SYNTHESIS OF COMMUNICATING BEHAVIOUR

by

Robin Milner

University of Edinburgh

April 1978

1. Introduction

We aim to explore the concept of communicating behaviour in a manner which refrains from specifying the nature of behaviours, except in the following imprecise way:

A behaviour represents the possible activities
of an agent which may communicate with its environment;
the environment - or part of it - may also be represented
by a behaviour.

By "communication" between two behaviours, we mean a value-passing act which they perform simultaneously. Our approach is synthetic; we propose algebraic operations which construct behaviours from behaviours, and which both (i) appear a priori to be meaningful in terms of our imprecise specification of behaviour, and (ii) correspond - not necessarily in an exact sense - with ways in which computing agents themselves (hardware or software) are built from component agents.

Dana Scott, in his Lattice of Flow Diagrams [7], did in effect the same thing for sequential activity, without the notions of communication or concurrency. He proposed the following operations over sequential activities:

(i) A constant I, the null action.

(ii) A binary operation of concatenation; (a_1;a_2)
means "do a_1 first, then a_2 ".

(iii) A family T of binary operations, where T is a given set which we may call tests; for t \in T, t(a_1,a_2) means
"if t holds do a_1 , otherwise do a_2 ". Let us write
   t(a_1,a_2) as "t \rightarrow a_1,a_2 ".

Now in terms of the imprecise notion of activity (its sequentiality) and the correspondingly imprecise meaning of the operations, it is reasonable to entertain laws which they satisfy; the laws go some way towards characterizing the concept of sequentiality. Scott entertained the following, among others:

   a; I = I; a = a

   a_1;(a_2;a_3) = (a_1;a_2);a_3

   (t \rightarrow a_1,a_2);a_3 = t \rightarrow (a_1;a_3),(a_2;a_3)
Further, actions may be defined as the solution of equations - that is, recursively. For example

$$a^* = t \rightarrow (a; a^*),$$

defines what we would normally call "while t do a". With the help of induction, one may use the laws to prove properties, such as

$$a^* ; a^* = a^*$$

and these properties are ensured for any algebra of activities which satisfies the laws and admits the solution of recursive equations.

In this paper, we embark on the same programme for communicating behaviours. Whether the aim is fully achieved cannot be settled mathematically; it will always depend on what algebras we wish to admit as behaviour algebras, and (dependent on this) whether our operations can define a rich enough subalgebra. To propose operations and laws therefore amounts to proposing a class or variety of behaviour algebras and - for each of them - a definable subalgebra; the outcome can then be assessed by whatever non-mathematical criteria we possess.

An algebraic treatment of concurrency has been advocated by Polish authors; see for example Mazurkiewicz [3] and Winkowski [8]. Their work bears a closer relation with Petri's Net Theory than ours; we say a little more about this later. The present work differs also in its more explicit treatment of communication.

Our approach here is rather informal. In the next section we look more closely at the vague concept of communicating behaviour, set up a little notation for communication, and discuss the kind of behaviour operations we want. Section 3 introduces these operations one by one in the course of an exercise in behaviour expression. In Section 4 we propose laws and comment on their adequacy.

Acknowledgment. I am grateful for the hospitality of the Computer Science Department at Aarhus while writing this report, and for their encouragement and criticism. I am also indebted to Tony Hoare and Dana Scott for prompting some of these ideas, and to Gordon Plotkin and Mogens Nielsen for many helpful discussions.