same would be true.

Each of the sections (a), (b) & (c) demonstrates that we

should be able to produce maps from "weaker" spaces to our idealized spaces which increase the sets of behaviours. There is a strong sense in which a process with more behaviours than another can be regarded as being more non-deterministic than it, so that by idealizing a process we are in fact assuming it to be worse than it really is. It is important to note that this work shows that spaces which satisfy our postulates are good candidates for being the space "C" in the section on "compositions of mappings", since the correctness condition for the function $Y:D \to C$ was that it increased the sets of behaviours of processes.

There is clearly much scope for further work on the above topic. Firstly there is a need for formal proofs of the various results which were derived informally above. Secondly it would be interesting to know how many of the postulates we could weaken simultaneously (e.g. can we weaken P2 and P3 as well as removing independence of states from time?). There should be no great difficulty in solving these problems. The formal analysis of behaviour sets which is necessary will be made easier by the following result.

8.8 Theorem

In a system satisfying Q1 - Q8 (whether or not it satisfies P1, P2 and P3) the behaviour set $B(c)$ of a process is uniquely determined by a knowledge of its initial state and a knowledge of which transitions $(\rho,X) \stackrel{a}{\to} \tau$ and $(\rho,X) \stackrel{\cdot}{\to} \tau$ are possible for all its states.

The proof of this is an easy induction on the length of finite elements of $B(c)$. The infinite elements fall into place using Q6, Q7 and Q8.

In systems which satisfy P1, P2 and P3 in addition to Q1-Q8 we can deduce that the behaviour sets of processes are uniquely determined by a knowledge of the possible transitions $\rho \stackrel{\cdot}{\to} \tau$ and $\rho \stackrel{a}{\to} \tau$. We can use this fact not only to help to justify the use of such spaces to model weaker ones, but also to prove the existence of canonical spaces satisfying our postulates which can be held to model **many others.** It is convenient at this stage to identify processes with their initial states, thereby embedding spaces of processes into

their underlying spaces of states. Without any significant loss of generality we can clearly identify the spaces of processes and states.

Suppose that C and C' are two spaces of processes (states) satisfying our postulates, and that $F:C \to C'$ is a function from one to the other. Define F to be a morphism if it satisfies the conditions:

(i) $\forall\sigma.\forall\rho.\forall\delta\in\Sigma\cup\{\cdot\}.\ \sigma \stackrel{\delta}{\to} \rho \Rightarrow F(\sigma) \stackrel{\delta}{\to} F(\rho)$

(ii) $\forall\sigma.\forall\rho.\forall\delta\in\Sigma\cup\{\cdot\}.\ F(\sigma) \stackrel{\delta}{\to} \rho \Rightarrow \exists\tau\in F^{-1}(\rho).\ \sigma \stackrel{\delta}{\to} \tau$.

Condition (i) essentially says that all transitions possible for an element of C must also be possible for its image, and that this is also true of the process after all transitions. Condition (ii) says that all transitions possible for the image of a process are also possible for the process itself.

8.9 Lemma

If C, C' and C" are three spaces of processes with morphisms $F:C \to C'$ and $G:C' \to C''$, then $G\circ F$ is a morphism from C to C".

proof

(i) If σ,ρ C' and $\delta\in\Sigma\cup\{\cdot\}$ are such that $\sigma \stackrel{\delta}{\to} \rho$ then
$F(\sigma) \stackrel{\delta}{\to} F(\rho)$ (as F is a morphism)
$\Rightarrow G(F(\sigma)) \stackrel{\delta}{\to} G(F(\rho))$ (as G is a morphism).

(ii) If $\sigma\in C$, $\rho\in C''$ and $\delta\in\Sigma\cup\{\cdot\}$ are such that $G(F(\sigma)) \stackrel{\delta}{\to} \rho$ then there is some $\tau\in C'$ s.t. $F(\sigma) \stackrel{\delta}{\to} \tau$ and $G(\tau) = \rho$. This in turn implies that there is some $\upsilon\in C$ s.t. $\sigma \stackrel{\delta}{\to} \upsilon$ and $F(\upsilon) = \tau$. We thus have $\sigma \stackrel{\delta}{\to} \upsilon$ and $\upsilon\in(G\circ F)^{-1}(\rho)$.

From here on let us use the term P,Q-space for a space of processes (states) which satisfies P1-3 and Q1-8. If C and C' are two P,Q-spaces and $F:C \to C'$ is any function we can extend F to the spaces of triples used in forming behaviours (and hence to the spaces of behaviours themselves) by $F(\sigma,X,\delta) = (F(\sigma),X,\delta)$. The following is a result which can be proved by induction which shows that when F is a morphism the spaces of behaviours of a process and its image under F correspond in a useful way.

8.10 Lemma

If C and C' are two P,Q-spaces and $F:C \to C'$ is a morphism, then for all $\rho\in C$ we have $B(F(\rho)) = F(B(\rho))$.

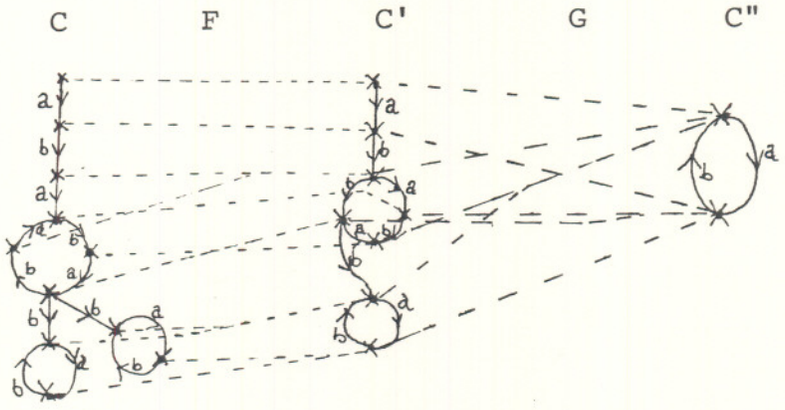
The proof of this result is omitted, being a fairly lengthy

argument by cases. The author's proof of the case of infinite sequences uses the axiom of choice.

The chief significance of this result is that it tells us that so far as the environment is concerned (assuming that it cannot "see" the actual internal state of a process) there is nothing to distinguish a process from its image under a morphism.

8.11 Example

The following diagram illustrates the concept of morphisms between spaces of processes. C, C' and C" are three P,Q-spaces, their possible actions indicated by unbroken lines. The two maps F and G (indicated by broken lines) are morphisms.



The following results will help us in our search for spaces which are canonical in the sense of having unique morphisms to them from large classes of P,Q-spaces.

8.12 Lemma

Suppose that C, C' and C" are three P,Q-spaces with morphisms $F:C \to C'$ and $G:C \to C"$. We can define an equivalence relation on C by $\sigma \sim \rho \Leftrightarrow \exists k, \tau_1, v_1, \ldots, v_k . \sigma = \tau_1 \;\&\; \rho = v_k \;\&\; \forall i . F(\tau_i) = F(v_i) \;\&\; \forall i . G(v_i) = G(\tau_{i+1})$.

The quotient space $C/\sim$ can be made into a P,Q-space by defining the transitions by $\overline{\sigma} \xrightarrow{\delta} \overline{\rho} \Leftrightarrow \exists \tau \in \overline{\sigma}, v \in \overline{\rho} . \tau \xrightarrow{\delta} v$ and the behaviours by Q1-8. This is a well-defined P,Q-space, and the map $H(\rho) = \overline{\rho}$ is a morphism.

The proof of this result is technical and is omitted.

The chief purpose of 8.12 is to show that if F and G are two morphisms of a P,Q-space C then there is a third which identifies all pairs of processes identified by either F or G. An immediate corollary to 8.12 is thus the fact that the relation on C defined by "$\sigma \backsim' \rho \Leftrightarrow \exists$some morphism F which identifies σ & ρ" is an equivalence relation. This equivalence relation is used to prove 8.14 below.

If C is any P,Q-space let us define a non-empty subset D of C to be a subspace of C if it is closed under "$\rightarrow$" in the sense $\exists \sigma \in D.\ \exists \delta\ .\ \sigma \xrightarrow{\delta} \rho \Rightarrow \rho \in D$. (An element of D has the same transitions and behaviours which it has as an element of C.) It is clear that with the spaces of transitions and behaviours described D is a P,Q-space.

8.13 Lemma

a) If C" is a subspace of C' and C' is a subspace of C then C" is a subspace of C.

b) If $F:C \rightarrow C'$ is a morphism and D is a subspace of C then F(D) is a subspace of C'. (Corollary: Im(F) is a subspace of C'.)

c) If $F:C \rightarrow C'$ is a morphism and D is a subspace of C' then $F^{-1}(D)$ is a subspace of C.

d) If D is a subspace of C and $F:D \rightarrow D'$ is a morphism such that $F(\sigma) = F(\rho)$ $(\rho,\sigma \in D)$, then there is a space C' and a morphism $F':C \rightarrow C'$ such that $F'(\sigma) = F'(\rho)$.

proof

(a) is obvious; (b) follows from clause (ii) of our definition of morphisms; (c) follows from clause (i) of our definition of morphisms; (d) follows by consideration of the equivalence relation $\backsim$" defined $\upsilon \backsim'' \tau$ if and only if either $\upsilon = \tau$ and $\upsilon \notin D$ or $\tau, \upsilon \in D$ and $F(\upsilon) = F(\tau)$. We can make $C/\backsim$" into a P,Q-space by endowing it with the transitions $\overline{\upsilon} \xrightarrow{\delta} \overline{\tau}$ s.t. $\exists \upsilon' \in \overline{\upsilon},\ \tau' \in \overline{\tau}.\ \upsilon' \xrightarrow{\delta} \tau'$. It is not hard to show that the map $F'(\upsilon) = \overline{\upsilon}$ is the morphism we require.

Having established the technical results 8.11 and 8.12 we are in a position to prove the following important result.

8.14 Theorem

If C is any P,Q-space and $\backsim'$ is the equivalence relation defined on it as above then the quotient space $C/\backsim'$, when

made into a P,Q-space with transitions $\overline{\sigma} \overset{\delta}{\to} \overline{\rho} \Leftrightarrow \exists \tau \in \overline{\sigma},\ \upsilon \in \overline{\rho}.\ \tau \overset{\delta}{\to} \upsilon$, has the property that the map $F^*:C \to C/\sim'$ defined by $F^*(\sigma) = \overline{\sigma}$ is a morphism. Call $C/\sim'$ by the name C^*. F^* is the only morphism from C to C^*, and if G is any morphism from C to a P,Q-space D then there is a unique morphism from Im(G) to C^*, and this morphism G^* satisfies $G^* \circ G = F^*$.

proof

We will first show that F^* is a morphism. Trivially $\sigma \overset{\delta}{\to} \rho \Rightarrow F^*(\sigma) \overset{\delta}{\to} F^*(\rho)$ for all $\sigma, \rho \in C$ (since $\sigma \in \overline{\sigma}$ and $\rho \in \overline{\rho}$). We must therefore show that $\overline{\sigma} \overset{\delta}{\to} \overline{\rho} \Rightarrow \exists \tau \in \overline{\rho}.\ \sigma \overset{\delta}{\to} \tau$. Suppose then that $\overline{\sigma} \overset{\delta}{\to} \overline{\rho}$; this means that there exist some $\sigma' \in \overline{\sigma}$ and $\rho' \in \overline{\rho}$ s.t. $\sigma' \overset{\delta}{\to} \rho'$. By definition of our equivalence relation there exist morphisms F and G of C such that $F(\sigma) = F(\sigma')$ and $G(\rho) = G(\rho')$. By 8.12 this implies the existence of some morphism $H:C \to C'$ (for some C') such that $H(\rho) = H(\rho')$ and $H(\sigma) = H(\sigma')$. Since H is a morphism we must have that $H(\sigma) \overset{\delta}{\to} H(\rho)$, which in turn implies that there exists some ρ'' s.t. $H(\rho) = H(\rho'')$ and $\sigma \overset{\delta}{\to} \rho''$. We have thus shown the existence of some H and ρ'' s.t. $H(\rho) = H(\rho'')$ (so that $\overline{\rho} = \overline{\rho}''$) and $\sigma \overset{\delta}{\to} \rho''$. This completes the proof that F^* is a morphism.

If C' is any other P,Q-space with a morphism $G:C \to C'$ then Im(G) is a subspace of C' by 8.13. We can define a map $G^*:Im(G) \to C^*$ by $G^*(G(\rho)) = F^*(\rho)$. This map is well-defined since whenever $G(\upsilon) = G(\tau)$ we have $\tau \sim' \upsilon$, which implies that $F^*(\tau) = F^*(\upsilon)$. G^* is a morphism since $G(\tau) \overset{\delta}{\to} G(\upsilon) \Rightarrow \exists \upsilon' \in G^{-1}(\upsilon).\ \tau \overset{\delta}{\to} \upsilon'$; we then have $F^*(\tau) \overset{\delta}{\to} F^*(\upsilon')$, which implies $G^*(G(\tau)) \overset{\delta}{\to} G^*(G(\upsilon))$ as required (as $G(\upsilon') = G(\upsilon)$). Secondly if $G^*(G(\rho)) \overset{\delta}{\to} \tau$ then $\tau' \in F^{*-1}(\tau).\ \rho \overset{\delta}{\to} \tau'$ (as F^* is a morphism); since G is a morphism we then get $G(\rho) \overset{\delta}{\to} G(\tau')$, which is what we require since $G(\tau') \in G^{*-1}(\tau)$ by construction. Thus G^* is indeed a morphism as claimed, and it is obvious from its definition that $G^* \circ G = F^*$.

It remains to examine the question of uniqueness. If F' were a second morphism $F':C \to C^*$ then by the above there exists a morphism $F'':Im(F') \to C^*$ such that $F^* = F'' \circ F'$. We must show that F'' is the identity morphism. Claim first that F'' is one-one. If it were not then we could (by 8.13(d)) find a space C'' and a morphism $F^+:C^* \to C''$ which was not one-one. However $F^+ \circ F^*$ is a morphism of C, so by the above

there exists a morphism $F^{**}:C'' \to C^*$ such that $F^{**} \circ F^{+} \circ F^* = F^*$. This means that $F^{**} = (F^{+})^{-1}$, contradicting the fact that F^{+} is not injective. Hence F" is injective as claimed.

Since F" is injective and surjective we can define an equivalence relation $\backsim^*$ on C* by $v \backsim^* \tau \Leftrightarrow \exists k.F''^{k}(\tau) = v$ or $F''^{k}(v) = \tau$. If we (as usual) consider $C^*/\backsim^*$ as a P,Q-space with transitions $\overline{v} \xrightarrow{\delta} \overline{\tau} \Leftrightarrow \exists v' \in \overline{v}, \tau' \in \overline{\tau}.\ v' \xrightarrow{\delta} \tau'$ the map $G:C^* \to C^*/\backsim^*$ defined $G(\sigma) = \overline{\sigma}$ can be shown to be a morphism. Now $G \circ F^*$ is a morphism from C to $C^*/\backsim^*$, so by our earlier work there exists a morphism $G^*:C^*/\backsim^* \to C^*$ such that $G^* \circ G \circ F^* = F^*$. Hence G is injective, so all the equivalence classes of $\backsim^*$ have only one element. This is easily seen to imply that F" is the identity map from C* to C*.

We have thus shown that $F^* = I \circ F'$ (I the identity map on C*), which implies that F* is indeed the only morphism from C to C*. The uniqueness of the maps $G^*:Im(G) \to C^*$ follows immediately from the uniqueness of F*.

This result has a neat statement in category theory: C* is a terminal object in the category of onto morphic images of C (with morphisms as arrows).

The next result, which has the same type of proof, is an extension to the above.

8.15 Theorem

Suppose that C and D are P,Q-spaces and that $F:C \to D$ is a morphism, then if D* is the abstraction of D produced by 8.14 and $F^*:D \to D^*$ is the (unique) morphism between them, we have that there is a unique morphism from D to C*, and its image is isomorphic to C* (the abstraction of C produced by 8.14). Thus the compound map $F^* \circ F$ is independent of our choice of F.

Our next step will be to attempt to build up a "universal" P,Q-space into which all others can be mapped by a unique morphism. One can attempt to do this in two essentially different ways. The most obvious approach is to try to construct such a space from scratch. This would have the advantage that we would know exactly how it was constructed, making it easier to calculate with.

The other approach is non-constructive. Suppose we are given a set S of P,Q-spaces (for example a representative of each of the isomorphism classes of P,Q-spaces with less than or equal to any given cardinality), then let us form a "separated union" $\underline{S}$ of these spaces by attaching to each of its elements the name of the set from which it originally came. ($\underline{S} = \{(\sigma, C) \mid C \in S \ \& \ \sigma \in C\}$) If the space is given the transitions inherited from the elements of S ($(\sigma, C) \xrightarrow{a} (\rho, D)$ iff $C = D \ \& \ \sigma \xrightarrow{a} \rho$ in C), then $\underline{S}$ can be thought of as a P,Q-space containing a copy of each element of S as a subspace. The space $\underline{S}^*$ can be regarded as a canonical space for S, since each $C \in S$ trivially has a morphism into $\underline{S}$, and so by 8.15 there is a unique morphism from each element of S to $\underline{S}^*$.

If as suggested we take S to be a set of representatives of isomorphism classes of spaces with less than a given cardinality, then it is clear that there is a unique morphism from every P,Q-space with less than this cardinality to $\underline{S}^*$.

If it were possible to find a bound on the cardinalities of the spaces $\underline{S}^*$ (though we will shortly be able to deduce that it is not possible) then it would be easy to extend the above work to a production of a completely universal space. The above gives a sufficient taste of the type of methods which might be adopted to produce universal spaces in non-constructive ways. With a little more sophistication in category theory one might extend it further, but without further ado we will switch over to the more constructive approach.

Our idea of what a morphism is is of a map which preserves the exact shape of the possible behaviours of a process. Instead of analysing spaces directly through their morphisms as we have done hitherto it does not seem unreasonable to try to analyse them directly through their shapes of transition spaces. We can define a space (for any alphabet Σ) which attempts to record all the possible shapes of transitions up to depth n (for any $n \in N$).

$T_0 = \{\emptyset\}$ (there is only one possible shape of depth 0)

$T_{n+1} = \mathcal{P}((\Sigma \cup \{\cdot\}) \times T_n)$ (a possible shape of depth n+1 can be regarded as a relation between the possible transitions and the shapes of depth n).

It is easy to see that for all n $T_n \subseteq T_{n+1}$. Because of this we can regard T_n as a P,Q-space in the obvious way, with the transitions $\sigma \xrightarrow{\delta} \rho \Leftrightarrow (\delta,\rho) \in \sigma$. It is possible to show that every morphism of T_n is injective, which tells us that T_n is isomorphic as a P,Q-space to T_n^*.

8.16 Lemma

Every morphism of T_n is injective.

proof

We prove this by induction on n; the result is trivially true when n=0, since T_0 has only one element.

Suppose true for T_n, and that $F:T_{n+1} \to C$ is a morphism for some space C. T_n is a subspace of T_{n+1}, so by induction F is injective on this subspace.

Suppose that σ and ρ are two distinct elements of T_{n+1}. There must be some $\delta \in \Sigma \cup \{\cdot\}$ and $\tau \in T_n$ such that (δ,τ) is contained in $\sigma - \rho$ (without loss of generality). We must show that $F(\sigma) \neq F(\rho)$. Since F is a morphism we have $F(\sigma) \xrightarrow{\delta} F(\tau)$; if $F(\sigma) = F(\rho)$ then there would be some $\gamma \in T_{n+1}$ s.t. $\rho \xrightarrow{\delta} \gamma$ and $F(\gamma) = F(\tau)$. However $\rho \xrightarrow{\delta} \gamma \Rightarrow \gamma \in T_n$, and F is injective on T_n, which implies that $\gamma = \tau$, contradicting the fact that $(\delta,\tau) \notin \rho$. Thus F is injective, completing our inductive proof.

The next step is to define functions which, given a process, produce the representation of its depth n behaviour. Given a P,Q-space we can expect to find a function $H_n^C : C \to T_n$ for each $n \in N$.

$$H_0^C(\sigma) = \emptyset$$
$$H_{n+1}^C(\sigma) = \{(\delta, H_n^C(\rho)) \mid \sigma \xrightarrow{\delta} \rho\}$$

8.17 Lemma

a) If D is a subspace of C and $\sigma \in D$, then the values $H_n^D(\sigma)$ produced relative to D are the same as those $H_n^C(\sigma)$ produced relative to C.

b) If $F:C \to D$ is a morphism then $H_n^C(\sigma) = H_n^D(F(\sigma))$ for all $\sigma \in C$.

c) If $F:C \to D$ is a morphism and $H_n^C(\sigma) \neq H_n^C(\rho)$ then we can be certain that $F(\sigma) \neq F(\rho)$.

The proofs of (a) and (b) are straightforward deductions from our definitions. Part (c) is a corollary to part (b).

One of the effects of 8.17 is that it allows us to ignore the superscript "C" in the notation $H_n^C(\sigma)$ without being likely to introduce errors. Henceforth we will omit it on the understanding that it could be inserted if desired.

8.18 Lemma

If we regard the spaces T_n as P,Q-spaces in the manner described earlier then the following are true:

a) If $\sigma \in T_n$ and $m \geqslant n$ then $H_m(\sigma) = \sigma$.

b) For any space C and $\sigma \in C$ we have $H_n(H_m(\sigma)) = H_k(\sigma)$ for all n,m,k s.t. $k=\min(n,m)$.

The proofs of these results are just easy inductions.

One might hope that since we have established (8.17(c)) that two processes mapped to different elements of any T_n must be kept separate by any morphism, there might be some reverse implication: if two processes are mapped to the same element of T_n for each n then they can be identified by some morphism. With this hope in mind we can define a space of behaviour spaces T_ω, which is the set of sequences of elements from the T_i which match up under the functions H_i.

$$T_\omega = \{(\sigma_0, \sigma_1, \ldots, \sigma_i, \ldots) \mid \forall i.\, \sigma_i \in T_i \ \&\ H_{i+1}(\sigma_{i+1}) = \sigma_i\}$$

(T_ω is just the inverse limit of the spaces T_i with projection functions H_i.)

We can easily define a function $H_\omega : C \to T_\omega$ for any P,Q-space C by $H_\omega(\sigma) = (H_0(\sigma), H_1(\sigma), .., H_i(\sigma), ..)$. The space T_ω is naturally made into a P,Q-space by the transitions $\underline{\sigma} \overset{\xi}{\to} \underline{\rho} \Leftrightarrow \forall i.\, \sigma_{i+1} \overset{\xi}{\to} \rho_i$ (where σ_i represents the ith component of the sequence $\underline{\sigma}$).

8.19 Lemma

If we regard T_ω as a P,Q-space as above, then:

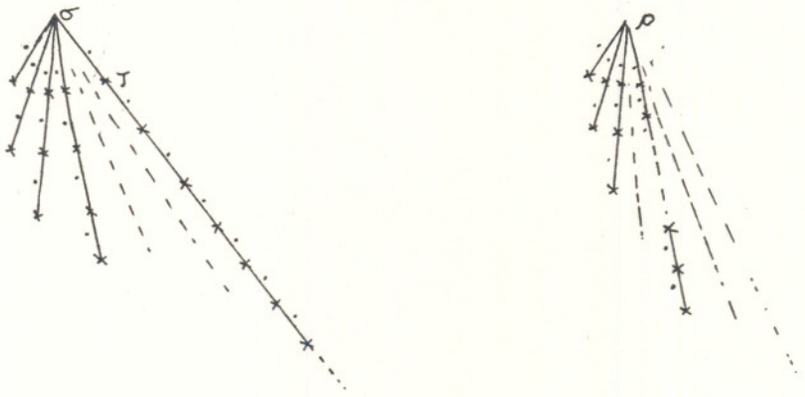
a) $H_n(\underline{\sigma}) = \sigma_n$; $H_\omega(\underline{\sigma}) = \underline{\sigma}$ for all $\underline{\sigma} \in T_\omega$.

b) Every morphism of T_ω is injective.

The proof of (a) is an induction on n; part (b) follows by part (a) and 8.17(c).

Note that (a) implies that for any space C and $\sigma \in C$ we have $H_n(H_\omega(\sigma)) = H_n(\sigma)$ for all n, and $H_\omega(\sigma) = H_\omega(H_\omega(\sigma))$.

One might hope that the map H_ω is a morphism; unfortunately however this is not so. This is because even though two processes have the same shapes of behaviour for all finite depths it does not mean that they are sufficiently similar to be identified by a morphism. As an example consider the two processes represented by the following diagram:



"σ" has the potential to perform an infinite sequence of internal actions whereas "ρ" does not. There can be no morphism F which identifies σ and ρ, for otherwise there would be some "γ" such that $\rho \stackrel{\cdot}{\to} \gamma$ and $F(\gamma) = F(\jmath)$. It is easy to prove inductively that this cannot be so. It is also not hard to verify that $H_n(\sigma) = H_n(\rho)$ for all $n \in N$, which implies that $H_\omega(\sigma) = H_\omega(\rho)$.

It is an easy consequence of our results that the higher the index of the function H_α, the more distinctions it makes. It is not unnatural to define such functions for higher ordinals than ω, therefore. For <u>countable</u> ordinals this can be done in a very similar way to the above. We define spaces T_α and projection functions H_α by mutual recursion.

$T_0 = \{\emptyset\}$

$H_0(\sigma) = \emptyset$

$\emptyset$ has no transitions;

$T_{\alpha+1} = \wp((\Sigma \cup \{\cdot\}) \times T_\alpha)$

$H_{\alpha+1}(\sigma) = \{(\delta, H(\rho)) \mid \sigma \stackrel{\delta}{\to} \rho\}$

$T_{\alpha+1}$ has transitions $\sigma \stackrel{\delta}{\to} H_{\alpha+1}(\rho)$ if $(\delta, \rho) \in \sigma$

$T_\lambda = \{\underline{\sigma} \mid \underline{\sigma}$ is a λ-sequence of elements of $\bigcup_{\beta<\lambda} T_\beta$ s.t.

$\beta < \lambda \Rightarrow (\sigma_\beta \in T_\beta \;\&\; ((\gamma < \beta) \Rightarrow H_\gamma(\sigma_\beta) = \sigma_\gamma))\}$

$H_\lambda(\sigma) = \underline{\rho}$, where $\rho_\alpha = H_\alpha(\sigma)$ for $\alpha < \lambda$.

$\underline{\sigma}$ has transition $\underline{\sigma} \stackrel{\delta}{\to} \underline{\rho} \Leftrightarrow (\delta, \rho_\gamma) \in \sigma_{\gamma+1}$ for all $\gamma < \lambda$.

For countable α it is possible to prove several results about T_α and H_α which are extensions of our earlier ones. Note that (b) below is necessary for the well-definedness of H_λ (λ limit ordinal) in that $H_\lambda(\sigma)$ is only an element of T_λ if $H_\beta(H_\alpha(\sigma)) = H_\beta(\sigma)$ for all $\beta<\alpha<\lambda$.

8.20 Theorem

If we regard each of the spaces T_α as a P,Q-space with the transitions described above then each of the following holds.

a) If $\sigma \in T_\alpha$ then $H_\alpha(\sigma) = \sigma$.
b) If C is a P,Q-space and $\sigma \in C$ then $\beta \leq \alpha \Rightarrow H_\beta(H_\alpha(\sigma)) = H_\beta(\sigma)$.
c) If λ is a limit ordinal and $\underline{\sigma} \in T_\lambda$ then $\alpha<\lambda \Rightarrow H_\alpha(\underline{\sigma}) = \sigma_\alpha$.
d) Every morphism of T_α is injective.
e) If $\sigma \in C$ and $F:C \to D$ is a morphism, then $H_\gamma(\sigma) = H_\gamma(F(\sigma))$.

These results can all be proved by technical manipulations, some of which (in the author's proof) depend on the fact that every countable limit ordinal λ has an ω-sequence of ordinals $\langle \alpha_i \mid i \in N \rangle$ such that $\forall i . \alpha_i < \lambda$ and $\bigcup_{i=0}^{\infty} \alpha_i = \lambda$.

Note that we need only look to the function $H_{\omega+1}$ to separate the two processes described earlier (which were identified by H_ω but not by any morphism). We can however deduce from 8.20(d) and the fact that none of the T_α has a largest cardinality (among the T_α), that none of the functions H_α can be a morphism on every P,Q-space C (it is not one on $T_{\alpha+1}$).

Note also that even for the countable ordinals which we have used so far, the spaces T_α get very large indeed by normal standards.

The author's proof of 8.20(c) (which is vital to his proof of (a), (b) and (d)) breaks down at limit ordinals with cofinality greater than ω. At the time of writing he does not know whether it holds or not. Because of this it is not possible even to define the pairs (T_α, H_α) for any ordinal greater than or equal to $\omega_1+\omega$ (ω_1 the first uncountable ordinal) using our existing definition. It is however possible to adjust our definition (in such a way that if 8.20 does in fact hold for spaces defined in the old way for arbitrary α the two correspond) and obtain workable spaces for all ordinals.

We adopt the same definition as before except that when λ is a limit ordinal with cofinality greater than ω we take:

$T'_\lambda = \{\underline{\sigma} \mid \underline{\sigma}$ is a λ-sequence of elements of $\bigcup_{\alpha<\lambda} T_\alpha$ such that $\alpha<\lambda \Rightarrow (\sigma_\alpha \in T_\alpha \ \& \ ((\beta<\alpha) \Rightarrow H_\beta(\sigma_\alpha) = \sigma_\beta))\}$

$H_\lambda(\sigma) = \underline{\rho}$, where $\rho_\alpha = H_\alpha(\sigma)$ for $\alpha<\lambda$.

In T'_λ $\underline{\sigma}$ has transitions $\underline{\sigma} \xrightarrow{\delta} \underline{\rho}$ if $(\delta, \rho_\gamma) \in \sigma_{\gamma+1}$ for all $\gamma<\lambda$.

Now let $T_\lambda = \{H_\lambda(\underline{\sigma}) \mid \underline{\sigma} \in T'_\lambda\}$, and redefine the transitions to be those of T'_λ with both states in the restricted space T_λ. (i.e. $\underline{\sigma} \xRightarrow{\delta} \underline{\rho}$ in T_λ if $\underline{\sigma}, \underline{\rho} \in T$ and $\underline{\sigma} \xRightarrow{\delta} \underline{\rho}$ in T'_λ.)

8.21 Theorem

With the definition given above each of the clauses of 8.20 holds for spaces T_α and functions H_α defined for arbitrary ordinals α. In addition H_α is a well-defined function from every P,Q-space to T_α for every α (this is not quite obvious in the case of non-ω-cofinal limit ordinals).

This result has a long and technical proof which is just an expansion of the one for 8.20.

Because clause (d) of 8.20 carries over to the general case we must give up all hope of finding a completely universal P,Q-space, as we can now find spaces with arbitrarily large cardinality whose only morphisms are injective. One can however prove that given any P,Q-space C then for sufficiently large ordinals α the maps $H_\alpha : C \to T_\alpha$ are morphisms. Given a P,Q-space C we can define a cardinal I(C) which is the index of non-determinism of C by

I(C) = smallest infinite regular cardinal strictly greater than that of $\{\rho \in C \mid \sigma \xRightarrow{\delta} \rho\}$ for every $\sigma \in C$ and $\delta \in \Sigma \cup \{\cdot\}$.

Thus $I(C) = \aleph_0$ if and only if there is no $\sigma \in C$ and $\delta \in \Sigma \cup \{\cdot\}$ s.t. $\{\rho \in C \mid \sigma \xRightarrow{\delta} \rho\}$ is infinite.

8.22 Theorem

If C is a P,Q-space and α is an ordinal such that $|\alpha| \geq I(C)$ then $H_\alpha : C \to T_\alpha$ is a morphism.

The proof of this result is not very difficult.

Corollaries to 8.22 are the facts that if C is finite-branching in the sense $I(C) = \aleph_0$ then the map $H_\omega : C \to T_\omega$ is a morphism, and if $I(C) = \aleph_1$ then $H_{\omega_1} : C \to T_{\omega_1}$ is a morphism (ω_1 the smallest uncountable ordinal).

8.22 tells us that for every alphabet Σ and cardinal $\aleph$ we can construct a space U into which there is a morphism from every space with this alphabet whose index of non-determinism does not exceed $\aleph$. The fact that these morphisms into U are unique is implied by the fact that all morphisms of U are injective, which means that U is isomorphic to U*, so we can apply 8.15. Thus U can be regarded as a universal space (in the sense described earlier) for a restricted class of spaces.

In most cases this universal space is much bigger than is necessary. As an example of this consider the result of constructing the universal space for finite branching processes over a countably infinite alphabet Σ. The successive approximations T_i to this space have cardinal $\beth_i$, where $\beth_0 = 1$; $\beth_1 = 2^{\aleph_0}$, and $n \geqslant 1 \Rightarrow \beth_{n+1} = 2^{\beth_n}$. The cardinal of the space T_ω is at least as large as the least upper bound of these cardinals. It is possible to cut this down very considerably by considering only those elements of T_n which can be the image under H_n of a finitely branching process. Let us define subspaces T_n^* and T_ω^* of T_n and T_ω respectively by the following:

$$T_0^* = \{\emptyset\}$$

$$T_{n+1}^* = \{X \subseteq (\Sigma \cup \{\cdot\} \times T_n^*) \mid \forall \delta.\ \{\sigma \mid (\delta, \sigma) \in X\} \text{ is finite}\}$$

$$T_\omega^* = \{\underline{\sigma} \in T \mid \forall n.\ \sigma_n \in T_n^*\}.$$

It is clear that T_ω^* is indeed a subspace of T_ω, and that the image under H_ω of any C such that $I(C) = \aleph_0$ is contained in T_ω^*. We can therefore regard T_ω^* as a universal space for finitely branching processes. The cardinals of the spaces T_n^* are successively $1, 2^{\aleph_0}, 2^{\aleph_0}, 2^{\aleph_0}, \ldots$, and the cardinal of T_ω^* is $2^{\aleph_0}$.

In fact T_ω^*, though it has minimal possible cardinal with respect to being a universal space for finitely branching processes, still contains elements which cannot be the image under H_ω of any element of a finite branching space. This can be seen by consideration of the processes quoted earlier as examples of processes identified by H_ω but not by any morphism. It is easy to see that their joint image under H_ω is an element of T_ω^*, but that this element of T_ω^* is one which cannot be the image of an element of a finite

branching space (there are infinitely many elements of T^*_ω to which it can be transformed by a single internal action). If we let U be the union of all the images in T^*_ω of finite branching spaces it is not hard to see that U is a (proper) subspace of T^*_ω. Furthermore U is itself a finite branching space, and so has a unique morphism into T^*_ω (which must be the identity morphism). Thus U is a universal space for all finite branching spaces, and there can be no smaller one. It is possible to construct U explicitly as follows:

Let $U_o = T^*_\omega$

$$U_{n+1} = \{\sigma \in T^*_\omega \mid (\forall \delta.\exists n \in N.\forall m.|\{\rho \mid (\delta,\rho) \in \sigma_m\}| < n)\}$$
$$\cap \{\sigma \in T^*_\omega \mid \forall \delta.\forall \rho.\ \sigma \xrightarrow{\delta} \rho \Rightarrow \rho \in U_n\}$$

$$U = \bigcap_{n=o}^{\infty} U_n$$

There is no reason why "T_ω" should not be used in place of "T^*_ω" in either of the above constructions of U, since we would have got the same answer. The only advantage gained from using T^*_ω is that it gives us a much better bound on the cardinal of the space U. It is also possible to use either of the above tricks to obtain a universal space U for P,Q-spaces with any bound on their index of non-determinism, and any size of alphabet Σ, such that U falls into the class of spaces which it models.

So far we have only proved the non-existence of completely universal spaces as a corollary to the difficult construction we devized for partially universal spaces. It does in fact have quite an easy proof by contradiction.

8.23 Theorem

For no alphabet Σ can there be a P,Q-space U to which there is a morphism from every P,Q-space with this alphabet.

proof

Suppose to the contrary that such a U does exist for some alphabet Σ. As we have seen U* would have a unique morphism to it from every P,Q-space with the given alphabet, and also every morphism of U* is injective. Consider the P,Q-space $V = (\{a\} \times U^*) \cup (\{b\} \times \wp(U^*))$ $(a \neq b)$, which has transitions

$(a,\sigma) \xrightarrow{\delta} (a,\rho)$ iff $\sigma \xrightarrow{\delta} \rho$ in U*;

$(b,X) \dashrightarrow (a,\sigma)$ iff $\sigma \in X$;

and no others.

Since V is a P,Q-space with alphabet Σ there must be some (unique) morphism $F:V \to U^*$. The cardinal of $\wp(U^*)$ is strictly greater than that of U^*, so there must be two distinct subsets X and Y of U^* such that $F(b,X) = F(b,Y)$. We may without loss of generality assume that there is some $\sigma \in X - Y$. It is easy to prove from the definition of morphisms that there must be some $\rho \in Y$ such that $F(a,\rho) = F(a,\sigma)$. Thus F is not injective on the subspace $\{a\} \times U^*$ of V. However this contradicts the fact that this subspace is isomorphic to U^*, since all morphisms of U^* are injective.

We may thus conclude that, as claimed, no such U can exist.

We have already demonstrated that we can, amongst all the P,Q-spaces with any given bound on their index of non-determinism, find one which is universal. Sometimes we will wish to make additional assumptions about the type of P,Q-space we are using, and when we do this it will be useful to be able to construct a universal space for spaces with this property (where possible). The following result shows one way in which this can be done.

8.24 Theorem

Suppose that "X" is a property of P,Q-spaces which satisfies the following laws:

(i) If C is a space with property X and $F:C \to D$ is a morphism, then Im(F) has property X.

(ii) If C is a space with subspaces $\{C_\alpha | \alpha \in A\}$ such that $C = \bigcup_{\alpha \in A} C_\alpha$ and each C_α has property X, then C also has property X.

then we may conclude that for each cardinal $\aleph$ there exists a P,Q-space U^X such that U^X has property X, $I(U^X) \leqslant \aleph$, and such that whenever C is a space with property X such that $I(C) \leqslant \aleph$ there exists a unique morphism $F:C \to U^X$.

proof

Let U be a universal space for spaces such that $I(C) \leqslant \aleph$ (for example T_α, where α is the initial ordinal with cardinal $\aleph^+$). Define U^X to be the union of all the images in U of spaces C s.t. $I(C) \leqslant \aleph$ and C has property X. It is easily seen that U^X, when so defined, satisfies all that is required of it.

In the last few pages we have developed a calculus which allows us to relate P,Q-spaces by means of maps called "morphisms", which are in some sense behaviour preserving maps between them. We have succeeded in producing universal spaces which can be held to model large classes of spaces in unique ways. The spaces we have been studying are in essence simple relational structures, which in various guises are used throughout mathematics. There is therefore much similarity between the above work and that of other authors which the author is aware of, and probably more with other work with which he is not familiar. For example the concept of "operational equivalence", as introduced in Hennessy and Milner ( ) and other works, is extremely similar to the equivalence induced by the operator H_ω. The chief difference is in the treatment of internal actions. Different types of "morphisms", similarly defined, can be used to analyze other, rather more complicated, types of process-spaces. For example this can be done (with almost exactly the same effect) for spaces where the relations represent finite sequences of visible actions, when given a suitable axiomatization.

Having built up enough machinery for our own purpose we are in a position to return to the main theme of this chapter, and to find out how the spaces we have constructed relate to the models we used in earlier chapters. We are essentially seeking functions "θ" such that, given processes "c" in a P,Q-space "C", the "c"s are mapped in a useful and realistic way to one of our models by the map $\Phi(c) = \{\theta(\underline{a}) \mid \underline{a} \in \beta(c)\}$. Our intuition about the models is that they represent some aspect of the observable behaviour of processes. Given a behaviour, which is an abstraction of one possible sequence of interactions between process and state, we must ask just which parts of it are observable by an experimenter who manipulates the environment. The two things which are certain to be observable are the environment component "X" of the triples making up behaviours, and any external actions which occur. We can also assume that the experimenter is aware of the postulates (general properties of processes) which processes satisfy, so that for example if he can be sure that no action has occurred

for sufficiently long (while one set, X, was offered) he can deduce that no action will ever occur (if he persists in offering X). We will assume that the internal state "σ" of a process is invisible to the experimenter. The only question to be decided is whether or not the experimenter can observe the presence or absence of internal actions. One might imagine that there is a light on the side of a machine which lights up when there is internal activity. We cannot expect the experimenter to be able to discern anything more about internal activity than its presence, in the loosest sense (if he can detect it at all): for example it does not seem reasonable to expect him to be able to count the number of internal actions which occur. We will find that there are two distinct maps from P,Q-spaces to the non-deterministic model, the choice between them being largely dependent on whether or not we believe internal activity to be observable.

The principle that the internal state of a process is invisible is important in justifying the use of morphisms and universal spaces. This is because of 8.10 which tells us that apart from the state components of behaviour, the behaviours of a process and its image under a morphism are identical. By this principle it seems reasonable to expect that the function "θ" which maps behaviours to their representations will be independent of the "state" components of behaviours. We will therefore expect it to be induced in the natural way by some function of $(\wp(\Sigma)\times(\Sigma\cup\{\cdot,-,*\}))^{\circledast}$ (= H, say). In future when "θ" is a function of H we will regard "$\hat{\theta}$" as being its natural extension to behaviours. This is formed by defining a projection function h to be the natural extension to sequences of the function $h(\sigma,X,\delta) = (X,\delta)$. "$\hat{\theta}$" then becomes $\theta\circ h$. If C and D are two P,Q-spaces then whenever F is a morphism from C to D and θ is a function of H we have $\{\theta(\underline{a}) \mid \underline{a}\in B(c)\} = \{\theta(\underline{a}) \mid \underline{a}\in B(F(c))\}$ for all $c\in C$.

An extension of this principle is to decree that not only will our modelling functions be independent of states, but also the correctness conditions "χ" of behaviours which we wish to prove will be independent of states. This can be

interpreted as saying that we shall judge our processes only by what they do (or fail to do) either internally or externally, and not by how they are actually constructed. We will thus generally expect our predicates of behaviour to be induced in the natural way by predicates of H. If "X" is a predicate of H we will write "$\hat{X}$" for the predicate which it induces on whatever space of behaviours we are currently using, ($\hat{X}(\underline{a}) \equiv X(h(\underline{a}))$). It is easy to see that if C and D are P,Q-spaces and X is a predicate of H then $\underset{\sim}{\hat{X}}(c) \Leftrightarrow \underset{\sim}{\hat{X}}(F(c))$ for all $c \in C$, whenever $F:C \to D$ is a morphism.

We may thus conclude that, so long as our functions and predicates are "forgetful" of internal states, both a process' image in a model and the truth of predicates about them are invariant through morphisms. Thus in these circumstances, to prove a predicate of a process "c" from any space C it is sufficient to prove the corresponding predicate of the process' image in any suitable universal space.

When we are using functions and predicates of this "forgetful" type, the set of the projections into H of the behaviours of a process is very important. If we define $B^*(c)$, the "reduced behaviour set", of $c \in C$ to be $\{h(\underline{a}) \mid \underline{a} \in B(c)\}$, it is easy to see that for any function θ of H we have $\{\hat{\theta}(\underline{a}) \mid \underline{a} \in B(c)\} = \{\theta(\underline{a}) \mid \underline{a} \in B^*(c)\}$. Similarly when X is a predicate of H we get $\underset{\sim}{\hat{X}}(c) \Leftrightarrow \forall \underline{a} \in B^*(c).X(\underline{a})$. Note that $B^*(c) \subseteq B^*(d)$ & $\underset{\sim}{\hat{X}}(d) \Rightarrow \underset{\sim}{\hat{X}}(c)$. Both the image of a process in a model and the truth of predicates "$\underset{\sim}{\hat{X}}$" about it are determined completely by its reduced behaviour set.

Before we get involved in maps to the non-deterministic model it is perhaps wise to see how we might use the machinery we have set up to construct and analyse maps from P,Q-spaces to the deterministic model P. There is really only one map worth considering, namely that induced by the following function "θ" of H:

$\theta(\langle\rangle) = \langle\rangle$; if $\underline{a}$ is finite then

$\theta(\langle\langle(X,\delta)\rangle \underline{a}) = \theta(\underline{a})$ if $\delta \in \{\cdot, -, *\}$ and $= \langle\delta\rangle\theta(\underline{a})$ if $\delta \in \Sigma$;

$\theta(\underline{a}) = \langle\rangle$ if $\underline{a}$ is infinite.

It is easy to show from our postulates that the "Φ" induced by $\hat{\theta}$ satisfies the following, when regarded as a function from a P,Q-space C to $\wp(\Sigma^*)$.

(i) $\Phi(c)$ is non-empty for all $c \in C$

(ii) $\Phi(c)$ is prefix closed for all $c \in C$.

To establish the finality of "$\surd$" we would have to make some additional postulate of our spaces such as

(D1) $(\sigma, X) \overset{\surd}{\Rightarrow} \rho \Rightarrow \neg((\rho, Y) \overset{\phi}{\Rightarrow} \tau)$.

It is easy to see that D1, when regarded as a property of P,Q-spaces, satisfies the hypothesis of 8.24. There is thus no difficulty in constructing universal spaces for P,Q-spaces which additionally satisfy D1. Let us call a P,Q-space which satisfies D1 a D1-space. It is easy to show that when C is a D1-space we have

(iii) $w\langle\surd\rangle v \in \Phi(c) \Rightarrow v = \langle\rangle$ for all $c \in C$.

Thus Φ is a well-defined map from every D1-space to the deterministic model P.

P can itself be regarded as a D1-space. Transitions are defined: $A \overset{a}{\rightarrow} B$ iff $B = A$ after $\langle a \rangle$; no internal transitions. It is a simple matter to prove $\Phi(A) = A$ for all $A \in P$. One easy consequence of this fact is that Φ is a surjective function from the universal finite branching D1-space to P. For the time being let us adopt this universal space as the "C" which we are trying to model by P.

The first thing which we must investigate, when studying the relationship between a "real" system and a model, is the way in which predicates which we wish to prove of the real system transfer to the model. Intuitively one might suspect that the map Φ is adequate for expressing many partial correctness conditions (those which demand that anything which a process actually does is correct), but is poor when it comes to total correctness conditions (which demand that a process must actually be willing to do things).

Because of the universal nature of the space C it is not hard to show that given any $c \in C$ there exists some (unique) element d of C whose only transitions are $d \dot{\rightarrow} c$ and $d \dot{\rightarrow} e$, where e is a process with no transitions. It is easy to see that $\Phi(c) = \Phi(d)$ (because they have the same possible

sequences of external actions). However it is also easy to see that $B^*(d) \supseteq B^*(f)$, where f is the element of C with a unique transition $f \stackrel{\cdot}{\rightarrow} e$. Suppose now that X is any predicate of H such that $(\hat{X})'$ (the weakest predicate Π of P s.t. $\forall c \in C.\Pi(\Phi(c)) \Rightarrow \hat{X}(c)$) is satisfiable. Suppose A is chosen so that $(X)'(A)$ holds, and that $c \in C$ is such that $\Phi(c) = A$ (such a "c" exists since Φ is surjective). Now construct "d" as above. Since $\Phi(c) = \Phi(d)$ holds we have $(\hat{X})'(\Phi(d))$; this implies $\hat{X}(d)$, which in turn implies $\hat{X}(f)$ (because $B^*(d) \supseteq B^*(f)$).

We are thus forced to conclude that whenever "$\hat{X}$" is a predicate of C sufficiently weak to allow it to be deduced of any process from the process' image in P, it must itself be satisfied by the process "f" which can only perform one (internal) action before deadlocking. This confirms the suspicion which we developed in chapter one, that the model P is not adequate for telling us anything reliable about potentially non-deterministic processes. Some indications were given in chapter one about the type of process which we felt was adequately described by P. Without going into any more detail on the modelling of the above system C let us now try to restrict our object space to processes which we can model accurately over P.

We wish to axiomatize "deterministic" behaviour. We might expect the chief sources of non-determinism to be firstly internal actions (which can "resolve" non-determinism) and secondly cases where one state has more than one external action with a particular name. The postulate which expresses the proscription of these types of behaviour is the following:

(D2) $\neg((\sigma,X) \stackrel{\cdot}{\rightarrow} \rho) \;\&\; (((\sigma,X) \stackrel{a}{\rightarrow} \rho) \;\&\; ((\sigma,X) \stackrel{a}{\rightarrow} \tau) \;\Rightarrow\; \rho = \tau)$

This is another property which satisfies the hypotheses of 8.24; thus so is the joint condition D1 & D2. We may thus deduce the existence of a universal D-space, where a D-space is defined to be a P,Q-space which satisfies D1 and D2. (The existence of a completely universal space follows from the fact that any space which satisfies D2 is automatically finite-branching). It is possible to weaken condition D2

slightly to allow a little internal behaviour. We can reasonably allow a deterministic process to have internal actions if they are all single (for each σ there is at most one ρ such that $\sigma \stackrel{\cdot}{\rightarrow} \rho$) and inevitable (if $\sigma \stackrel{\cdot}{\rightarrow} \rho$ is possible then $\sigma \stackrel{a}{\rightarrow} \tau$ is not). A modified D2 which expresses this is

(D2') $(((\sigma,X) \stackrel{\delta}{\rightarrow} \rho) \ \& \ ((\sigma,X) \stackrel{\delta}{\rightarrow} \tau) \ \Rightarrow \ \rho=\tau)$
$\& \ \neg(((\sigma,X) \stackrel{\cdot}{\rightarrow} \rho) \ \& \ ((\sigma,X) \stackrel{a}{\rightarrow} \tau))$.

D2' is also a condition which satisfies the hypotheses of 8.24, so in a similar manner to the above we may deduce the existence of a universal D'-space (a P,Q-space which satisfies D1 and D2'). Note that every D-space is a D'-space.

Clearly P, when made into a D1-space as before, is a D-space. We can deduce from this that Φ is a surjective function to to P from each of the universal D-space and the universal D'-space.

It is left as an exercise for the interested reader to verify that in either of the above types of space the set of predicates we can reasonably expect to prove by reference to the model is much larger and more useful than in the earlier case. It is worthwhile to make two remarks however. Firstly it is not hard to show that the universal D-space is isomorphic to P (when P is regarded as a D-space in the usual way), and that Φ is a morphism from any D-space to P. Thus (recalling 8.1) it is not surprizing that predicates should transfer well between the two systems. Secondly, in the D'-space case, it is interesting to note that the function Φ identifies the processes "e", which has no actions, and "d", which has a single action $d \stackrel{\cdot}{\rightarrow} d$. ($\Phi$ also identifies any pair of processes whose structure is the same except for the substitution of "e" for "d" at some points, or vice-versa.) It thus identifies "divergence" or "infinite internal chatter" with "computed termination". Thus the absence of divergence is not expressible as a predicate of P which does not also imply freedom from deadlock. This issue (the difference between divergence and computed deadlock) will be more important later, when we come to consider the non-deterministic model.

Another interesting point which arises from the study of the relationship between these "real" systems and the model

P, is that P is very much a class 2 model for these systems. The most obvious way of seeing this is to note that if P were thought of as a class 1 model, this would mean that we only allowed ourselves monotonic predicates. We would then find ourselves in very much the same boat which we were in with the Dl-space, for every satisfiable predicate which we allowed ourselves would be satisfied by abort. The processes "e" and "d" (as defined in the last paragraph) would (as constant functions) be correct implementations of every operator.

The basic reasons for this arise from the structure of the function "Φ" in the following way. The function "Φ" is based on a function "$\hat{\theta}$" of behaviours which is essentially one-sided in that it only reflects one of the two aspects of behaviour which are essential to most total correctness predicates. It is based purely on the "positive" aspects of behaviour (triples of the form (σ,X,a) for $a\in\Sigma$), and not the equally important "negative" aspects (triples of the form $(\sigma,X,*)$ and infinite sequences of internal actions). The restrictive conditions which we have placed on D- and D'-spaces enable us to discover enough about the negative aspects of behaviour which are possible by studying the positive aspects. To do this we have to exploit the relationship which states that (in D- and D'-spaces) the more positive behaviours there are possible, the more "negative" behaviours can be deduced impossible. Thus, if we wish to check that an undesirable "negative behaviour" is impossible in a process by studying its image in P, it will often be necessary to check that some "positive" behaviour is possible. For example to ensure that a process c cannot deadlock on its first step it is necessary to check that $\Phi(c)^{O}\neq\emptyset$. It is because of this "upside-down exclusion" of negative behaviours that we cannot expect monotonic predicates to be sufficiently expressive, since by removing elements from an element of P we are adding possibly incorrect negative behaviours to its pre-image.

Having decided that our model is to be regarded as class 2 it is necessary that our implementations of the various operators and constructs of our language be exact. So far

we have been too involved with the construction and interpretation of our "real" systems to consider the problem of how we might seek to implement our language in them. What we would like is an operational semantics for our language. We have not got space here to go into this subject in very much detail; we will therefore quickly survey the various options open to us, draw a few general conclusions, and pass on to the study of the non-deterministic model.

There are several ways in which one might seek to give an operational semantics to our language; some of these are more abstract than others. One approach would be to define exactly what was meant by the "state" of a process: how it stores and recalls the values of its various variables, and how it decides which actions to take (with what influence on itself). If one did this it would be necessary to check that the space of states which resulted satisfied whatever postulates were required of them. Without going into technical detail it is only possible to make a few general remarks about this approach.

(i) We cannot expect an operator to be able to see any more about its operands than an environment could. It is not reasonable to expect an operator to be able to predict what its operands will do after actions which have not yet been completed (nor to anticipate divergence). One useful way to think of an operator is as a "black box" which is placed around its operands and which has certain powers over them, for example:

(a) An operator can "switch on" or "switch off" its operands. On switching an operand on for the first time the operator must initialize all its variables. Only "on" operands may perform any action. The act of switching will itself be an internal action of the total state.

(b) Any internal action performed by an operand is an internal action of the total state, and uncontrollable by the operand.

(c) The operator can act as environment to its operands and communicate with them without telling its own environment. Any such communications are internal actions of the total state.

(d) An operator can communicate with its environment without reference to its operands.

(e) An operator can regulate communication between the environment and its operand(s). For example it might transform the alphabet in some way, synchronize several of its operands, or become completely transparent.

(If we tried to implement our existing operators as "black boxes", we might expect the above features to appear in the following in important ways:

(a) $[]$, a →, x:T →, recursion

(c) /X, ;

(d) a →, x:T →

(e) a., || .)

The above approach, while it might be considered to leave something to be desired, is a valuable aid to the intuition when considering the "reasonableness" of operators defined over universal spaces in abstract ways.

(ii) We would expect the values stored in process variables to be processes "switched off" (unevaluated code?) with the property that, when activated, they ignore any values assigned to their variables (except recursion parameters "λ") and adopt values stored with them in some way. There should be no problem in showing recursion to be well-defined, for all operators are in some sense "non-destructive" of actions because of principle (b) above, etc., and the act of making a recursive call is constructive because it involves "switching-on", which is an internal action.

In general each of the principles seems largely consistent with the idea that an operator (k-place) is a function from C'^{k} to C' (where $C' = C^{\Theta} \times S \to C$), as in the first part of this chapter.

Alternatively we might choose to define a semantics in a more abstract way. This could be done by direct reference to the structure of universal spaces (8.16 - 8.24). This should be done bearing the above principles carefully in mind with a view to later implementation by the more practical methods above. This is the most obvious approach to

adopt when we are using P,Q-spaces as idealized models of other spaces. Because of the nature of universal spaces it is possible to define elements uniquely by their transitions. Examples of the ways one might wish to do this are the state "$a \to \sigma$", which is defined to have a single transition, namely "a" after which it becomes σ, and "$\sigma;\rho$" which has its transitions recursively defined $\sigma \xrightarrow{\delta} \tau \Rightarrow (\sigma;\rho) \xrightarrow{\delta} (\tau;\rho)$ $(\delta \neq \surd)$

$\sigma \not\to v \Rightarrow (\sigma;\rho) \not\to \rho$.

The first type of operator is easy to define: if (as is common) the universal space in use is the subspace U of some T_λ (λ a limit ordinal) defined through 8.24 by some property X, then we can define "$a \to \sigma$" as follows. We take it to be the element ρ of T_λ such that $\rho_{\gamma+1} = \{(a,\sigma_\gamma)\}$ for all $\gamma \in \lambda$ (the other components can be deduced from these). If U is a proper subspace of T_λ then we are obliged to show that the ρ so defined is an element of U. This can be done in this case by checking that the space U', which is U with an extra element "ρ'" adjoined with the single transition $\rho' \xrightarrow{a} \sigma$, has the defining property X. If this is the case then there is a morphism from U' to U, and the image of ρ' is ρ.

The second type of operator requires a little more work. To define "$\sigma;\rho$" explicitly we have to appeal to transfinite recursion. Let us once again suppose that U is a subspace of T_λ. We define $(\sigma;\rho)_0 = \emptyset$

$$(\sigma;\rho)_{\gamma+1} = \{(\delta,(\tau;\rho)_\gamma) \mid \sigma \xrightarrow{\delta} \tau \ \&\ \delta \neq \surd\} \cup \{(\cdot,\rho_\gamma) \mid \sigma \not\to v\}$$

$$((\sigma;\rho)_{\lambda'})_\gamma = (\sigma;\rho)_\gamma \qquad (\gamma \in \lambda') \quad .$$

Once again to show that ";" is a well-defined operator on U (if it is) we take a space U' which includes a copy of U and a disjoint copy of U×U. The transitions of the elements of the copy of U are those inherited from U. A pair (σ,ρ) has transitions $(\sigma,\rho) \xrightarrow{\delta} (\tau,\rho)$ if $\delta \neq \surd$ and $\sigma \xrightarrow{\delta} \tau$ in U

$(\sigma,\rho) \not\to \rho$ if $\sigma \not\to v$ in U.

One must show that U' has the defining properties of U, so that there is a morphism from U' to U. If this is so the pair (ρ,τ) can be shown to map under this morphism to $\rho;\tau$, the element of T_λ defined above.

Note that provided that the space U is closed under each of the above operators it is easy to prove the relations

$a \to (\rho;\tau) = (a \to \rho;\tau)$ and $((\sigma;\rho);\tau) = (\sigma;(\rho;\tau))$.

Other operators can be defined in very much the same way. This should always be done in a way which does not violate the principles derived from our "black box" discussion. The first of these is that at every stage the transitions of the result of applying an operator should depend only on the available transitions of the operands (in the states in which they currently find themselves), and past history; if a transition is executed which depends on the existence of some transition in one of the operands then the operand must execute that transition. The second principle is that every transition executed by an operand must be represented by a corresponding transition in the "finished product". These principles are guaranteed by stipulating that every operator must have some defining equation of the form below. A zero-place operator is a constant.

A one-place operator "op" must have exactly the transitions

$$op(\sigma) \xrightarrow{\delta} op_\alpha(\sigma) \qquad (\delta,\alpha) \in D$$
$$op(\sigma) \xrightarrow{\delta} op'_\alpha(\rho) \qquad \sigma \xrightarrow{\delta'} \rho \;\&\; (\delta,\delta',\alpha) \in D'$$

(The first line corresponds to the operator carrying out some action without reference to its operand, the second line corresponds to the operator transforming some action of its operand. In each case the operator may transform itself (non-deterministically) into another, dependent on the action which occurs.) One might wish to strengthen the above to ensure that internal actions of operands can neither become external actions of the finished product nor influence the composition of the operator. This can easily be done by editing the second line above.

A two-place operator "op" must have exactly the transitions

$$op(\sigma,\rho) \xrightarrow{\delta} op_\alpha(\sigma,\rho) \qquad (\delta,\alpha) \in D$$
$$op(\sigma,\rho) \xrightarrow{\delta} op'_\alpha(\tau,\rho) \qquad \sigma \xrightarrow{\delta'} \tau \;\&\; (\delta,\delta',\alpha) \in D'$$
$$op(\sigma,\rho) \xrightarrow{\delta} op''_\alpha(\sigma,\tau) \qquad \rho \xrightarrow{\delta'} \tau \;\&\; (\delta,\delta',\alpha) \in D''$$
$$op(\sigma,\rho) \xrightarrow{\delta} op^*_\alpha(\tau,\upsilon) \qquad \sigma \xrightarrow{\delta'} \tau \;\&\; \rho \xrightarrow{\delta''} \upsilon \;\&\; (\delta,\delta',\delta'',\alpha) \in D^*.$$

(The four lines here correspond to the operator carrying out some action without reference to its operands, transforming some action of its first operand, transforming some action of its second operand, and transforming and co-ordinating a pair of actions respectively.) Once again one

might choose to tighten up on the last three lines to ensure that internal actions of operands cannot have undue influence.

Three and higher place operators can be constructed by combining two-place operators.

(Note of clarification: in the above definitions it is the relations D, D' etc. which define the operators, together with the operators op_α, etc. which are assumed to be defined in the same way. "op_α" can vary with α, which is assumed to range over some indexing set.)

As an example a hiding operator might be defined by the scheme

$(\sigma/X) \doteq (\rho/X)$ if $\sigma \doteq \rho$ or $\sigma \xrightarrow{a} \rho$ for some $a \in X$;

$(\sigma/X) \xrightarrow{a} (\rho/X)$ if $\sigma \xrightarrow{a} \rho$ and $a \notin X$.

This and both our earlier schemes can easily translate to the form described formally above.

Note that we cannot expect the above hiding operator to be well-defined on any universal space U which does not reliably model all branching smaller than the smallest infinite cardinal greater than $\|X\|$.

The theory of operators defined in this way is quite interesting, but we do not have space to go into it in any depth. We merely quote the next result, which helps to formally justify our assertion that "properly defined" operators are in some sense non-destructive.

8.25 Lemma

Suppose that op is a k-place operator on a space U which is a subspace of some T_λ, and that op and all the other operators on which it depends in its definition are defined by the methods set out above, then if $\underline{\sigma}_i$ & $\underline{\sigma}'_i$ are two sets of elements of U such that $\forall i \in \{1,..,k\}.(\underline{\sigma}_i)_\gamma = (\underline{\sigma}'_i)_\gamma \quad (\gamma \in \lambda)$ we have $op(\underline{\sigma}_1,\ldots,\underline{\sigma}_k)_\gamma = op(\underline{\sigma}'_1,\ldots,\underline{\sigma}'_k)_\gamma$.

We need to be able to extend operators defined, as above, on a space U, to the space $U' = U^\Theta \times S \to U$. This can be done in very much the same way as before. Suppose that elements of U' are denoted e*, f*,.. and that elements of $U^\Theta \times S$ (states) are denoted π, o ,... . If we have a k-place operator op over

U then op can be re-defined as an operator $\underline{op}$ over U' by

$$\underline{op}(e_1^*,\ldots,e_k^*)(\pi) = op(e_1^*(\pi),\ldots,e_k^*(\pi)).$$

We also need operators defined over U' which are not simply extensions of operators over U. There are basically two categories of these: recursions and "others". Non-recursive operators should be defined by transition schemes similar to those used above. These should observe the general principle that nothing is observable of the contents of the U^Θ component of a "state" π without switching on any processes we wish to observe and treating them as "normal" operands. For most practical purposes one can get away with using zero and one-place operators of this type. We will therefore stipulate that any non-recursion operator not of the above form must be of one of the two forms set out below.

We will write an element π of $U^\Theta \times S$ as (π_1,π_2), π_1 being the U^Θ-component and π_2 being the S-component.

A zero-place operator $\underline{e}^*$ (constant element of U') must be defined:

$\underline{e}^*(\pi_1,\pi_2) = \rho$, where ρ has exactly the transitions

$$\rho \xrightarrow{\delta} \underline{f}_\alpha^*(\pi_1,\pi_2') \qquad (\delta,\pi_2,\pi_2',\alpha) \in D$$

$$\rho \dashrightarrow \pi_1(\theta) \qquad (\theta,\pi_2) \in D'$$

($\underline{f}_\alpha^*$ is assumed to be another zero-place operator, similarly defined.)

A one-place operator $\underline{op}$ must be defined:

$\underline{op}(e^*)(\pi_1,\pi_2) = \rho$, where ρ has exactly the transitions

$$\rho \xrightarrow{\delta} \underline{op}_\alpha(e^*)(\pi_1,\pi_2') \qquad (\delta,\pi_2,\pi_2',\alpha) \in D$$

($\underline{op}_\alpha$ is assumed to be another one-place operator defined in the same way or as an extension of a U-operator.)

As examples of these we can define the one-place operator "x:T → " and the zero-place operator "B" (call of the process variable B).

$(x{:}T \rightarrow e^*)(\pi) = \rho$, where ρ has exactly the transitions

$$\rho \xrightarrow{a} e^*(\pi[a/x]) \qquad a \in T(\pi_2).$$

(We assume that the set T may be a function of one or more non-process variables. The "$\underline{op}_\alpha$" used is the extension to U' of the identity function on U.)

$B(\pi) = \rho$, where ρ has the single transition

$\rho \xrightarrow{\cdot} \pi(B)$.

Existence proofs for all operators defined by any of our "transition scheme" methods can be carried out by extending the methods used for ";" and "a → " earlier. This will involve the setting up of a P,Q-space of syntactic/state objects, possibly including a copy of the space U as a subspace, then showing that the space one has set up satisfies enough for there to exist a (unique) morphism into U. If a definition of such an operator is required in the form given for "a →" and ";" (exact definition of the components of the result of an operator regarded as an element of T_λ) this can be done recursively.

If $U \subseteq T_\lambda$ then there is an obvious way in which one can define the projection π_γ of a "state" π into the space $(T_\gamma^\Theta)\times S$. The pay-off of all our careful definitions above is that, so long as e* is an element of U' defined only by operators of the types we allow, it can be proved that whenever π and π' are two "states" such that $\pi_\gamma = \pi'_\gamma$ we have $e^*(\pi)_{\gamma+1} = e^*(\pi')_{\gamma+1}$ for all $\gamma \in \lambda$. This is easily shown to imply that each recursive fixed-point equation has at most one solution. The existence of solutions to these equations can be proved without too much difficulty so long as the space U we are using satisfies some simple and natural closure conditions. As an example of a recursion one might use, to define "recB.e*" (e* defined using only permitted operators) we would say $(\text{recB}.e^*)(\pi) = \rho$ s.t. $\rho = e^*(\pi[\rho/B])$. The value of $\text{recB}.e^*(\pi)$ is defined as follows. We define a function f from $\lambda+1$ to U as follows:

$f(o)$ is chosen from U at random

$f(\gamma+1) = e^*(\pi[f(\gamma)/B])$

$f(\lambda')$ is chosen (by closure conditions) to be such that $f(\lambda')_\gamma = f(\gamma)_\gamma$ for all $\gamma \in \lambda'$.

The value of $\text{recB}.e^*(\pi)$ is $f(\lambda)$.

Other, more complex, recursive operators can be defined similarly.

The use of nested recursions (i.e. recursions within recursions) can also be justified without too much difficulty.

Conclusions for the deterministic model

We do not have space here to launch into any attempts to formally implement our operators over P (nor will we when we later examine the non-deterministic model). What we can do is to get a good impression of what is, and what it not, possible.

It is clear that unless we relax substantially our conditions upon operators there is little prospect of our being able to implement all our operators over D-spaces (where internal actions are banned). Operators which appear to be impossible because of their dependence on internal actions are ";", "/X", calls of recursive variables and meaningful recursion. Difficulty arises in a less expected way with the operator "□", when applied to processes whose initials are not disjoint. The obvious implementation scheme

$$\sigma \Box \rho \xrightarrow{a} \tau \quad \text{if } \sigma \xrightarrow{a} \tau \ (\& \neg\exists \nu.\ \rho \xrightarrow{a} \nu)$$

$$\sigma \Box \rho \xrightarrow{a} \tau \quad \text{if } \rho \xrightarrow{a} \tau \ (\& \neg\exists \nu.\ \sigma \xrightarrow{a} \nu)$$

$$\sigma \Box \rho \xrightarrow{a} \tau \Box \nu \quad \text{if } \sigma \xrightarrow{a} \tau \ \& \ \rho \xrightarrow{a} \nu$$

suffers from the drawback that the bracketed terms are not permissible within the rules which we set out earlier (essentially they would require the environment to be able to detect more about its operands than we have thought correct hitherto). If these offending terms were withdrawn then the operator would not preserve the postulate D2, for it would introduce multiple branching.

One might hope that D'-spaces, with their weaker postulates, might give us an easier ride. This is in some senses true, since it is now possible to correctly implement each of ";", "/X", process variables and recursion so long as syntactic rules similar to those set out in chapter one are observed. Problems still arise with the "□" operator, however, and of a more serious nature than before. This results from the fact that the map "Φ" identifies deadlock with divergence. Consider for a moment the situation which will arise when we try to compose a simply diverging process with a process which can perform some action, "a" say. Since the operator is unable to detect that its operand is diverging (it cannot know that it will not decide to perform some visible action or halt at some future point) it must allow it to perform

its successive internal actions. These will be reflected in internal actions of the resultant process, so it must itself be able to diverge. It is easily seen that this is impossible in any element of a D'-space which can execute the transition "a". We must conclude that some strong syntactic condition is required to ensure that "$\Box$" is implementable. Such a condition is the requirement that "$\Box$" only be used in the context "$(a \rightarrow *) \Box (b \rightarrow *)$" where $a \neq b$. The (very desirable) use of more general guards on the two sides of "$\Box$", such as "x" (alphabet variable) or "x:T" (input) would create problems because of the requirement that guards should be distinct.

It is possible to invent a third type of "deterministic" P,Q-space where many problems disappear. Unfortunately it is not quite so natural as the other two. One postulates that branching is finite, that divergence is absent, and that while multiple branching and internal actions may occur they may not influence the external actions.

(D2") $\forall\sigma.\forall\delta.\forall X.\ \{\rho \mid (\sigma,X) \stackrel{\delta}{\rightarrow} \rho\}$ is finite

$\&\ \forall\delta.\forall\sigma.\forall X.\forall\rho,\tau.\ (\sigma,X) \stackrel{\delta}{\rightarrow} \rho\ \&\ (\sigma,X) \stackrel{\delta}{\rightarrow} \tau \Rightarrow \Phi(\rho) = \Phi(\tau)$

$\&\ \forall\sigma.\forall\rho.\forall X.\ (\sigma,X) \stackrel{\cdot}{\rightarrow} \rho \Rightarrow \Phi(\sigma) = \Phi(\rho)$

$\&\ \neg\exists\sigma_0,\sigma_1,\ldots.\ \exists X.\forall i.\ (\sigma_i,X) \stackrel{\cdot}{\rightarrow} \sigma_{i+1}$

D2" is a postulate which satisfies the conditions of 8.24, so as before we can deduce the existence of a universal D"-space (P,Q-space which satisfies D1 and D2"). Over this space U it is possible to implement each of our operators correctly with the limitations described below:

(i) Non-constructive recursions are simply not defined when they give rise to divergence. This is the main weakness of this type of space.

(ii) Hiding is not defined where it would give rise to non-determinism.

(iii) The operators ";" and "$\Box$", while they can be fully defined, are very inefficient unless the rules set out in chapter one are followed, because backtracking is required if this is not done.

All one does when one tries to implement the deterministic model is to confirm the prejudices we developed in chapter one. It is now time to examine the non-deterministic model.

The non-deterministic model: first attempt

Recall that we interpret the value N(c) in the non-deterministic model M of a process "c" as being the set of sequences of external actions possible for "c" paired with the sets of symbols which "c" can refuse after accepting them. What is basically at issue here is the notion of "refusal"; this is closely linked with the observability of internal actions.

Let us first examine the implications of an assumption that an experimenter cannot observe what is going on inside any process. What must his criterion for deducing refusal be? Such an experimenter cannot tell the difference between a process which is deadlocked and one which is engaged in internal communication (whether or not this internal activity will eventually cease). There is no period after which he can deduce that any non-empty set he is offering is refused (i.e. no element of it has been or will be accepted). This is because there may, at any finite time, still be activity going on internally which will later result in acceptance. If we let c_n be a process which can (and must) perform n internal actions before becoming able to perform the external action "a", then any finite deduction of refusal by our experimenter would be incorrect (if he were offering the set {a} to one of the processes c_n) for sufficiently large n. Thus an experimenter can only deduce refusal when it is too late: when a set has been offered for an infinite time without response.

Bearing in mind that we wish to construct a map to the non-deterministic model based on the observed behaviour of processes, our next step must be to see how we can extract the observable parts of behaviour from the "full" behaviours of a process. From the point of view of an experimenter who cannot observe any internal behaviour any experiment will consist of finite applications of sets, either with or without observable response from the process, and possibly a (final) infinite application of a set without visible response. A very plausible procedure for the translation of our existing behaviours to "observations" is the following:

(i) Delete all the state components of the triples (i.e. project into H).

(ii) Replace all "·"s and non-final "*"s by "-".

(iii) Replace any final infinite sequence of the form $\langle(X_1,-)(X_2,-)\ldots\rangle$ by $(Z,*)$, where $Z = \tilde{\bigcup}_{i=1}(\tilde{\bigcap}_{j=i} X_j)$.

(iv) Delete any term of the form $(X,-)$ which is followed by one of the form $(X,-)$ or (X,a) (same X).

To illustrate this procedure let us apply it to the behaviour $\langle(\sigma_o,X_o,\cdot)(\sigma_1,X_o,*)(\sigma_1,X_1,a)(\sigma_2,X_2,\cdot)(\sigma_3,X_3,-)\ldots\rangle$ where all subsequent terms (infinitely many of them) have one of the forms $(\sigma_3,X_i,-)$ and $(\sigma_3,X_i,-)$.

The first and second steps translate this to $\langle(X_o,-)(X_o,-)(X_1,a)(X_2,-)(X_3,-)\ldots\rangle$, all subsequent terms having the form $(X_i,-)$. These steps relect the facts that the experimenter cannot see the structure of the state, and that he cannot detect internal actions or long intervals without them.

The third step translates this to $\langle(X_o,-)(X_o,-)(X_1,a)(Z,*)\rangle$ where Z is the liminf of all the X_is occurring in the final sequence. This step says that if the experimenter applies an infinite sequence of sets without response he can infer the refusal of all symbols applied continuously for an infinite period.

The fourth and final step reduces this to $\langle(X_o,-)(X_1,a)(Z,*)\rangle$. This step has the effect of collapsing contiguous applications of the same set into one application.

One point in the above procedure which seems a little suspect is the retention of final "*"s, since we interpret these (usually) as being infinite or sufficiently long finite waits without response. We cannot be sure that "*" represents an infinite wait (though we can be sure that it does not if it is not final). This complaint is rather academic however, since postulate Q8 ensures that final sets of "observations" of processes are the same whether or not we translate such "*"s as "-".

Let us define (for an element c of P,Q-space C) the set Obs(c) which results from the translation of each of the elements of

B(c). There are several results which one can prove of Obs(c) from our postulates. In the following we denote the sequences of pairs which constitute observations by $\underline{s}$, $\underline{t}$, ...

8.26 Lemma

If C is a P,Q-space and $c \in C$ then we have:

a) $\langle\rangle \in Obs(c)$

b) $\underline{s}.\underline{t} \Rightarrow \underline{s}'\langle(\emptyset,*)\rangle \in Obs(c)$ & $\underline{s}''\langle(X,-)\rangle \in Obs(c)$
where $\underline{s}'$ is $\underline{s}$ stripped of any final "(Y,*)" or "(Y,-)"s and $\underline{s}''$ is $\underline{s}$ stripped of any final (X,-)

c) $\underline{s}\langle(X,a)\rangle\underline{t} \in Obs(c) \Leftrightarrow a \in X$ & $\underline{s}\langle(\{a\},a)\rangle\underline{t} \in Obs(c)$
provided $\underline{s}$ has neither of the forms $\underline{s}''\langle(\{a\},-)\rangle$ and $\underline{s}''\langle(X,-)\rangle$

d) $\underline{s}\langle(X,\delta)\rangle\underline{t} \in Obs(c)$ & $X \neq Y$ & $(\neg\exists \underline{r}.\ \underline{s} = \underline{r}\langle(Y,-)\rangle)$ & $\delta \neq *$
$\Leftrightarrow \underline{s}\langle(Y,-)(X,\delta)\rangle\underline{t} \in Obs(c)$

e) $\underline{s}\langle(X,-)(X,\delta)\rangle\underline{t} \notin Obs(c)$; $\underline{s}\langle(X,-)(Y,*)\rangle \notin Obs(c)$

f) $\underline{s}\langle(X,*)\rangle\underline{t} \in Obs(c) \Rightarrow \underline{t} = \langle\rangle$

g) $\underline{s}\langle(X,*)\rangle \in Obs(c)$ & $Y \subseteq X \Rightarrow \underline{s}\langle(Y,*)\rangle \in Obs(c)$

h) $\underline{s}\langle(X,*)\rangle$ $Obs(c)$ & $(\forall a \in Y.\ \underline{s}\langle(\{a\},a)(\emptyset,*)\rangle \notin Obs(c))$
$\Rightarrow \underline{s}\langle(X \cup Y,*)\rangle \in Obs(c)$

The above tells us that we can effectively deduce nothing from the components of observations which have the form (X,-), so we might as well ignore them. In fact it tells us that the set of finite observations (i.e. observations with only finitely many components) can be deduced from a knowledge of which observations of the form $\langle(\{a\},a)\ldots(\{d\},d)(X,*)\rangle$ are in the set Obs(c) (i.e. finite sequences of single symbols offered and accepted, followed by a set infinitely refused). This clearly is closely related to the non-deterministic model. Define a map "ζ" from observations to $(\Sigma^* \times \wp(\Sigma))$ by $\zeta(\underline{s}) = (\langle\rangle,\emptyset)$ if $\underline{s}$ does not have the above "canonical" form; $\zeta(\underline{s}) = (\langle a\ldots d\rangle, X)$ if $\underline{s} = \langle(\{a\},a)..(\{d\},d)(X,*)\rangle$. Let us call the function from H to observations represented by steps (ii) - (iv) of the earlier translation procedure by the name "η". Define $\theta = \zeta \circ \eta$. Define a map "Ψ" from C (a P,Q-space) to $\wp(\Sigma^* \times \wp(\Sigma))$ by $\Psi(c) = \{\hat{\theta}(\underline{a}) \mid \underline{a} \in B(c)\}$. From the above discussion we can deduce several things about this function.

Firstly $\Psi(c)$ is almost an element of the non-deterministic model M. $\Psi(c)$ is non-empty, has a prefix-closed domain, and

satisfies the two conditions $(s,X) \in \Psi(c)$ & $Y \subseteq X \Rightarrow (s,Y) \in \Psi(c)$ and $(s,X) \in \Psi(c)$ & $(\forall a \in Y.(s\langle a\rangle,\emptyset) \notin \Psi(c)) \Rightarrow (s,X \cup Y) \in \Psi(c)$.

Secondly we can deduce from $\Psi(c)$ exactly what the finite elements of Obs(c) are. Furthermore, if c and d are two processes such that $\Psi(c) \subseteq \Psi(d)$, then every finite element of Obs(c) is also an element of Obs(d). This means that every predicate of C which depends only on the observed response of a process to finite sequences of sets can be exactly determined from the process' image under Ψ. Any predicate of the form "each finite observation is correct" can be translated to a predicate "χ" of H for which $\hat{\chi}^{+}(\Psi(c)) \Leftrightarrow \hat{\chi}(c)$. If we restrict ourselves to predicates of this form (of which more later) we can regard the image of Ψ as a class 1 model for C.

There is no necessity that $\Psi(c)$ should satisfy the directed closure condition which we imposed on the non-deterministic model. To see this we simply have to consider the following case, where it is assumed that $N \subseteq \Sigma$ (natural numbers).

Define a P,Q-space C to contain the elements σ (which has no transitions), ρ_n (for each $n \in N$) which has exactly the transitions $\rho_n \xrightarrow{m} \sigma$ $(m \geq n)$, and finally τ which has exactly the transitions $\tau \dashrightarrow \rho_n$ $(n \in N)$. A little thought reveals that $(\langle\rangle,X) \in \Psi(\tau)$ for each finite $X \subset N$, but that $(\langle\rangle,N) \notin \Psi(\tau)$.

There are essentially two ways of putting this right: either one closes up under the rule (i.e. modifying the function Ψ to include all pairs (s,X) implied by directed closure) or one restricts consideration to spaces C which satisfy it naturally. The simpler of these two options is the second, and it is this one which we shall follow here. There are two alternative conditions we can adopt to ensure directed closure: either Σ must be finite or the P,Q-space we study must be finite branching. In the first case directed closure is trivial; in the second case it follows from Konig's lemma.

Let us consider now the expressive power of the type of predicates described above. A predicate of the form "every finite observation is correct" is clearly the same as one which says that "every incorrect finite observation is impossible". Thus any predicate of behaviour with the property

that all infringements of it are both observable and detectable after finitely many external actions, is of the correct form for reliable and monotonic determination from $\Psi(c)$. As an example consider the implications of the predicate Buff (first introduced in chapter five) when true of $\Psi(c)$. (Note that Buff is a monotonic predicate.)

Buff($\Psi(c)$) implies several facts about Obs(c), which can informally be written as follows.

(i) "c" is a partially correct buffer, in that its output is at all times a prefix of its input.

(ii) When "c"s output is the same as its input (is "empty") it will not infinitely resist communicating with any experimenter who persists in offering it some set of input symbols.

(iii) When "c" has output less than it has input it will not infinitely resist communicating with an experimenter who persists in offering it (at least) the symbol which it next ought to output.

Note one fact which is illustrated by this example, namely that our insistence that certain infinite (in time) observations be absent implies the presence of certain finite (in time) observations (with certainty, rather than possibility, of occurrence). It is because of this influence on the set of observations which we can reliably expect to occur in finite time that we need to consider the possible infinite refusals. The other type of infinite observation, namely cases where infinite sequences of external actions occur, has no such influence.

We have established that the non-deterministic model M is reasonably thought of as a class 1 model for U, the universal finite branching P,Q-space (relative to the function Ψ and whichever alphabet we care to use). The first difference which one notes between this case and our study of the deterministic model is the fact that M, when regarded as a P,Q-space in the natural way, is not finite branching and so is not naturally "modelled by itself". We would expect to think of M as a P,Q-space by the law $N_1 \xrightarrow{a} N_2$ if $N_2 \subseteq N_1 \underline{\text{after}} \langle a \rangle$. Even if Σ is finite this gives rise to infinite branching. If Σ is infinite the situation is

irredeemable since there are elements of M which cannot be the image under Ψ of any element of U. An example of such a process is given by $\{(\langle\rangle,X) \mid \forall i.\{2i,2i+1\}\cap X \neq \emptyset\}\cup\{(\langle i\rangle,X) \mid i \in N\}$ (once again we assume $N\subseteq\Sigma$). If Σ is finite it is possible to make M into a finite branching P,Q-space on which the map Ψ is the identity. One way of doing this is with the transitions

$$a \in N \Rightarrow N \xrightarrow{a} (N \text{ after } \langle a\rangle)$$

$$(\langle\rangle,X) \in N \;\&\; (\Sigma - X) \subsetneq N^{o} \Rightarrow N \dashrightarrow (\{(\langle\rangle,Y) \mid Y \subseteq X\}\cup\{(\langle a\rangle s,Y) \in N \mid a \notin X\}).$$

We can thus conclude that the map $\Psi:U \to M$ is onto if and only if Σ is finite.

It is now time to consider the question of implementation. We have a class 1 model so our requirements are not on the face of it so stringent as in the case of the deterministic, class 2, model. We know from long experience that each of our operators over the model M is monotonic, which is all that is required of them for the class 1 model theory to work. Recall our definitions: $e^* \in U'$ is a correct implementation of $e \in M'$ if for all "states" $\pi \in U^{\Theta}\times S$ we have (adopting the same notation as in our study of the deterministic model) $e(\Psi(\pi_1),\pi_2) \supseteq \Psi(e^*(\pi))$. If $\underline{op}:M'^{k} \to M'$ is a k-place operator over M then $\underline{op}^*:U'^{k} \to U'$ is said to implement op ($\underline{op}^*$ imp $\underline{op}$) if for all $e_1,\ldots,e_k \in M'$ and $e_1^*,..,e_k^* \in U'$ such that e_i^* implements e_i correctly for all i, we have that $\underline{op}^*(e_1^*,\ldots,e_k^*)$ is a correct implementation of $\underline{op}(e_1,\ldots,e_k)$. In the case of operators $\underline{op}$ and $\underline{op}^*$ which are just the natural extensions to M' and U' of operators over M and U (op and op*, say) it is easy to see that it is enough to prove that $op(\Psi(c_1),\ldots,\Psi(c_k)) \supseteq \Psi(op^*(c_1,\ldots,c_k))$ for all $c_1,..,c_k \in U$. (This remains true in a simple way even when $\underline{op}^*$ is the composition of more than one "extended" operators, since the extension of a composition is the same as the composition of extensions.) Thus in attempting to implement the majority of our operators over M' which are extensions of operators over M it is sufficient to consider the implementation of the M-operators by U-operators.

Notice the fact that infinite hiding is impossible to define properly over the model M, and also impossible to implement properly over U in any obvious way (because of the creation of infinite branching). This emphasizes the link between

these two systems, and goes at least part of the way towards explaining the difficulties which arise with infinite hiding over the non-deterministic model.

Some of the operators, notably a $\rightarrow$, ;, /X (finite X) are easy to define correctly. Indeed it is not too hard to show that the versions of these three operators introduced earlier (in the section on operators over universal spaces) are all well-defined over U and correct implementations of the corresponding operators over M. Also the operators over U' we defined to represent "x:T $\rightarrow$" and recursion are not hard to justify. An interesting example is the case of recursion. Firstly the existence of fixed points of "constructive" functions of U is easy to prove because of the comparatively simple structure of the space U. Suppose that e* is an element of U' defined using operators of the types described earlier which correctly implements some element e of M'. If B is any process variable (element of Θ) then for each "state" π the equation $\sigma = e^*(\pi[\sigma/B])$ has a unique solution which we will call σ. To show that the recursion operator over U' correctly implements that over M' it is sufficient to show that $\Psi(\sigma) \subseteq \bigsqcup_{n=0}^{\infty} F^n(CHAOS)$, where $F:M \rightarrow M$ is defined $F(A) = e(\pi'[A/B])$ $(\pi' = \Psi(\pi))$. To do this it is sufficient to show that $\Psi(\sigma) \subseteq F^n(CHAOS)$ for each n; we will do this by induction.

$\Psi(\sigma) \subseteq F^0(CHAOS) = CHAOS$ trivially.

Suppose $\Psi(\sigma) \subseteq F^n(CHAOS)$,

then $e(\pi'[\Psi(\sigma)/B]) \subseteq e(\pi'[F^n(CHAOS)/B]) = F^{n+1}(CHAOS)$

as e is monotonic; also $\pi'[\Psi(\sigma)/B] = \Psi(\pi[\sigma/B])$ so

$e(\pi'[\Psi(\sigma)/B]) \supseteq \Psi(e^*(\pi[\sigma/B]))$ as e* correctly implements e.

Putting these facts together we get $F^{n+1}(CHAOS) \supseteq \Psi(e^*(\pi[\sigma/B]))$, which is what we wanted to show since $\sigma = e^*(\pi[\sigma/B])$.

The analysis of mutual recursions is no more difficult.

Problems arise however when one tries to implement the two important operators "$\Box$" and "||". In each case the problem occurs because of the necessity of being able to cope with one diverging and one non-diverging operand. It is not hard to see from the non-destructive nature of our operators and the finite-branching nature of U, that each of these operators, when faced with an initially diverging operand,

must itself have the possibility of diverging before performing any external actions. This is because they must be able to cope with processes which execute n internal actions before doing any external action for each $n \in N$, and so can perform arbitrarily long initial sequences of internal actions; we can then deduce the existence of an infinite sequence by Konig's lemma.

With "$\Box$" this problem is avoided if we restrict the use of "$[]$" to the case where each side is guarded in some way (whether or not the guards are disjoint).

However with "||" the problems are more serious. There seems to be no way of implementing "||" correctly without appealing to some kind of "fairness". One would require that neither operand could perform infinitely many actions while there was some action continuously possible for the other. While this certainly does not seem an unreasonable assumption it is not possible to make such a stipulation in systems satisfying the postulate Q7. An example of the problem we face is given below.

Let σ be the element of U with the single transition $\sigma \stackrel{\cdot}{\to} \sigma$ and let ρ be the element of U with the single transition $\rho \stackrel{a}{\to} \tau$, where τ has no transitions ($a \in \Sigma$). The values assigned by Ψ to σ and ρ are respectively <u>abort</u> and a → <u>abort</u>. Thus $(\Psi(\sigma) \underset{\{b\}}{\|}_{\{a\}} \Psi(\rho)) =$ a → <u>abort</u>. Thus whenever op* is a correct implementation of "$\|_{\{a\}}^{\{b\}}$" the process op*(σ, ρ) cannot initially diverge (as $\Psi(\upsilon)$ contains $(\langle\rangle, \Sigma)$ whenever υ can initially diverge).

The introduction of fairness would require the alteration of Q7. This would rather complicate matters. Firstly our work concerning morphisms and universal spaces would no longer be valid because 8.10 would no longer hold. Secondly allowing fairness would mean that the image $\Psi(\sigma)$ of a finite-branching process was not necessarily directed-closed (because Konig's lemma would no longer be applicable). While it is likely that these problems could be overcome to some extent and some sort of consistent theory produced we have not got space here to investigate this topic further. In any case it is perhaps better not to assume fairness unless we have to, and we will see shortly that we can do

without it. It might also be remembered that the map Ψ identifies divergence with deadlock, something which does not seem consistent with the philosophy expressed in the introductory booklet (e). This fact is brought out further by the observation that, with respect to the map Ψ, the hiding operator "/X" which we earlier defined over U is a correct implementation of the (non-continuous) alternative hiding operator "\X" defined over M by

$$A \backslash X = \{(s/X, Y) \mid (s, X \cup Y) \in A\} \cup \{(s/X, Y) \mid \{s' \mid s'/X = s/X\} \text{ is infinite}\} .$$

All this seems to imply that the map "Ψ" is not quite the one we want. We will thus give up this attempt and try again.

The non-deterministic model: second attempt

In the last section we made the assumption that our modelling function was based on the observations of an experimenter who cannot detect anything about what goes on inside processes. We have previously mooted the possibility that an experimenter might be able to detect the presence of internal activity by means of a light on the side of the machine or suchlike. The chief consequence of assuming this would be the finite detectability of deadlock by the experimenter. If a set of symbols is offered to a process while it is inactive (the light is out) for a sufficiently long time the experimenter may (correctly) deduce that no further action (internal or external) can occur while he persists in offering the same set (or a subset of it).

There is thus often no need to wait infinitely to detect refusal of a set. Indeed the need to wait infinitely would be rather unfortunate, implying both an infinite consumption of energy by the process and infinite patience on the part of the experimenter. It is important to discriminate between the notions of finite refusal (brought about by the process coming into some "stable" state) and infinite refusal (brought about by divergence, an infinite sequence of internal actions). Because our experimenter now has the ability to detect refusal finitely, and because divergence is inherently undesirable, we must expect that he will desire that

all refusals will be finite (i.e. that processes are free from divergence).

Let us assume that our experimenter is chiefly interested in the behaviour of processes over finite intervals. We will thus examine the behaviours which can result from the application of a finite sequence of sets to a process. There are several things which the experimenter might see when he applies a set:

finite inconclusive wait	(denoted by -)
finitely observed refusal	(*)
communication of some $a \in \Sigma$	(a)
divergence (infinite wait without response)	(?)

We will assume that the experimenter is not confident enough to record any other details about internal activity than that which is implied in the above.

We will assume that he bases all correctness conditions upon a process' observed reactions to such experiments. Note that any observation of this type which does not include divergence is completed in a finite time, and that divergence can only occur at the end of an observation. This is of course a very good reason for defining divergent observations to be incorrect.

The following is a translation procedure designed to extract from the behaviour set of a process those behaviours which can result from the application of a finite sequence of sets, and then extract the observable features of these (from the point of view of the experimenter described above).

(i) Delete the state components of a behaviour (i.e. project into H).

(ii) If the resulting sequence has one of the following "inadmissible" types replace it by $\langle\rangle$.

a) Sequence with infinitely many external actions, "-"s and "*"s.

b) Any sequence with an infinite tail of (X,·)s in which the "X"s are not eventually constant.

(iii) Replace all "·"s by "-".

(iv) Replace any infinite tail of "(X,-)"s (with constant X) by (X,?).

(v) Delete all "$(X,-)$"s which are followed by some (X,δ) (same X).

The explanation of these steps is as follows:

(i) The internal state is not observable.

(ii) These types of sequences cannot result from behaviours which occur when the experimenter applies a finite sequence of sets. Sequences of type (a) are impossible because the end of each pair of any of the forms $(X,-)$, (X,a) or $(X,*)$ represents the end of an application of a set. This is not so for pairs of the form $(X,\cdot)$ because individual internal actions are not observable. The only infinite sequences which are left after (a) are those with an infinite tail of the form $\langle(X_1,\cdot)(X_2,\cdot)\ldots\rangle$; clause (b) eliminates all of these which cannot occur when a finite sequence of sets is applied.

(iii) Individual internal actions are not observable.

(iv) Any infinite tail of the form $\langle(X,-)(X,-)\ldots\rangle$ must have resulted from an infinite sequence of internal actions.

(v) Any $(X,-)$ which is followed by another application of the same set can be ignored.

Define ξ to be the function of behaviours implied by steps (ii) - (v) of the above procedure; define Obs'(c) to be $\{\hat{\xi}(\underline{a}) \mid \underline{a} \in B(c)\}$ for any process c. Obs'(c) is the set of observations which our experimenter can make of c. The following are all easy consequences of our postulates.

8.27 Lemma

If C is a P,Q-space and $c \in C$ then

a) $\langle\rangle \in$ Obs'(c)

b) $\underline{s}.\underline{t} \in$ Obs'(c) $\Rightarrow$ ($\underline{s}'(\emptyset,*) \in$ Obs'(c) $\vee$ $\underline{s}'(\emptyset,?) \in$ Obs'(c)) & $\underline{s}''(X,-) \in$ Obs'(c)

where $\underline{s}'$ & $\underline{s}''$ result from stripping $\underline{s}$ of any final $(\emptyset,-)$ and $(X,-)$ respectively after removing any final $(Y,?)$

c) $\underline{s}\langle(X,a)\rangle\underline{t} \in$ Obs'(c) $\Leftrightarrow$ $a \in X$ & $\underline{s}\langle(\{a\},a)\rangle\underline{t} \in$ Obs'(c) if $\underline{s}$ has neither of the forms $\underline{s}'\langle(X,-)\rangle$ and $\underline{s}'\langle(\{a\},-)\rangle$

d) $\underline{s}\langle(X,\delta)\rangle\underline{t} \in$ Obs'(c) & $X \neq Y$ & $\neg\exists\underline{r}.\ \underline{s} = \underline{r}\langle(Y,-)\rangle$ $\Leftrightarrow$ $\underline{s}\langle(Y,-)(X,\delta)\rangle\underline{t} \in$ Obs'(c)

e) $\underline{s}\langle(X,?)\rangle\underline{t} \in$ Obs'(c) $\Rightarrow \underline{t} = \langle\rangle$ (f) $\underline{s}\langle(X,*)(Y,?)\rangle \notin$ Obs'(c)

g) $\underline{s}\langle(X,*)(Y,a)\rangle\underline{t}\in Obs'(c) \Rightarrow a\notin X$

h) $\underline{s}\langle(X,*)\rangle\underline{t}\in Obs'(c)\ \&\ Y\subseteq X \Rightarrow \underline{s}'\langle(Y,*)\rangle\underline{t}\in Obs'(c)$
where $\underline{s}'=\underline{s}$ unless $\underline{s}=\underline{s}''\langle(Y,-)\rangle$ in which case $\underline{s}'=\underline{s}''$

i) $\underline{s}\langle(X,*)\rangle\underline{t}\in Obs'(c)\ \&\ (\forall a\in Y.s\langle(X,*)(\{a\},a)(\emptyset,-)\rangle\notin Obs'(c))$
$\Rightarrow \underline{s}'\langle(X\cup Y,*)\rangle\underline{t}\in Obs'(c)$
where $\underline{s}'=\underline{s}$ unless $\underline{s}=\underline{s}''\langle(X\cup Y,-)\rangle$ in which case $\underline{s}=\underline{s}''$

j) $\underline{s}\langle(X,*)\rangle\underline{t}\in Obs'(c) \Rightarrow \underline{s}'\underline{t}\in Obs'(c)$
where $\underline{s}'=\underline{s}$ unless there are some $\underline{s}''$, $\underline{t}'$, Y and δ such that $\underline{s}=\underline{s}''\langle(Y,-)\rangle$ and $\underline{t}=\langle(Y,\delta)\rangle\underline{t}'$ in which case $\underline{s}'=\underline{s}''$

k) $\underline{s}\langle(X,?)\rangle\in Obs'(c) \Rightarrow \underline{s}'(Y,?)\in Obs'(c)$
where $\underline{s}'=\underline{s}$ unless $\underline{s}=\underline{s}''\langle(Y,-)\rangle$ in which case $\underline{s}'=\underline{s}''$

l) $\underline{s}\langle(X,-)\rangle\ Obs'(c) \Rightarrow \underline{s}\langle(X,*)\rangle\in Obs'(c)\vee \underline{s}\langle(X,?)\rangle\in Obs'(c)$
$\exists a\in X.\ \underline{s}\langle(X,a)(\emptyset,-)\rangle\in Obs'(c)$

m) $\underline{s}\langle(X,*)(Y,*)\rangle\underline{t}\in Obs'(c) \Leftrightarrow \underline{s}\langle(X\cup Y,*)\rangle\underline{t}\in Obs'(c)$
provided $\underline{s}$ has neither of the forms $\underline{s}'(X,-)$ or $\underline{s}'(X\cup Y,-)$

Once again, by d, we can deduce nothing from the components of observations with the form (X,-) so we might as well ignore them (from the above the only purpose they seem to serve is to complicate matters). Once again it is possible to deduce the exact form of Obs'(c) from a knowledge of which of a class of "canonical" elements it contains. These are the ones which are of the form $\{(\{a\},a),(X,*),(\emptyset,?)\mid a\in\Sigma,X\subseteq\Sigma\}^*$ with $(\emptyset,?)$ only occuring at the end of sequences and components of the form $(X,*)$ being separated by at least one of the form $(\{a\},a)$.

Thus every predicate of processes of the form "each observation of behaviour is correct" can be re-written in the form "each canonical observation is correct".

Because of the various rules of 8.27, notably (b), (f), (g), (j), (k) and (m) a very expressive subset of the canonical observations are those of the two forms

$\langle(\{a\},a)\ldots(\{d\},d)(X,*)\rangle$ (+)

$\langle(\{a\},a)\ldots(\{d\},d)(\emptyset,?)\rangle$ (++).

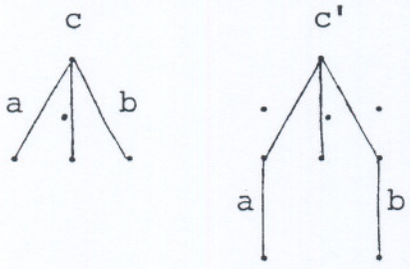
From a knowledge of which observations of these two types are present in Obs'(c) one cannot deduce the whole shape of Obs'(c) (see below) but one can answer such questions as

"Can c perform the string ⟨a...d⟩ and then finitely refuse the set X?"

"Can c perform the string ⟨a...d⟩ and then diverge when offered X?"

accurately. (For example, if $\langle\langle(\{a\},a)\ldots(\{d\},d)(X,*)\rangle \in Obs'(c)$ then there is no element of Obs'(c) in which the symbols a..d are accepted from any sets followed by the finite refusal of X, whether or not there are any "(Y,-)"s and/or "(Y,*)"s between these events.)

There do exist processes with identical sets of observations of the forms (+) and (++) but different sets Obs'. As an example of this consider the two processes illustrated below.



$(\Sigma = \{a,b\})$

The processes c and c' have the same observations of the forms (+) and (++), but $\langle(\{a\},*)(\{b\},b)(\emptyset,*)\rangle$ is an element of Obs'(c') without being one of Obs'(c).

The minor differences between processes with identical sets of (+) and (++) observations are usually unimportant from a correctness point of view, however. The use of observations of this simplified form has the advantages of simplicity and the production of a natural map into the non-deterministic model M.

Let us suppose that our experimenter, aware of the expressive power of observations of types (+) and (++), contents himself with checking to see which if them are possible. It is sufficient for him to check those of type (+), since any possible observation of type (++) will become apparent as he does this. To check the observation $\langle(\{a\},a)\ldots(\{d\},d)(X,*)\rangle$ (which we abbreviate as $(\langle a..d\rangle,X)$ he will apply the sets $\{a\},..,\{d\}$, X and $\emptyset$ in turn, with each set except the last waiting until he gets some definite response. There are essentially three possible outcomes to such an experiment.

(i) It may succeed: i.e. the process may accept a,..,d in turn and then refuse X finitely.

(ii) It may fail finitely: i.e. it may finitely refuse one of the sets $\{a\},..,\{d\}$ or it may accept some element of X.

(iii) It may diverge: i.e. the process may not give any definite response at some stage where one is required.

Define the three sets E_1^c, E_2^c and E_3^c to be the sets of possible passes, failures and divergences amongst the experiments he might carry out on the process c. (We will assume that the elements of these sets are written in the abbreviated form, so that E_1^c, E_2^c and E_3^c are all subsets of $\Sigma^* \times \wp(\Sigma)$.) We can deduce E_1^c, E_2^c and E_3^c from Obs'(c) as follows:

$$E_1^c = \{(\langle a..d\rangle, X) \mid \langle(\{a\},a)\ldots(\{d\},d)(X,*)\rangle \in Obs'(c)\}$$

$$E_2^c = \{(\langle a..d\rangle, X) \mid \langle(\{a\},*)\rangle \in Obs'(c) \vee \ldots \vee \langle(\{a\},a)..(\{d\},*)\rangle \in Obs'(c) \vee \exists b.\ \langle(\{a\},a)\ldots(\{d\},d)(X,b)(\emptyset,-)\rangle \in Obs'(c)\}$$

$$E_3^c = \{(\langle a..d\rangle, X) \mid \langle(\{a\},?)\rangle \in Obs'(c) \vee \ldots \vee \langle(\{a\},a)..(\{d\},?)\rangle \in Obs'(c) \vee \langle(\{a\},a)\ldots(\{d\},d)(X,?)\rangle \in Obs'(c)\}$$

(We might note at this point that the three sets each depend monotonically on the set Obs'(c).)

These sets satisfy some simple laws, which are easily provable from 8.27 and their definitions.

8.28 Lemma

(i) $(s,X) \in E_3^c \Rightarrow (st,Y) \in E_3^c$

(ii) $(s,X) \in E_1^c \ \&\ Y \subseteq X = (s,Y) \in E_1^c$

(iii) $(s,X) \in E_1^c \ \&\ (\forall a \in Y.(s\langle a\rangle,\emptyset) \notin E_1^c \cup E_3^c) \Rightarrow (s,X \cup Y) \in E_1^c$

(iv) $(st,X) \in E_1^c \cup E_3^c \Rightarrow (s,\emptyset) \in E_1^c \cup E_3^c$

(v) $E_1^c \cup E_2^c \cup E_3^c = \Sigma^* \times \wp(\Sigma)$

(vi) $(\langle\rangle,\emptyset) \in E_1^c \cup E_3^c$

The set $E_1^c \cup E_3^c$ represents the set of experiments which need not necessarily fail finitely, and as such is very important. If we desire that every observation of type (+) which occurs is correct in some sense then this can be checked by trying out all incorrect ones and expecting them to fail finitely. By the above this must occur if each element of $E_1^c \cup E_3^c$ is correct. Because of (i) above there is also a simple way to exclude undesirable divergence (observations of type (++))

from consideration of the set $E_1^c \cup E_3^c$. In particular the set Obs'(c) is completely free from divergence if $E_1^c \cup E_3^c$ satisfies the condition

$$\forall s.\forall X.(s,X) \in E_1^c \cup E_3^c \Rightarrow \exists t,Y.\ (st,Y) \notin E_1^c \cup E_3^c$$

since this implies that $E_3^c = \emptyset$. When E_3^c is empty every finite experiment (whether or not of the form used to determine E_1^c, E_2^c and E_3^c) must terminate finitely. When E_3^c is empty it is not hard to see that E_2^c depends monotonically on E_1^c (i.e. the more experiments there are which can pass, the more there are which can fail).

Thus $E_1^c \cup E_3^c$ is a very useful set. It also satisfies all the laws of the non-deterministic model except directed closure. When the P,Q-space in question is finite-branching we once again have directed closure. We will therefore once again assume that the space we are seeking to model is U, the universal finite branching P,Q-space.

We are, in $E_1^c \cup E_3^c$, provided with a natural and expressive map from U to the non-deterministic model M. We have not however defined it in the usual fashion (a function $\hat{\theta}$ of behaviours). It can in fact be written in a correct form, but we need to invoke our right to use a relation rather than a function. We already have the function $\hat{\zeta}$ for producing Obs'(c) from B(c). Define a relation μ on the sequences making up Obs'(c) as follows.

$$\begin{aligned}
\mu(\underline{s}) &= \emptyset \quad \text{if } \underline{s} \text{ has neither of the forms (+) and (++)} \\
&= \{(\langle a..d\rangle, X)\} \quad \text{if } \underline{s} = \langle(\{a\},a)..(\{d\},d)(X,*)\rangle \\
&= \{(\langle a..d\rangle t, X) \mid t \in \Sigma^* \ \&\ X \subseteq \Sigma\} \text{ if} \\
&\qquad \underline{s} = \langle(\{a\},a)..(\{d\},d)(\emptyset,?)\rangle
\end{aligned}$$

If we now let $\rho = \mu \circ \hat{\zeta}$ it is clear that $E_1^c \cup E_3^c = \cup\{\rho(\underline{a}) \mid \underline{a} \in B(c)\}$.

Let us define $X(c) = E_1^c \cup E_3^c$. Because of the expressive power of the set $E_1^c \cup E_3^c$ it is fair to regard M as a class 1 model for U relative to the map X.

This map seems far more satisfactory than Ψ from an aesthetic point of view: the discrimination between divergence and deadlock appears to correspond far better to our earlier expectations. Despite this the problems of implementing our standard operators are worse rather than better.

"a →", "x:T →", "a.x:T →", "x →" , "a.x →" and "or" present no difficulty, and neither do hiding and recursion. (Note that hiding and recursion are the two operators most closely associated with divergence.) All the other operators seem to give rise to unsurmountable problems. This is because of the ways in which they deal with the representations of diverging processes. The value given by X to any process which can diverge without communicating externally is CHAOS. If any of the operators "a.", "$\Box$" and "$\|$" is presented with such a process in any argument or if ";" is presented with one in its first argument then, by the same arguments which we used in the last section for "$\|$", the result must also be able to diverge without communicating externally (and so have value CHAOS assigned to it by X). This is inconsistent with the following observations:

$a.CHAOS \neq CHAOS$

$CHAOS;\underline{abort} \neq CHAOS$

$CHAOS \,\Box\, (a \rightarrow \underline{skip}) \neq CHAOS$

$(CHAOS_X \|_Y \underline{abort}) \neq CHAOS$ unless $X = \Sigma$ and $Y = \emptyset$.

(This time there is no hope of mending "$\|$" by an assumption of fairness.)

The basic problem here is that, though in the initial construction of our model we were careful to identify the "bad" processes we created with CHAOS (witness the correctness of hiding and recursion), we did not follow through our arguments to see how these "bad" processes would behave when operated upon. In short it seems to be the operators which we defined over M which are at fault here rather than the modelling function: they do not appear to be reasonable (in the sense defined earlier). All these failings can be remedied by adjusting the definitions of the operators: making them "strict" in some sense. The following are definitions of more acceptable operators "$\Box'$", ";'", "a.'" and "$\|'$".

$A \,\Box'\, B = A \,\Box\, B$ if $A \neq CHAOS$ and $B \neq CHAOS$

$= CHAOS$ otherwise

$A;'B = A;B \cup \{(s,X) \mid A \underline{\text{ after }} s = CHAOS\}$

$a.'B = a.B \cup \{((a.s)t,X) \mid B \underline{\text{ after }} s = CHAOS\}$

$(A_X \|'_Y B) = (A_X \|_Y B) \cup \{(st,X) \mid s \in (X \cup Y)^* \;\&\; s\restriction X \in dom(A) \;\&$

$s\restriction Y \in dom(B) \;\&\; CHAOS \in \{A \underline{\text{ after }} s\restriction X,\ B \underline{\text{ after }} s\restriction Y\}\}$

Each of these operators seems to be implementable with respect to X. There is however a price to pay. Some of the theory which we developed for the old operators no longer holds. A notable example of this is that the operator $||'$ is not associative: suppose X,Y is a non-trivial partition of Σ (i.e. $X \neq \emptyset$ & $Y \neq \emptyset$).

Let $A = \{(s,S) \mid s \in X^*\}$

$B = \{(s,S) \mid s \in Y^*\}$

Observe that $((A_X \|'_Y B)_\Sigma \|'_\Sigma \underline{\text{abort}}) = \text{CHAOS}$

$(A_X \|'_\Sigma (B_Y \|'_\Sigma \underline{\text{abort}})) = \underline{\text{abort}}$

It is ironic that lemma 5.35 now holds in general (i.e. we no longer need to assume the absence of infinite internal chatter). These two lemmas (5.35 and associativity of $||$) were both critical in proving the associativity of "$\gg$", the non-universality of which was one of the more paradoxical properties of our old operators. 5.35 holds in our new system generally because the parallel operator is no longer allowed to "hide" divergence from the environment.

Most of the troubles with the new operators seem to arise from the special way in which CHAOS is treated, the identification of a diverging process with one which can do anything but always terminates finitely. In the next section we will see how this problem can be removed through an adjustment to the model.

The main part of our earlier theory which is hit by the new operators is recursion through the parallel operator. For a discussion of how this is affected see the next section.

An improved model

This section is an extension of the last, so we will use the same notation.

The obvious way round the problems which arise from the confusion over CHAOS is to separate the notions of diverging and passing experiments more. One way of doing this would be to adopt the pair (E_1^C, E_3^C) as our representation of a process. This would have the advantage of being much more expressive; it has the disadvantage that the underlying model is not nearly so elegant or mathematically versatile as the old model M.

We can arrive at a satisfactory compromise in the following manner. Firstly we re-state our view that divergence of an experiment is worse than anything else. This time we will therefore keep a separate record of all experiments which may diverge. However we note again that when an experiment diverges there is no way in which an experimenter can finitely detect that it will not pass. We therefore adopt as our new representation of a process

$$\Omega(c) = E_1^c \cup E_3^c \cup \{(s,?) \mid (s,\emptyset) \in E_3^c\}.$$

We take as our new model the subset Q of $\wp(\Sigma^* \times (\wp(\Sigma) \cup \{?\}))$, defined to be those elements N which satisfy the following conditions. (We use the same notation as before, except that α will now conventionally represent an element of $\wp(\Sigma) \cup \{?\}$.)

(i) dom(N) is non-empty and prefix closed

(ii) $(s,?) \in N \Rightarrow (st,X) \in N$ & $(st,?) \in N$ for all t,X

(iii) $(s,X) \in N$ & $Y \subseteq X \Rightarrow (s,Y) \in N$

(iv) $(s,X) \in N$ & $(\forall a \in Y.(s\langle a\rangle,\emptyset) \notin N) \Rightarrow (s,X \cup Y) \in N$

(v) $\{X \mid (s,X) \in N\}$ is directed closed

Ω is easily seen to be a well-defined map from U to Q, and can easily be shown to be generated by a relation on behaviours in a very similar way to X.

The new model Q seems to possess all of the useful properties of M and a few more besides. There is a natural order on Q in exactly the same way as on M ($A \sqsubseteq B$ if $B \subseteq A$) and Q is a complete semilattice with respect to "$\sqsubseteq$". Q has minimum element $\Sigma^* \times (\wp(\Sigma) \cup \{?\})$ which we will call CHAOS to distinguish it from CHAOS ($= \Sigma^* \times \wp(\Sigma)$) which is not minimal in Q, though it is the minimal element free from divergence. Q has the same maximum elements as M. We should note that if c is an element of U free from divergence then $\Omega(c) = X(c) = \Psi(c)$. The map Ω makes more distinctions than either of the others, guaranteeing that if c is a process free from divergence and c' is one which can diverge then $\Omega(c) \neq \Omega(c')$.

The operators we define over this model should correspond to our original operators over M for processes free from divergence and take heed of our observations in the last

section when their arguments can diverge. The following is a list of the operators we adopt over Q, together with a few remarks on how one might expect them to be implemented.

$$a \rightarrow A = \{(\langle\rangle, X) \mid a \notin X\} \cup \{(\langle a\rangle s, \alpha) \mid (s,\alpha) \in A\}$$

(We use the same scheme as before, switching "A" on after "a".)

$$A;B = \{(s,X) \mid s \in (\Sigma^-)^* \ \&\ \surd \notin X \ \&\ (s,X) \in A\}$$
$$\cup \{(st,\alpha) \mid s \in (\Sigma^-)^* \ \&\ (s\langle\surd\rangle, \emptyset) \in A \ \&\ (t,\alpha) \in B\}$$
$$\cup \{(s,\alpha) \mid (s,?) \in A\}$$

(We use the scheme defined in the section on operators. A is switched on initially, and when A communicates a hidden "$\surd$" it is switched off and B is switched on.)

$$A \,\square\, B = \{(\langle\rangle, X \cap Y) \mid (\langle\rangle, X) \in A \ \&\ (\langle\rangle, Y) \in B\}$$
$$\cup \{(s,\alpha) \mid (s,\alpha) \in A \cup B \ \&\ s \neq \langle\rangle\}$$
$$\cup \{(s,\alpha) \mid (\langle\rangle, ?) \in A \cup B\}$$

(Initially both are switched on and are allowed to perform any action. As soon as one performs an internal action the other is switched off.)

$$A \underline{\text{or}}\ B = A \cup B$$

(This is trivial to implement.)

$$(A \,{}_X\|_Y\, B) = \{(s, (X \cap V) \cup (Y \cap W) \cup T) \mid s \in (X \cup Y)^* \ \&\ (s \lceil X, V) \in A \ \&$$
$$(s \lceil Y, W) \in B \ \&\ T \cap (X \cup Y) = \emptyset\}$$
$$\cup \{(st,\alpha) \mid s \in (X \cup Y)^* \ \&\ (s \lceil X, \emptyset) \in A \ \&\ (s \lceil Y, \emptyset) \in B \ \&$$
$$(s \lceil X, ?) \in A \vee (s \lceil Y, ?) \in B)\}$$

(Both processes are always switched on and their external communications co-ordinated in the obvious way.)

$$a.A = \{(a.s, a.X \cup Y) \mid (s,X) \in A \ \&\ Y \cap a.\Sigma = \emptyset\}$$
$$\cup \{((a.s)t, \alpha) \mid (s,?) \in A\}$$

$$A/X = \{(s/X, Y) \mid (s, X \cup Y) \in A\} \cup \{((s/X)t, \alpha) \mid (s,?) \in A\}$$
$$\cup \{((s/X)t, \alpha) \mid \{s' \in dom(A) \mid s'/X = s/X\} \text{ is infinite}\}$$

(These operators are implemented by operating on the external communications of "A" in the obvious ways.)

The various operators over Q' which we need (x:T →, recursion etc.) are defined in the obvious ways. Each of the above is a monotonic and continuous function of its operands.

In the above we thus seem to have definitions of our operators over Q which are implementable and which do not suffer from the artificial strictness we needed with χ. Because we do not need this strictness we recover all the pleasant properties of old operators with the addition of the universal truth of (h) below (Lemma 5.35).

8.29 Theorem

The operators we have defined over Q satisfy the following.

a) $A \,[]\, A = A$

b) $A \,[]\, B = B \,[]\, A$

c) $(A \,[]\, B) \,[]\, C = A \,[]\, (B \,[]\, C)$

d) $(A;B);C = A;(B;C)$

e) $(a \rightarrow A);B = a \rightarrow (A;B)$

f) $(A \,[]\, B);C = (A;C) \,[]\, (B;C)$ if $\surd \notin A^{o} \cup B^{o}$

g) $((A_X \|_Y B)_{X \cup Y} \|_Z C) = (A_X \|_{Y \cup Z} (B_Y \|_Z C))$

h) $(A/X_Y \|_Z C) = (A_{X \cup Y} \|_Z C)/X$ if $X \cap Z = \emptyset$

i) each of "a →", ";", "[]", "||", "a." and "/X" distributes over "or"

etc., etc.

Because we have both (g) and (h) we now have that "»" (if defined in the obvious way with a suitably defined "strip" operator) is in general associative on processes whose domains are contained in $(?T \cup !T)^*$.

Most of the theory of chapter five appears to go through practically unaltered. Operators remain non-destructive (in the obviously defined sense) in the same circumstances as before, and "a →", "x:T →" etc. are constructive. The parallel operator is never constructive over Q except in trivial circumstances; we do not use any constructive properties of "»" in chapter 5 and its non-destructiveness remains over Q in the circumstances of 5.30.

Circumstances are different in chapter six, however. Here we extensively use constructiveness properties of (A||a::B), an operator which is defined using "||". If this operator were defined in the same way over Q then it could never be constructive in its second argument when A was not CHAOS. This would be a very serious problem. It can be avoided by defining the operator seperately (i.e. without using

the parallel operator we defined over Q). To do this we must look at what this operator represents. It is fundamentally different from the ordinary parallel operator in that it that it implies the dominance of one operand over the other. Let us imagine that the dominant operand is in effect the "black box" of our earlier discussion with the power to switch the other one on or off. The "master" might only switch on the "slave" when it had any need of it (i.e. when it had the ability to communicate with it). There is thus reason to believe that the operator

$$
\begin{aligned}
(A \,||\, a::B) = \{&(s/a.\Gamma, X) \mid \exists Y,Z.(s,Y) \in A \;\&\; (Y \cup a.(\mathrm{swap?!}(Z)) = X \cup a.\Gamma \\
&\;\&\; (\mathrm{swap?!}(\mathrm{stripa}(s\restriction a.\Gamma)), Z) \in B\} \\
\cup \{&((s/a.\Gamma)\,t, \alpha) \mid (s,?) \in A \;\&\; (\mathrm{swap?!}(\mathrm{stripa}(s\restriction a.\Gamma)), \emptyset) \in B\} \\
\cup \{&((s/a.\Gamma)\,t, \alpha) \mid (s,\emptyset) \in A \;\&\; (\mathrm{swap?!}(\mathrm{stripa}(s\restriction a.\Gamma)), \emptyset) \in B \\
&\;\&\; \exists s' \leqslant s, \exists b \in a.\Gamma, s'\langle b\rangle \in \mathrm{dom}(A)\} \\
\cup \{&((s/a.\Gamma)\,t, \alpha) \mid \{s' \in \mathrm{dom}(A) \mid s/a.\Gamma = s'/a.\Gamma \\
&\;\&\; \mathrm{swap?!}(\mathrm{stripa}(s\restriction a.\Gamma)) \in \mathrm{dom}(B)\} \text{ is infinite}\}
\end{aligned}
$$

is implementable (it seems likely that an implementation might resolve some of the non-determinism inherent in the above). This operator can be shown to satisfy the relation $a \neq b \Rightarrow ((A \,||\, a::B) \,||\, b::C) = ((A \,||\, b::C) \,||\, a::B)$ (the failure of this over M was a consequence of the untruth of 5.35 when infinite internal chatter was present).

One pleasing aspect of the above definition is that it resolves the problems we had in chapter six with "network chatter" because the function $F(B) = (A \,||\, a::B)$ is no longer necessarily constructive if A satisfies the condition C_0^a (though it is with C_1^a). To see this consider the first example we quoted there, namely the process defined

$A \Leftarrow (X \,||\, a::A)$

$X \Leftarrow ?x \rightarrow a!x \rightarrow \underline{\text{abort}}$

The progress of the two sequences of approximations (over M and Q) is as follows.

	over M	over Q
$F^0(\perp)$	CHAOS	CHAOS
$F^1(\perp)$	?x → abort	?x → CHAOS
$F^2(\perp)$	?x → abort	?x → CHAOS
. . . . .	. . . . .	. . . . .

The value assigned to this process in Q is far more satisfactory.

The buffer and stack examples of chapter six should go through virtually unaltered (because their recursions are both C_1^a). The sort examples will need more care. We will need to apply the analysis previously used to prove them free of network chatter to prove them well-defined and correct (there seems little doubt that this could be done with a little care).

It is to be expected that all of the theory of chapter seven will apply equally to the new model, with the rider that we should no longer have to make assumptions on freedom from infinite internal chatter for "↔" to be associative, etc. The various conditions we develop here and in earlier chapters for the avoidance of infinite chatter are of course still of considerable use, as they will now imply freedom from divergence (when divergence-free processes are combined).

Thus the revised model Q seems to have considerable advantages over the old one, as it appears to remove several of the less satisfactory features of that model. There is of course much further work to be done in formally transferring the results of M to Q, and a final verdict must await that together with more rigorous analysis of implementation. All we can say at this point is that there is considerable circumstan ial evidence pointing towards Q on both counts.

Conclusion

This has been a very long chapter, and yet it has left a lot of loose ends to be tidied up. There is more work to be done at several points. We have however developed at least the skeleton of a theory for comparing "real" systems with abstract systems and operational semantics with abstract semantics.

Conclusion

It is now time to look back over the work in this thesis, in order both to identify the points where further work is desirable and to make a few comparisons with similar work elsewhere. We will first go through the topics covered roughly in the order in which they have been presented, and later make a few general remarks.

Chapter one was of an introductory nature. The language it introduces is of a rather abstract type, though as stated there is no reason why more conventional languages should not be given semantics in the model introduced (or in the other models used later). It is possible to define congruent semantics for the same language over different domains. Indeed the present author has done this in two ways over Scott domains, both without continuations (presented in (j)) and with continuations. The first of these two was implemented by S.D. Brookes using Mosses' S.I.S. system, confirming amongst other things the awfulness of the "palindromes" example (1.19(iii)). The formal semantic techniques used in this chapter are useful but by no means essential for the abstract language used in this thesis, but are almost indispensable when dealing with more conventional languages with more advanced use of variables and perhaps jumps. We will return to the subject of such languages shortly.

The proof rules introduced in chapter two, amplified in chapters three and five, and used throughout this thesis seem to have certain advantages over the more common, and very similar type which requires us to prove a (possibly vacuous) predicate R_o of fix(F) and the relation "$R_i(A) \Rightarrow R_{i+1}(F(A))$" to be able to deduce $\forall i.R_i(\text{fix}(F))$. In the cases of our simpler rules, when used with respect to restriction operators, most of the advantages are aesthetic: proofs tend to be more elegant and there is no need to break up a predicate R into infinitely many R_i. The more advanced forms, for example those described in 2.21, 2.29, 2.33 and 2.34, seem to arise more naturally from the type of rule adopted here. The more abstract cases (where strong and extra continuity are used) do not appear to translate at all into the other type.

A further advantage of this type of rule is the form of the conditions required of predicates for them to be amenable to proof. The considerable sharpening of our insight gained in chapter three is a direct consequence of this. As in the case of chapter two this chapter and its appendix seem to be a fairly comprehensive treatment of its subject. The questions raised at the end of the appendix, while interesting from a mathematical point of view, can have little bearing on any practical work. The topology generated by extra-continuity (which coincides with strong continuity over countable alphabets) is of a type which has found several other uses in computer science; it is for example practically the same as the "Cantor topology" of Plotkin (i).

Chapter four is a summary of some of the author's contributions to the foundations of the non-deterministic model. Because this was a joint project much material desirable for a proper understanding of this model is missing. The set-theoretic principle proposed in 4.10 and proved in 4.15 is clearly the main result of this chapter. In addition to the proof of propositional compactness which forms part of 4.15 there are several other cases where 4.10 can be substituted for (the strictly stronger) Zorn's lemma in proofs of standard results in a natural way. Examples of this are the ultrafilter lemma and the classic paradoxical dissection of the unit sphere. (Of course the fact that we have the double implication in 4.15 places it exactly in the known hierarchy, but it is nevertheless pleasing that our result proves other results in natural ways.) The results of this chapter will apply equally to the revised model suggested at the end of chapter eight.

The proof rules introduced in chapter five are of course the same as those of chapter two, so the same comments apply. The main thing which is missing is a topological study of the non-deterministic model in the spirit of chapter three. If this were done it seems likely that the results obtained would be very similar to those of chapter three, with the exception that in a domain without a "top" it is impossible to obtain any of the proof rules derived from strong and extra continuity. It is clear from chapter eight that monotonic predicates play a special (though by no means exclusive)

role over both the model of chapter four and the similar model of the last section of chapter eight. The study of these will certainly generate further (weaker) topologies over our spaces and may well allow us to develop further proof rules. (The topology of continuous monotonic predicates will be very similar to that of 3.21.)

The discussion of buffers in chapter five is more or less self contained. The techniques developed for the study of buffers are likely to have much wider applications, however. It should not require more than a few adjustments to transfer most of the work of chapter five to the revised model Q suggested in chapter eight. As suggested earlier the transfer of the work in chapter six will require a little more care because of the reduced constructiveness of $(A \| a::B)$ in its second argument. It would be interesting to compare the work done in chapter six on the elimination of network chatter over M with conditions required to prove the absence of divergence over Q. Hopefully it can be proved that the two are more or less equivalent: the implication of absence of divergence by the absence of network chatter would be a justification of our definition of network chatter. There is also the point that if divergence is excluded by conditions such as E_1^a then any fixpoint is greater than (divergence-free) CHAOS; the coincidence of the two systems of operators over the region $\{A \mid A \supseteq CHAOS\}$ could then be used to justify the old analysis of such processes as Quicksort (6.9) over the new model.

The work in the first half of chapter seven requires few comments. Translation to the revised model should bring a few improvements. Since the "matrix" operator is not obviously suited to recursive use it is likely that its theory and the proof techniques for processes defined using it will be more akin to those used with "$\gg$" than those of "$(A \| a::B)$". The advantage of our non-deterministic models M and Q when deadlock is studied is the extremely simple way in which it manifests itself (i.e. $\Sigma \in A(s)$). There is clearly room for much work in extending the results and techniques developed in this section. It is also possible to study other, similar predicates such as "liveness", the predicate demanding that

it is always possible for a process to return to its initial state. (The author has developed several methods for proving this predicate, which is rather less easy to prove than the absence of deadlock.)

Chapter eight is a summary of some of the author's most recent work, linked by the common theme of the study of the relationships between "real" systems and their models. It is of a less complete nature than the other chapters, several of its sections requiring further work. Despite this the author feels the conclusions reached are too important to leave out. Further formal analysis is required especially in the sections on the connection between weak-postulate spaces and P,Q-spaces, operators over universal spaces and the analysis of particular models. It is of course still necessary to formally **analyze** the revised model Q in the contexts of the earlier chapters. It would also be interesting to compare our relational "universal spaces" with other objects such as powerdomains (Plotkin (i), Smyth (1)) used to model non-determinism. Several comparative studies might be possible, such as an examination of the operational semantics of Cousot (b), and with CCS. It will be especially interesting to compare the treatments of diverging processes. (The author believes that Brookes (a) will contain some analysis of the connections between the model M and CCS.) There must also be comparisons between our study of processes through the tests they pass (E_1^c, E_2^c and E_3^c) and the work of Kennaway (g).

In addition to the specific points mentioned in the above paragraphs where further work is required, there are several wider fields where further work is desirable. The first of these is the application of our various methods to more "realistic" versions of C.S.P., with assignment to variables, "if.." statements, less rich recursion (and perhaps jumps?). These could probably be tackled best by a denotational semantics over the model Q in the style of chapter one.

Secondly it might be desirable to attempt to formalize some of the rather ad hoc proof techniques into more systematic methods. A good example of such methods is the technique developed in 6.16 - 6.18 for proving E_i^a and C_i^a. Some interesting formal rules over the deterministic models have been produced by Zhou (n).

Thirdly it would be interesting and probably instructive to attempt to apply our methods to some larger examples than those which we have used as illustrative examples. These might include the specification and justification of communications protocols, and the proof of correctness of a model operating system.

References

(a) S.D. Brookes: D.Phil thesis, to appear.

(b) P. Cousot and R. Cousot: Semantic analysis of communicating sequential processes; ICALP 80 proceedings Springer-Verlag.

(c) C.A.R. Hoare: Communicating sequential processes; Comm. ACM 21, 8(1978).

(d) C.A.R. Hoare: Communicating sequential processes, in "Construction of programs" Ed. McKeag & MacNaughton, C.U.P. 1980.

(e) C.A.R.Hoare, S.D. Brookes and A.W. Roscoe: A theory of communicating sequential processes; Technical monograph PRG 16, May 1981 (Also to appear in an extended form in J.A.C.M.).

(f) M. Hennessy and R. Milner: On observing nondeterminism and concurrency; ICALP 80 proceedings, Springer-Verlag.

(g) J.R. Kennaway and C.A.R. Hoare: A theory of nondeterminism, ICALP 80 proceedings, Springer-Verlag.

(h) C. Mead and L. Conway: Introduction to VLSI systems, Addison-Wesley 1980.

(i) G.D. Plotkin: A powerdomain construction, SIAM Journal on computing 5, Vol.3, pp.452-487, 1976.

(j) A.W. Roscoe: D.Phil. qualifying dissertation, 1979.

(k) D.S. Scott: Data types as lattices, SIAM Journal on computing 5 1976, pp.522-587.

(l) M.B. Smyth: Power domains; J. Comp. Syst. Sci. 16, (1978).

(m) S. Willard: General topology, Addison-Wesley 1968.

(n) Zhou Chou Chen and C.A.R. Hoare: Partial correctness of communicating sequential processes, Proc. Int. Conf. on distributed computing, April 1981.

In addition to the above works, which were specifically referred in the text, there are several others which have had a significant influence. Amongst these are the following.

(1) J.H. Conway: Regular algebra and finite machines, Chapman and Hall, 1971.

(2) E.W. Dijkstra: A discipline of programming, Prentice-Hall 1976.

(3) H.B. Enderton: Elements of set theory, Academic Press, 1977.

(4) K. Kuratowski: Introduction to set theory and topology, Permagon Press. (Especially chapter XIX, but beware the false theorem 2.4.)

(5) R. Milner: A calculus of communication systems, Springer Verlag 1980.

(6) D.S. Scott: Mathematical theory of computation, Oxford University lecture notes, Michaelmas term 1980.

(7) J.E. Stoy: Denotational Semantics, M.I.T. press 1977.

There are two further categories of works which should be mentioned. The first of these are numerous locally circulated documents, particularly those of C.A.R. Hoare and S.D. Brookes. The second category comprises the sources from which the author has borrowed the ideas for many of his worked examples.