Applied Categorical Structures **4:** 343–360, 1996.
© 1996 *Kluwer Academic Publishers. Printed in the Netherlands.*

Matrices, Machines and Behaviors*

STEPHEN L. BLOOM [1],** N. SABADINI[2] and R. F. C. WALTERS[3]
[1] *Stevens Institute of Technology, Department of Computer Science, Hoboken, NJ 07030, U.S.A.*
[2] *Departimento di Scienze dell'Informazione, University di Milano, Milano, Italia*
[3] *Sydney University, Department of Mathematics, Sydney, Australia*

(Received: 5 December 1994; accepted: 12 September 1995)

Abstract. Sabadini, Walters and others have developed a categorical, machine based theory of concurrency in which there are four essential aspects: a distributive category of data-types; a bicategory **Mach** whose objects are data types, and whose arrows are input-output machines built from data types; a semantic category (or categories) **Sem**, suitable to contain the behaviors of machines, and a functor, "behavior": **Mach**→ **Sem**. Suitable operations on machines and semantics are found so that the behavior functor preserves these operations. Then, if each machine is decomposable into primitive machines using these operations, the behavior of a general machine is deducible from the behavior of its parts.The theory of non-deterministic finite state automata provides an example of the paradigm and also throws some light on the classical theory of finite state automata.

We describe a bicategory whose objects are natural numbers, in which an arrow $M\colon\ n \to p$ is a finite state automaton with n input states, p output states, and some additional internal states; we require that no transitions begin at output states or end at input states. A machine is represented by an $q+n$ by $q+p$ matrix. The bicategory supports additional operations: non-deterministic choice, parallel interleaving, and feedback. Enough operations are imposed on machines to show that each machine may be obtained from some atomic ones by means of the operations.

The semantic category is the (Bloom–Ésik) iteration theory $\mathbf{Mat}_{\mathcal{L}(X^*)}$ whose objects are natural numbers and whose arrows from n to p are $n \times p$ matrices with entries in the semiring of languages. The behavior functor associates to a machine $M\colon\ n \to p$ a matrix $|M|$ of languages, one language to each pair of input and output states. Behavior preserves composition, feedback, takes non-deterministic choice to union, and parallel-interleaving to shuffle. Thus, behavior gives a compositional semantics to a primitive notion of concurrent processes.

Mathematics Subject Classifications (1991). 18B20, 68Q05, 68Q68.

Key words: Bicategories, finite automata, iteration theories, compositional semantics.

1. Introduction

This paper arose from the happy connection between two research programmes. The first programme is the study of *iteration theories*, initiated by Calvin Elgot [4], and elaborated and extended in a long series of papers by Bloom and Ésik. (Most of their results are contained in the monograph [2].) The second programme

* This work has been supported by the Australian Research Council, by CEC under grant number 6317, ASMICS II, by Italian MURST, and by the Italian CNR.

** Visit to Sydney supported by a grant from the Australian Research Council.

† Presented at the European Colloquium of Category Theory, Tours, France, 25–31 July 1994.

is the development of a categorical, machine-based analysis of concurrency, by Sabadini, Walters and others [6, 7, 5, 8].

In this second programme, there are four essential aspects.

- a distributive category of data-types;
- a bicategory **Mach** whose objects are data types, and whose arrows are input-output machines built from data types;
- a semantic category (or categories) **Sem**, suitable to contain the behaviors of machines,
- and a functor, "behavior": **Mach**→ **Sem**.

Part of the programme is to find suitable operations on machines and semantics so that the behavior functor preserves these operations. Then, if each machine is decomposable into primitive machines using these operations, the behavior of a general machine is deducible from the behavior of its parts (compositionality).

During Bloom's visit to Sydney in March, 1994, the above ideas were clarified and we realized that iteration theories were precisely categories of the semantic kind mentioned above. In fact an example of a machine category and various behavior categories already exist in the case that the machines are flowcharts.

But we realized also, and this is the content of the present paper, that the theory of non-deterministic finite state automata provides a very simple, clear and elegant example of the paradigm. We believe this example is worth writing out in detail, at the very least as a guide to the development of the theory of the much more complicated examples which incorporate data types, true parallelism, etc. We think this example also throws some light on the classical theory of finite state automata.

We describe a bicategory whose objects are natural numbers, in which an arrow $M\colon n \to p$ is a finite state automaton with n input states, p output states, and some additional internal states; we require that no transitions begin at output states or end at input states. A machine is represented by an $q + n$ by $q + p$ matrix. The bicategory supports additional operations: non-deterministic choice, parallel interleaving, and feedback. Enough operations are imposed on machines to show that each machine may be obtained from some atomic ones by means of the operations.

The semantic category is precisely the iteration theory $\mathbf{Mat}_{\mathcal{L}(X^*)}$ whose objects are natural numbers and whose arrows from n to p are $n \times p$ matrices with entries in the semiring of languages. The behavior functor associates to a machine $M\colon n \to p$ a matrix $|M|$ of languages, one language to each input-output pair. Behavior preserves composition, feedback, takes non-deterministic choice to union, and parallel-interleaving to shuffle. Thus, behavior gives a compositional semantics to a primitive notion of concurrent processes.

Among all machines we pick out a subcategory of *normal* machines and prove that the behavior functor, restricted to these machines, is the universal functor out of the machine category which collapses the 2-cells. The classical automata theory behind this proof is the theory of minimal deterministic automata (adapted from [1] to this more general situation).

2. Matrices

We denote the fact that A is an $n \times p$ matrix over a semiring S by

$$A\colon n \to p.$$

We **compose** $A\colon n \to p$ with $B\colon p \to q$ using matrix multiplication to obtain $A \cdot B\colon n \to q$. Thus, matrices over S form a category, $\mathbf{Mat}_S$. In this category, the objects are the nonnegative integers and the integer $n+p$ is both the categorical coproduct and product of n and p. We refer to [3] for all details.

Remark 2.1. We occasionally treat $\mathbf{Mat}_S$ as a 2-category, all of whose vertical morphisms are identities.

A 0–1 matrix is one whose entries are either 0 or 1. We identify a 0–1 matrix $\rho\colon n \to p$ with the set-theoretic relation $\rho'\colon [n] \to [p]$ defined by:

$$i\ \rho'\ j \Leftrightarrow \rho_{ij} = 1.$$

We say a 0–1 matrix is functional, injective, surjective, when the corresponding relation has that property. Recall that a semiring S is *ω-complete* if, for any countable family $(a_i)_{i\in I}$ of elements of S, there is an element $\sum_{i\in I} a_i$ such that

$$\begin{aligned}
\sum_{i\in[n]} a_i &= a_1 + \cdots + a_n,\\
\sum_{(i,j)\in I\times J} a_i b_j &= \Big(\sum_{i\in I} a_i\Big)\cdot\Big(\sum_{j\in J} b_j\Big),\\
\sum_{i\in I} a_i &= \sum_{j\in J}\Big(\sum_{i\in I_j} a_i\Big),
\end{aligned}$$

where I is the union of the pairwise disjoint sets I_j, $j \in J$. It follows that $\sum \emptyset = 0$ and that summation is commutative, associative and completely distributive. Any ω-complete semiring S may be enriched by a star operation

$$\begin{aligned}
*\colon S &\to S,\\
s &\mapsto s^*,\\
&:= 1 + s + s^2 + \cdots
\end{aligned}$$

which satisfies a number of identities (see [3]). In particular, we mention the star sum and product identities:

$$\begin{aligned}
(x+y)^* &= (x^*y)^*x^*,\\
(xy)^* &= 1 + x(yx)^*y.
\end{aligned}$$

Further, one may extend the star operation to square matrices over S by either defining

$$A^* := 1 + A + A^2 + \cdots$$

or by using the "pairing identity":

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^* = \begin{pmatrix} a^* + a^*b(d+ca^*b)^*a^*b & a^*b(d+ca^*b)^* \\ (d+ca^*b)^*ca^* & (d+ca^*b)^* \end{pmatrix}, \tag{1}$$

where a and d are square. The square matrices also satisfy the star sum and product identities.

We will be mainly interested in matrices over the semiring $\mathcal{L}(X^*)$ of subsets of words on a finite alphabet X. In this semiring, the star operation both on $\mathcal{L}(X^*)$ and on the square matrices over $\mathcal{L}(X^*)$ satisfies the additional equations

$$\begin{aligned} 1^* &= 1, \\ x^* &= (1+x)^*. \end{aligned}$$

We define a **feedback operation** on matrices as follows: if A is the $k+m$ by $k+n$ matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

where a is $k \times k$, b is $k \times n$, etc., then $A^\dagger \colon m \to n$ is the matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^\dagger := ca^*b + d.$$

If there may be confusion as to the dimensions of the square array a, we may write $\dagger^k$ for the feedback operation. Notation: For $i \in [n]$, the matrix i_n is the $1 \times n$ 0–1 matrix with a 1 only in position i. For any $n \times p$ matrix a, its transpose is denoted a^T, as usual.

2.1. SHUFFLE PRODUCT OF MATRICES

In this section, we assume that all matrices are over the semiring $\mathcal{L}(X^*)$ of subsets of X^*. We want to define the shuffle product $A \otimes B$ of two matrices. We need some preliminary definitions.

If $u, v \in X^*$, we define the subset $u \otimes v$ of X^* inductively by:

$$\begin{aligned} 1 \otimes u &:= u \otimes 1 = \{u\}, \\ (xu) \otimes (yv) &:= x(u \otimes yv) + y(xu \otimes v), \end{aligned}$$

where $x, y \in X$. If now $U, V \subseteq X^*$,

$$U \otimes V := \bigcup_{u \in U, v \in V} u \otimes v.$$

We note the following facts about this operation. For all $u, v, w \in \mathcal{L}(X^*)$,

$$\begin{aligned} u \otimes v &= v \otimes u, \\ u \otimes (v \otimes w) &= (u \otimes v) \otimes w, \\ u \otimes (v + w) &= (u \otimes v) + (u \otimes w), \\ 0 \otimes u &= 0, \\ 1 \otimes u &= u. \end{aligned}$$

If $x \in X^*$, and if B: $n \to p$ is a matrix over $\mathcal{P}(X^*)$, then $x \otimes B$: $n \to p$ is the matrix whose entries are given by the equation

$$(x \otimes B)_{ij} := x \otimes B_{ij}.$$

Finally, if A: $n \to p$ and B: $m \to q$, then $A \otimes B$: $nm \to pq$ is the matrix indicated:

$$A \otimes B := \begin{pmatrix} A_{11} \otimes B & A_{12} \otimes B & \dots & A_{1p} \otimes B \\ \vdots & \vdots & & \vdots \\ A_{n1} \otimes B & A_{n2} \otimes B & \dots & A_{np} \otimes B \end{pmatrix}$$

(Thus, it is being assumed that the pairs of integers in $[n] \times [m]$ and $[p] \times [q]$ are being enumerated in lexicographic order.) Note that if A and B are 0–1 matrices, so is $A \otimes B$.

3. Machines

3.1. THE MODEL

In this section, we consider the question of how to model a machine with n initial states, p exit states, and q "internal states" by a matrix whose entries are finite subsets of $X \cup \{1\}$. (We will write X_1 to abbreviate $X \cup \{1\}$.)

One possible definition of a **machine from** n **to** p is the following: a machine

$$M = (Q, X, b, e, \delta)$$

consists of a finite set of states Q, a set of n begin states, say $b_1, \dots, b_n$ in Q, a set of p exit states in Q, say $e_1, \dots, e_p$, and a function

$$\delta\colon\ Q \times X_1 \to \mathcal{P}(Q)$$

subject to the following restrictions.

- The sets $\{b_1, \dots, b_n\}$ and $\{e_1, \dots, e_p\}$ are disjoint.
- For each exit state e_i, and each $x \in X_1$, $\delta(e_i, x) = \emptyset$.
- For each begin $b_i, x \in X_1$, $b_i \notin \delta(q, x)$, any $q \in Q$.

In graph theoretic terms, according to this definition, a begin is not the target and an exit is not the source of an arrow.

The notion of a path from state q to state q' and the label of such a path is defined in the standard way. The **behavior** $|M|$ of the machine M is the $n \times p$ matrix over the semiring of (regular) subsets of X^* in which $|M|_{ij}$ is the set of all words which are labels of paths from b_i to e_j.

With this understanding of a machine, we may give an equivalent description of a machine n to p as a $t+n$ by $t+p$ matrix whose entries are finite subsets of X_1. Indeed, suppose that there are t states which are neither begin nor exit states, and they are enumerated as $q_1, \ldots, q_t$. Then define the $t \times t$ matrix a by:

$$a_{ij} := \{x \in X \cup \{1\}\colon\ q_j \in \delta(q_i, x)\}.$$

Define the $n \times t$ matrix c by:

$$c_{ij} := \{x \in X \cup \{1\}\colon\ q_j \in \delta(b_i, x)\}.$$

b is the $t \times p$ matrix

$$b_{ij} := \{x \in X \cup \{1\}\colon\ e_j \in \delta(q_i, x)\}.$$

Last, let d be the $n \times p$ matrix

$$d_{ij} := \{x\colon\ e_j \in \delta(b_i, x)\}.$$

We write

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Using this matrix description, we may express the behavior of M by the equation:

$$|M| = ca^*b + d. \tag{2}$$

We now *define* a machine M over X from n to p with weight q as a $q+n$ by $q+p$ matrix whose entries are finite subsets of X_1. More precisely, M is the pair (q, A), where A is the matrix. We usually identify M with A. The behavior $|M|$ of M is defined by (2).

3.2. MACHINE OPERATIONS

Suppose that $M\colon\ n \rightarrow p$ and $N\colon\ p \rightarrow r$ are machines of weight q and q', respectively. Write

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

$$N = \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix},$$

where a: $q \to q$ and q': $q' \to q'$, etc. We define two operations of composition on machines. The machine $M \circ N$: $n \to r$ of weight $q + q' + p$ is defined by:

$$M \circ N := \begin{pmatrix} a & 0 & b & 0 \\ 0 & a' & 0 & b' \\ 0 & c' & 0 & d' \\ c & 0 & d & 0 \end{pmatrix}. \tag{3}$$

Second, define $M \cdot N$: $n \to r$ of weight $q + q'$ by

$$M \cdot N := \begin{pmatrix} a & bc' & bd' \\ 0 & a' & b' \\ c & dc' & dd' \end{pmatrix}. \tag{4}$$

Remark 3.1. The machine operation $M \cdot N$ has the defect that some of the entries in $M \cdot N$ may be *products* of letters, not just single letters. Thus, we must enlarge the definition of machine to include finite sets of words as entries in our matrices. But note that in the case that each of the matrices b, c, d, b', c', d' are 0–1 matrices, and if entries in a and a' are sets of letters only, the matrix $M \cdot N$ has no entries which contain words of length more that one, and dc', bd', dd' are also 0–1 matrices. We will use this fact below, see Proposition 4.5.

First, we show that there is a reason to prefer the second definition (4) for composition, since with this definition the machines form a category whose objects are the nonnegative integers. For each $n \geqslant 0$, let id_n: $n \to n$ be the machine of weight 0, consisting of the identity matrix.

PROPOSITION 3.2. *For machines M: $n \to p$, N: $p \to r$, P: $r \to s$ of weights q_1, q_2, q_3 respectively,*

$$\begin{aligned} \mathsf{id}_n \cdot M &= M, \\ M \cdot \mathsf{id}_p &= M, \\ (M \cdot N) \cdot P &= M \cdot (N \cdot P). \end{aligned}$$

Proof. We prove the third statement only, since the first two are easy to see. Suppose that

$$\begin{aligned} M &= \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \\ N &= \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix}, \\ P &= \begin{pmatrix} a'' & b'' \\ c'' & d'' \end{pmatrix}. \end{aligned}$$

Then

$$M \cdot N = \begin{pmatrix} a & bc' & bd' \\ 0 & a' & b' \\ c & dc' & dd' \end{pmatrix}$$
$$= \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

where A is $q_1 + q_2$ by $q_1 + q_2$, etc. Then

$$(M \cdot N) \cdot P = \begin{pmatrix} A & Bc'' & Bd'' \\ 0 & a' & b'' \\ C & Dc'' & Dd'' \end{pmatrix}$$
$$= \begin{pmatrix} a & bc' & bd'c'' & bd'd'' \\ 0 & a' & b'c'' & b'd'' \\ 0 & 0 & a'' & b'' \\ c & dc' & dd'c'' & dd'd'' \end{pmatrix}. \tag{5}$$

Similarly,

$$N \cdot P = \begin{pmatrix} a' & b'c'' & b'd'' \\ 0 & a'' & b'' \\ c' & d'c'' & d'd'' \end{pmatrix}$$

so that

$$M \cdot (N \cdot P) = \begin{pmatrix} a & bc' & bd'c'' & bd'd'' \\ 0 & a' & b'c'' & b'd'' \\ 0 & 0 & a'' & b'' \\ c & dc' & dd'c'' & dd'd'' \end{pmatrix}. \tag{6}$$

Comparing Equations (5) and (6), we see that composition is associative. □

If M and N are both machines $n \to p$, of weights q and q', respectively, we define the machine $M + N$: $n \to p$ of weight $q + q'$ by:

$$M + N := \begin{pmatrix} a & 0 & b \\ 0 & a' & b' \\ c & c' & d + d' \end{pmatrix}.$$

Now suppose that M: $k + n \to k + p$ is a machine of weight q. We define the k-feedback of M, $M^{\uparrow k}$: $n \to p$ as the same matrix, considered as a machine with weight $q + k$. Formally, if $M = (q, A)$: $k + n \to k + p$, then $M^{\uparrow} = (q + k, A)$: $n \to p$.

We introduce two final machine operations. Suppose that M: $n \rightarrow p$ and N: $m \rightarrow p$ are machines of weights q, q', respectively. Write

$$\begin{aligned} M &= \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \\ N &= \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} \end{aligned}$$

as usual. The machine $\langle M, N\rangle$, the **source tupling** of M and N, is defined as the following machine $n + m \rightarrow p$ of weight $q + q'$:

$$\langle M, N\rangle := \begin{pmatrix} a & 0 & b \\ 0 & a' & b' \\ c & 0 & d \\ 0 & c' & d' \end{pmatrix}.$$

If M: $n \rightarrow p$ and N: $n \rightarrow q$, the **target tupling** $[M, N]$: $n \rightarrow p + q$ is the machine with weight $q + q'$:

$$[M, N] := \begin{pmatrix} a & 0 & b & 0 \\ 0 & a' & 0 & b' \\ c & c' & d & d' \end{pmatrix}.$$

Remark 3.3. Note that if $M = [d]$ and $N = [d']$ both have weight 0, then

$$\begin{aligned} \langle M, N\rangle &= \langle d, d'\rangle \\ &= \begin{pmatrix} d \\ d' \end{pmatrix}, \\ [M, N] &= [d, d']. \end{aligned}$$

We omit the proof that both operations are associative.

PROPOSITION 3.4.

$$\begin{aligned} \langle M, \langle N, P\rangle\rangle &= \langle\langle M, N\rangle, P\rangle, \\ [M, [N, P]] &= [[M, N], P], \end{aligned}$$

when M, N, P are machines with appropriate sources and targets.

For each x in X, let x: $1 \rightarrow 1$ denote the machine of weight 0 consisting of the 1×1 matrix whose only entry is x. The machine 0: $1 \rightarrow 1$ of weight 0 has 0 as its only entry. The operations of composition, $+$, source and target tupling, together with the feedback operation are complete in the following sense.

PROPOSITION 3.5. *The least class of machines containing the identity* $\mathsf{id}_1\colon 1 \rightarrow 1$, *the zero* $0\colon 1 \rightarrow 1$ *and the atomic machines* $x\colon 1 \rightarrow 1$, *for each* $x \in X$, *which is closed under the operations of composition,* $+$, *source and target tupling, and the feedback operation is the class of all machines.*

Proof. Given a machine $M\colon n \rightarrow p$ of weight k, we can write $M = N^{\uparrow k}$, for a unique machine $N\colon k+n \rightarrow k+p$ of weight 0. But

$$N = \langle N_1, \ldots, N_{k+n} \rangle,$$

where the machines $N_i\colon 1 \rightarrow k+p$ are the rows of N. Further, each row can be written as the target tupling of its entries.

$$N_i = [N_{i1}, \ldots, N_{i,k+n}].$$

Now each entry N_{ij} is a finite set of words on X, and can be obtained from the atoms, zero and identity by using the composition and sum operations. □

3.3. SHUFFLE PRODUCT OF MACHINES

The shuffle product of machines is given its own section, since it is not needed to construct an arbitrary machine from atomic machines.

Suppose that $M\colon n \rightarrow p$ has weight s, and $N\colon m \rightarrow q$ has weight t. We define the machine $M \otimes N\colon nm \rightarrow pq$ of weight st as follows. Write

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

$$N = \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix},$$

where $a\colon s \rightarrow s$, $a'\colon t \rightarrow t$, etc. Then

$$M \otimes N := \begin{pmatrix} (a + \mathsf{id}_s) \otimes (a' + \mathsf{id}_t) & b \otimes b' \\ c \otimes c' & d \otimes d' \end{pmatrix}.$$

For later use, note that when all matrices except a and a' are 0–1 matrices, then only the upper left block of the matrix for $M \otimes N$ may contain entries in $X^* - \{1\}$.

Remark 3.6. Note that in general, the machines $M \otimes N$ and $M \cdot N + N \cdot M$ are distinct.

3.4. NORMAL MACHINES

A subclass of the machines will play an important role: the normal machines. A machine $M\colon n \rightarrow p$ of weight q

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

is a **normal machine** if b, c are 0–1 matrices and d is the zero matrix. This collection of machines is closed under composition. Normal machines will usually be denoted

$$M = (\alpha, A, \beta)\colon\ n \twoheadrightarrow p,$$

where $\alpha\colon\ n \twoheadrightarrow s$ and $\beta\colon\ s \twoheadrightarrow p$ are 0–1 matrices, and $A\colon\ s \twoheadrightarrow s$ is the transition matrix of the internal states.

Remark 3.7. Normal machines have the following set-theoretic description as generalized automata. Suppose that $M = (\alpha, A, \beta)\colon\ n \twoheadrightarrow p$ is a normal machine with weight s. For each $i \in [n]$, the i-th row of the 0–1 matrix α determines a subset I_i of $[s]$, namely, the set of those $j \in [s]$ with $\alpha_{ij} = 1$; similarly, for each $j \in [p]$, the j-th column of β determines the subset $E_j = \{k \in [s]\colon\ \beta_{kj} = 1\}$. For a fixed i, j, the resulting nondeterministic automaton with states $Q = [s]$

$$\mathcal{A}_{ij} = (Q, X, I_i, E_j, \delta)$$

recognizes precisely the set of all words in

$$(i_n \cdot \alpha) \cdot A^* \cdot (\beta \cdot j_p^T),$$

the entries in the i-th row and j-th column of the behavior of M. Hence, the behavior of M is determined by np pairs of subsets of the set of states $[s]$. The set-theoretic counterpart of M is a generalized automaton with indexed collections of initial and final states. It will be denoted

$$(Q, X, \{I_i\}_{i\in[n]}, \{E_j\}_{j\in[p]}, \delta).$$

Remark 3.8. We indicate the set-theoretic version of $M \otimes N$ in the case that both M and N are normal machines. Suppose that the automata corresponding to the two machines are

$$\begin{aligned} \mathcal{A} &= (Q, X, \{I_i\}, \{E_j\}, \delta), \\ \mathcal{B} &= (Q', X, \{I'_{i'}\}, \{E'_{j'}\}, \delta'). \end{aligned}$$

Then the automaton corresponding to $M \otimes N$ is

$$\mathcal{A} \otimes \mathcal{B} = (Q \times Q', X, \{I''_{ii'}\}, \{E''_{jj'}\}, \delta''),$$

where

$$\begin{aligned} I''_{ii'} &= \{(q, q')\colon\ q \in I_i \text{ and } q' \in I'_{i'}\}, \\ E''_{jj'} &= \{(q, q')\colon\ q \in E_j \text{ and } q' \in E'_{j'}\}, \\ (q_1, q'_1) \in \delta''((q, q'), x) &\Leftrightarrow q_1 \in \delta(q, x) \quad \text{and} \quad q'_1 = q' \\ &\qquad \text{or} \quad q'_1 \in \delta'(q', x) \quad \text{and} \quad q_1 = q. \end{aligned}$$

4. Behaviors

In this section, we show that the operations defined on machines are preserved by the behavior map.

We consider composition first.

PROPOSITION 4.1. *For machines M: $n \rightarrow p$ and N: $p \rightarrow r$,*

$$\begin{aligned} |M \circ N| &= |M \cdot N| \\ &= |M| \cdot |N|. \end{aligned}$$

Proof. We use the by now familiar matrix notation for M and N.

$$\begin{aligned} |M \circ N| &= (c\ 0\ d) \begin{pmatrix} a^* & a^*bc'a'^* & a^*b \\ 0 & a'^* & 0 \\ 0 & c'a'^* & 1 \end{pmatrix} \begin{pmatrix} 0 \\ b' \\ d' \end{pmatrix} \\ &= ca^*bc'a'^*b' + dc'a'^*b' + ca^*bd' + dd' \\ &= (ca^*b + d)(c'a'^*b' + d') \\ &= |M||N|. \end{aligned}$$

Also,

$$\begin{aligned} |M \cdot N| &= (c\ dc') \begin{pmatrix} a^* & a^*bc'a'^* \\ 0 & a'^* \end{pmatrix} \begin{pmatrix} bd' \\ b' \end{pmatrix} + dd' \\ &= ca^*bd' + ca^*bc'a'^*b' + dc'a'^*b' + dd' \\ &= (ca^*b + d)(c'a'^*b' + d') \\ &= |M||N|. \end{aligned}$$

□

We omit the easy calculation which proves

PROPOSITION 4.2. *For machines M, N: $n \rightarrow p$,*

$$|M + N| = |M| + |N|.$$

Next, we show that feedback is preserved by behavior.

PROPOSITION 4.3. *Suppose that M: $k + n \rightarrow k + p$ has weight q. Then*

$$|M^{\uparrow k}| = |M|^{\uparrow k}.$$

Proof. Write

$$M = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & k \end{pmatrix},$$

where a is $q \times q$, b is $q \times k$, c is $q \times p$, etc. Then

$$\begin{aligned} |M| &= \begin{pmatrix} d \\ g \end{pmatrix} \begin{pmatrix} a^*b & a^*c \end{pmatrix} + \begin{pmatrix} e & f \\ h & k \end{pmatrix} \\ &= \begin{pmatrix} e + da^*b & f + da^*c \\ h + ga^*b & k + ga^*c \end{pmatrix}. \end{aligned}$$

Thus,

$$|M|^{\uparrow} = (h + ga^*b)\,\xi\,(f + da^*c) + (k + ga^*c), \tag{7}$$

where

$$\xi := (e + da^*b)^*.$$

Now

$$\begin{aligned} |M^{\uparrow}| &= (g\ h) \begin{pmatrix} a & b \\ d & e \end{pmatrix}^* \begin{pmatrix} c \\ f \end{pmatrix} + k \\ &= (g\ h) \begin{pmatrix} a^* + a^*b\,\xi\,da^* & a^*b\xi \\ \xi\,da^* & \xi \end{pmatrix} \begin{pmatrix} c \\ f \end{pmatrix} + k, \end{aligned}$$

using the pairing identity (1). Hence, after some calculation,

$$\begin{aligned} |M^{\uparrow}| &= ga^*c + ga^*b\,\xi\,da^*c + \\ &\quad + h\,\xi\,da^*c + ga^*b\,\xi\,f + h\,\xi\,f + k. \end{aligned} \tag{8}$$

It is not hard to check now that (7) and (8) are the same. □

Last, we turn to source and target tupling.

PROPOSITION 4.4. *Behavior preserves source and target tupling:*

$$\begin{aligned} |\langle M, N\rangle| &= \langle |M|, |N| \rangle, \\ |[M, N]| &= [|M|, |N|], \end{aligned}$$

when M, N have the appropriate sources and targets.

Proof. We prove only the first statement. Suppose that M: $n \to p$, N: $m \to p$ have weights q, q', respectively. Then

$$\begin{aligned} |\langle M, N\rangle| &= \begin{pmatrix} c & 0 \\ 0 & c' \end{pmatrix} \begin{pmatrix} a^* & 0 \\ 0 & a'^* \end{pmatrix} \begin{pmatrix} b \\ b' \end{pmatrix} + \begin{pmatrix} d \\ d' \end{pmatrix} \\ &= \begin{pmatrix} ca^*b + d \\ c'a'^*b' + d' \end{pmatrix} \\ &= \langle |M|, |N| \rangle. \end{aligned}$$

□

4.1. NORMAL MACHINES ARE SUFFICIENT

The category of all machines has a subcategory with the same descriptive power: the normal machines.

PROPOSITION 4.5. *For any machine* $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$: $n \rightarrow p$ *there is a normal machine* $M' = \left(\begin{smallmatrix} A & \beta \\ \alpha & 0 \end{smallmatrix}\right)$: $n \rightarrow p$ *with* $|M| = |M'|$.

Proof. We identify the machine M with a labeled graph. For each arrow $b_i \rightarrow e_j$ labeled $x \in X_1$ from a begin to an exit, add two new internal states, say q, q' and add the following arrows: $b_i \rightarrow q$, labeled 1, $q \rightarrow q'$ labeled x, and $q' \rightarrow e_j$, labeled 1. Now delete the original arrow. Clearly the behavior of the new machine is the same as that of M. Thus, by induction on the number of X_1-labeled arrows from begins to exits, we can assume that we have an equivalent machine with no such arrows. We remove X-labeled arrows from begins to internal states and from internal states to exits in the same way. If $b_i \rightarrow q$ is labeled x, add a new internal state, say q' and an arrow $b_i \rightarrow q'$ labeled 1 and an arrow $q' \rightarrow q$ labeled x. Now remove the original arrow. □

Thus, from now on we restrict attention to normal machines.

PROPOSITION 4.6. *Suppose that* M: $n \rightarrow p$ *and* N: $m \rightarrow q$ *are normal machines of weights* s *and* t *respectively. Then*

$$|M \otimes N| = |M| \otimes |N|.$$

Proof. Using the set-theoretic description in Remark 3.8, the result is clear. □

We note that if (α, A, β): $n \rightarrow p$ is a normal machine with some occurrences of 1 in A_{ij}, for some i, j, we can write

$$A = B + \gamma,$$

where B has no occurrences of 1, and γ is a $s \times s$ 0–1 matrix. But then,

$$\begin{aligned} |M| &= \alpha A^* \beta \\ &= \alpha(\gamma^* B)^* \gamma^* \beta, \end{aligned}$$

by the star-sum identity. But γ^* is another $s \times s$ 0–1 matrix. So if we define $C := \gamma^* B$, and $\beta' := \gamma^* \beta$, the normal machine

$$M' := (\alpha, C, \beta')$$

has the same behavior as M and the transition matrix C contains no 1's.

DEFINITION 4.7. Suppose that $M = (\alpha, A, \beta)$: $n \rightarrow p$ is a normal machine of weight s; M is **deterministic** if each row of α contains at most one occurrence of 1, and for each $i \in [s]$, $x \in X$, the set $\{j \in [s]\colon x \in A_{ij}\}$ has size at most 1. Lastly, $1 \notin A_{ij}$, $i, j \in [s]$.

For each normal machine M: $n \to p$, we show how to find a deterministic machine MD: $n \to p$ with the same behavior. Let $M = (\alpha, A, \beta)$ be a normal machine such that A contains no occurrence of 1's. Consider the corresponding automaton

$$(Q, X, \{I_i\}_{i \in [n]}, \{E_j\}_{j \in [p]}, \delta).$$

Let Q' consist of the nonempty subsets of Q, and define the transition function δ': $Q' \times X \to Q'$ as usual by:

$$\delta'(U, x) := \begin{cases} \bigcup_{q \in U}\{q' \colon\ q' \in \delta(q, x)\} & \text{if nonempty,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The set I_i, if nonempty, is a point in Q', and is defined as the i-th begin. The set E'_j is defined as the collection of those subsets of Q which contain some state in E_j. The corresponding triple of matrices is the deterministic machine MD.

DEFINITION 4.8. Suppose that $M = (\alpha, A, \beta)$ and $N = (\alpha', B, \beta')$ are normal machines $n \to p$ of weights s and t, respectively. A **morphism**

$$\rho\colon\ M \to N$$

is a 0–1 matrix $s \to t$ such that the following diagram commutes in $\mathbf{Mat}_{\mathcal{L}(X^*)}$.

$$\begin{tikzcd}
 & s \arrow[r, "A"] \arrow[dd, "\rho"] & s \arrow[dd, "\rho"'] \arrow[dr, "\beta"] & \\
n \arrow[ur, "\alpha"] \arrow[dr, "\alpha'"'] & & & p \\
 & t \arrow[r, "B"'] & t \arrow[ur, "\beta'"'] &
\end{tikzcd}$$

To say that the left-hand triangle commutes means that the matrix product $\alpha \cdot \rho$ is α'; i.e., for all $i, j \in [n] \times [t]$,

$$\alpha'_{ij} = \sum_{k=1}^{s} \alpha_{ik}\rho_{kj}.$$

Remark 4.9. ρ: $(\alpha, A, \beta) \to (\alpha', B, \beta')$ is a morphism iff the following matrix equation holds.

$$\begin{pmatrix} A & \beta \\ \alpha & 0 \end{pmatrix} \cdot \begin{pmatrix} \rho & 0 \\ 0 & \mathrm{id}_p \end{pmatrix} = \begin{pmatrix} \rho & 0 \\ 0 & \mathrm{id}_n \end{pmatrix} \cdot \begin{pmatrix} B & \beta' \\ \alpha' & 0 \end{pmatrix}.$$

Remark 4.10. Note that if

$$\rho\colon M \dashrightarrow N \quad \text{and} \quad \rho'\colon N \dashrightarrow P$$

are morphisms between the normal machines M, N, P, then the matrix product $\rho\rho'$ is a morphism from $M \dashrightarrow P$. In fact, it is easy to check that the collection of all normal machines together with the identity machines id_n, $n \geqslant 0$, form a 2-category, **NMach**, with the 0–1 matrices as vertical morphisms.

Note that if $M = (\alpha, A, \beta)\colon n \dashrightarrow p$ is a normal machine, then

$$M^T := (\beta^T, A^T, \alpha^T)\colon p \dashrightarrow n$$

is a normal machine. Further, if $\rho\colon M \dashrightarrow N$ is a morphism, so is $\rho^T\colon N^T \dashrightarrow M^T$.

Let $\sim$ be the least equivalence relation on machines $n \dashrightarrow p$ such that if there is a relational morphism $\rho\colon M \dashrightarrow N$ then $M \sim N$. Thus, $M \sim N$ iff there is a sequence of machines M_i, $i = 0, 1, \ldots, k$, and relational morphisms ρ_i such that $M = M_0$, $N = M_k$ and $\rho_i\colon M_i \dashrightarrow M_{i+1}$ or $\rho_i\colon M_{i+1} \dashrightarrow M_i$, for each $i < k$.

LEMMA 4.11. *If $M \sim N$, then $|M| = |N|$.*

Proof. It is enough to show that if $\rho : M \dashrightarrow N$ is any relational morphism, then $|M| = |N|$. But using the fact that the matrices over an ω-complete semiring satisfy the implication

$$A\rho = \rho B \Rightarrow A^*\rho = \rho B^*, \tag{9}$$

(for any ρ), we compute:

$$\begin{aligned} |M| &= \alpha A^* \beta \\ &= \alpha A^* \rho \beta' \\ &= \alpha \rho B^* \beta' \\ &= \alpha' B^* \beta' \;=\; |N|. \end{aligned}$$

□

We prove the converse holds as well.

THEOREM 4.12. *If two normal machines $M, N\colon n \dashrightarrow p$ have the same behavior, then $M \sim N$. Thus $|M| = |N|$ iff $M \sim N$.*

Proof. Suppose that $\mathcal{A} = (Q, X, \{I_i\}, \{E_j\}, \delta)$ is the automaton corresponding to $M = (\alpha, A, \beta)$ and that $\mathcal{D} = (Q', X, \{I_i'\}, \{E_j'\}, \delta')$ is the equivalent deterministic machine constructed above. Let $M' = (\alpha', A', \beta')$ be the machine corresponding to $\mathcal{D}$. Let $\rho\colon Q' \dashrightarrow Q$ be the relation defined by:

$$U \,\rho\, q \Leftrightarrow q \in U.$$

We show ρ is a morphism. Indeed, $(\alpha' \cdot \rho)_{iq} = 1$ iff $q \in I_i$ iff $\alpha_{iq} = 1$. Secondly,

$$\begin{aligned} x \in (A' \cdot \rho)_{U,q} &\Leftrightarrow \exists V \quad (x \in A'_{U,V} \text{ and } q \in V) \\ &\Leftrightarrow \exists q' \in U \, \exists V \quad (x \in A_{q',q} \text{ and } q \in V) \\ &\Leftrightarrow x \in (\rho \cdot A)_{U,q}. \end{aligned}$$

Last,

$$\begin{aligned} \beta'_{Uj} = 1 &\Leftrightarrow \exists q \quad (q \in U \text{ and } q \in E_j) \\ &\Leftrightarrow (\rho \cdot \beta)_{Uj} = 1. \end{aligned}$$

Further, if $\mathcal{A}$ is the accessible part of $\mathcal{D}$, then the inclusion $\mathcal{A} \to \mathcal{D}$ and the quotient map $\mathcal{A} \to \mathcal{R}$ to the minimal deterministic machine are both (functional) morphisms. Since two automata with the same behavior determine the same minimal deterministic machine, the result follows. □

Remark 4.13. In [1], it is shown that $\sim$ is the least equivalence relation such that if either $M \xrightarrow{\rho} N$ or $M \xrightarrow{\rho^T} N$, where ρ is a functional morphism, then $M \sim N$.

Remark 4.14. The minimal automaton M for a collection $R_{ij} \subseteq X^*$ of regular languages, indexed by the pairs in $[n] \times [p]$, may be constructed as follows. The states in M are p-tuples of languages. For each $i \in [n]$ and each word $w \in X^*$, let

$$q^i_w := (w^{-1}R_{i1}, \ldots, w^{-1}R_{ip}).$$

Let $Q\colon = \{q^i_w\colon\ i \in [n],\ w \in X^*\}$. The i-th begin is the point q^i_λ; the set E_j is the set of all q^i_w with $w \in R_{ij}$; the transition function is defined by

$$\delta(q^i_w, a) := q^i_{wa}.$$

It may now be shown that if M' is any deterministic, accessible machine $n \to p$ which recognizes the collection R_{ij}, then there is a unique morphism from M' to this one.

COROLLARY 4.15. *Behavior,* $M \mapsto |M|$*, is a 2-functor from* **NMach***, the 2-category of normal machines to* $\mathbf{Mat}_{\mathcal{L}(X^*)}$*, which necessarily collapses all vertical arrows to identities. Further, if* F*:* $NMach \to \mathcal{C}$ *is any other such functor,* F *factors uniquely through the behavior functor.* □

References

1. Bloom, S. L. and Ésik, Z.: Equational axioms for regular sets, *Mathematical Structures in Computer Science* 3 (1993), 1–24.

2. Bloom, S. L. and Ésik, Z.: *Iteration Theories: The Equational Logic of Iterative Processes*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1993.
3. Bloom, S. L. and Ésik, Z.: Matrix and matricial iteration theories, Part I, *Journal of Computer and System Sciences* **46** (1993), 381–408.
4. Elgot, C. C.: Monadic computation and iterative algebraic theories, in J. C. Shepherdson (ed.), *Logic Colloquium 1973, Studies in Logic*, Vol. 80, North-Holland, Amsterdam, 1975.
5. Khalil, W. and Walters, R. F. C.: An imperative language based on distributive categories II, in *Informatique Théorique et Applications*, 1993.
6. Sabadini, N. and Walters, R. F. C.: On functions and processors: an automata-theoretic approach to concurrency through distributive categories, in *School of Mathematics and Statistics Research Reports,* University of Sydney (93-7), November 1993.
7. Sabadini, N., Walters, R. F. C., and Weld, Henry: Distributive automata and asynchronous circuits, May 1993, CTCS93. Available by anonymous `ftp` at `ghost.sm.dsi.unimi.it` in the directory `pub2/papers/sabadini` and at `maths.su.oz.au` in the directory `sydcat/papers/walters`.
8. Walters, R. F. C.: *Categories and Computer Science*, Carslaw Publications (1991), Cambridge University Press (1992), 1991.