REWRITING TABLES AND MEMORIZED SEMIRINGS

Gérard H.E. Duchamp$^{(1)}$, *Khalaf Khatatneh*$^{(2)}$, *Cyrille Bertelle*$^{(3)}$

$^{(1)}$Institut Galilée - University of Paris XIII - 99, avenue Jean-Baptiste Clément,
93430 Villetaneuse, France.
$^{(2)}$ Al-Balqa'Applied University - Al-Salt, 19117, Jordan.
$^{(3)}$ LIH - University of Le Havre - 25 rue Ph. Lebon, BP 540, 76058 Le Havre cedex, France

Keywords: Tables, k-subsets, efficient data structures, efficient algebraic structures.

1. INTRODUCTION

The following is intended to be a contribution in the area of what could be called *efficient algebraic structures* or *efficient data structures*. In fact, we define and construct a new data structure, the tables (abstract and reduced), which are at first special multisets of two-raws arrays. The first raw is filled with words (or more generally, elements taken in some semigroup) and the second with some coefficients taken in a semiring [7].

Reduced tables realize the (finite) k-sets sets of Eilenberg [6], they are versatile (one can vary the monomials and the coefficients), easily implemented and fast computable. Varying the scalars and the transformations on them, one can obtain many different structures and, among them, semirings. Examples will be provided and worked out in full detail.

As an application, we present a new semiring (with several semiring structures) which can be applied to the necessity of automatic processing multi-agents behaviour problems. The purpose of this account/paper is to expound also the basic elements of these new structures from a combinatorial point of view. These structures show a bunch of properties. They will be endowed with several laws namely : Sum, Hadamard product, Cauchy product, Fuzzy operations (min, max, complemented product). Two groups of applications are presented.

The first group is linked to the process of "forgetting" information in the tables and then obtaining, for instance, a memorized semiring. The latter is specially suited to solve the *shortest path with addresses* problem using the repeated squaring algorithm on matrices with entries in this semiring.

The second, linked to multi-agent systems, is announced by showing a methodology to manage emergent organization from individual behaviour models.

2. DESCRIPTION OF THE DATA STRUCTURE

2.1. Tables

The input alphabet being set by the automaton under consideration, we will here rather focus on the definition of semirings providing transition (or transfer) coefficients. Indeed, in many applications, we have to compute the weights of paths in some weighted graph (shortest path problem, enumeration of paths, cost computations, automata, transducers to cite only a few) and the computation goes with two main rules: multiplication in series (i.e. along a path), and addition in parallel (if several paths are involved). This involves the least structure endowed with two operations $(k, +, \times)$ and for which matrix computation remains valid, say the structure of a semiring [7].

A table T is a two-rows array, the first row being filled with monomials (upper row) taken in some given semigroup (see [4], [8] or [9]). This is equivalent to saying that a Table is a finite *multiset* of columns.

Multisets are extensively used in rewriting theory and sometimes named *bags*. A multiset $M = s_1, s_2, \cdots$ is a set in which elements are allowed to be repeated which amounts to the data of a mapping $f : Dom(f) \mapsto \mathbb{N}^+$, $Dom(f)$ is the *support* of the multiset and f the multiplicity function. If the support is finite, so is called the multiset.

The set of monomials which are present in the first row will be called the *indices* of the table $(I(T))$ and for the second row $(V(T))$ the *values* or (*coefficients*) of the table. The order of columns is not relevant (as in Computer Algebra Systems where this data structure is impemented).

In the sequel, we will consider two types of laws: pointwise type (subscript $_p$) and convolution type (subscript $_c$).

In order to define the pointwise and convolution composition, we must first construct one of the two central features of the paper, namely the reduction system $\mathbf{red}$.

3. DESCRIPTION OF THE DATA STRUCTURE

3.1. Tables

We will first concentrate on the definition of semirings providing transition (or transfer) coefficients. For convenience, we first begin with various examples of laws on $\mathbf{R}_+ := [0, +\infty[$ including

1. $+$ (ordinary sum)
2. $\times$ (ordinary product)
3. min (if over $[0, 1]$, with neutral 1, otherwise must be extended to $[0, +\infty]$ and then, with neutral $+\infty$) or max
4. $+_a$ defined by $x +_a y := log_a(a^x + a^y)$ $(a > 0)$
5. $+_{[n]}$ (Hölder laws) defined by $x +_{[n]} y := \sqrt[n]{x^n + y^n}$
6. $+^s$ (shifted sum, $x +^c y := x + y - 1$, over whole $\mathbf{R}$, with neutral 1)
7. $\times^c$ (complemented product, $x + y - xy$, can be extended also to whole $\mathbf{R}$, stabilizes the range of probabilities or fuzzy $[0, 1]$ and is distributive over the shifted sum)

For other examples and applications see, for example [7].
A table T is a two-rows array, the first row being filled with monomials taken in a given semigroup S (see [4], [8] or [9]). To be more precise, if

$$T = \begin{array}{c|c|c|c} u_1 & u_2 & \cdots & u_k \\ \hline p_1 & p_2 & \cdots & p_k \end{array}$$

individual columns are allowed to be repeated that is, for instance that one can get

$$\begin{array}{c|c|c|c|c} \cdots & u_i & \cdots & u_i & \cdots \\ \hline \cdots & p_i & \cdots & p_i & \cdots \end{array}$$

and columns commute between themselves, that is

$$\begin{array}{c|c|c|c|c} \cdots & u_i & \cdots & u_j & \cdots \\ \hline \cdots & p_i & \cdots & p_j & \cdots \end{array} = \begin{array}{c|c|c|c|c} \cdots & u_j & \cdots & u_i & \cdots \\ \hline \cdots & p_j & \cdots & p_i & \cdots \end{array}$$

This is equivalent to saying that a Table is a finite *multiset* of columns.
Multisets [11] are extensively used in rewriting theory and sometimes named *bags*. A multiset $M = s_1, s_2, \cdots$ is a set in which elements are allowed to be repeated which amounts to the data of a mapping $f : Dom(f) \mapsto \mathbf{N}^+$, $Dom(f)$ is the *support* of the multiset and f the multiplicity function. If the support is finite, so is called the multiset.

The set of words which are present in the first row will be called the *indices* of the table ($I(T)$) and for the second row the *values* or (*coefficients*)of the table. The order of the columns is not relevant (as in Computer Algebra Systems where this data structure is impemented).
In the sequel, we will consider two types of laws:
pointwise type (subscript $_p$) and convolution type (subscript $_c$).
In order to define the pointwise and convolution composition, we must first construct the one of the two central features of the paper, namely the reduction system $\mathbf{red}$.

3.2. The reduction system red

3.2.1. Reduction

Let T be a table with indices in a semigroup S and values in a commutative semigroup $(k, +)$. If T owns two columns $c_i = \begin{array}{|c|} u_i \\ \hline p_i \end{array}$ $i = 1, 2$ with the same index $u_1 = u_2 = u$, we define the reduction

$$T \overset{\mathbf{red}}{\longrightarrow} T - \{c_1, c_2\} + c_3$$

where $c_3 = \begin{array}{|c|} u \\ \hline p_1 + p_2 \end{array}$

Definition/Proposition 1 *i) The reduction system defined above, denoted* $\mathbf{red}$*, is Noetherian and Confluent.*
ii) The result of the complete application to the process to a table T

$$T \overset{\mathbf{red}^*}{\longrightarrow} T_1$$

will be denoted $\mathbf{red}(T)$*.*

3.2.2. Operations on tables

Let us consider, two tables T_1, T_2 and a commutative and associative law $*$ over the coefficients. One has the proposition

Proposition 2 *i) The reduction of $T_3 := T_1 \uplus T_2$ (multiset union) depends only on the reduction of T_i; $i = 1, 2$.*
ii) More precisely set

$$R_1 = \mathbf{red}(T_1) = \begin{array}{c|c|c|c} u_1 & u_2 & \cdots & u_k \\ \hline p_1 & p_2 & \cdots & p_k \end{array}$$

and

$$R_2 = \mathbf{red}(T_2) = \begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_l \\ \hline q_1 & q_2 & \cdots & q_l \end{array}$$

then $\mathbf{red}(T_1 \uplus T_2) = R_1 \boxplus_p R_2 = R_3$ *where* $R_3[w] = T_i[w]$ *if* $w \in I(T_i)$ *and* $w \notin I(T_{3-i})$ *and by* $R_3[w] = T_1[w] * T_2[w]$ *if* $w \in I(T_1) \cap I(T_2)$

In particular one has $I(R_1 \boxplus_p R_2) = I(R_1) \cup I(R_2)$*.*

Note 1 *i) At this stage one does not need any neutral. The structure automatically creates it (the empty table, see* algebraic remarks *below).*
ii) The above is a considerable generalization of an idea appearing in [3], aimed only to semirings with units.
iii) If $(k, +)$ possesses a neutral 0_k, one can add to $\mathbf{red}$ *an extra step of withdrawing the columns with coefficient (or value) 0_k. The new system is Noetherian and Confluent and avoids the creation of another neutral.*

For convolution type, one needs two laws over k, say $\oplus, \otimes$, the second being distributive over the first, i.e. identically

$$\begin{aligned} x \otimes (y \oplus z) &= (x \otimes y) \oplus (x \otimes z) \text{ and} \\ (y \oplus z) \otimes x &= (y \otimes x) \oplus (z \otimes x) \end{aligned} \quad (1)$$

Taking into account that a table is, in a unique way a (commutative) sum of columns, it is sufficient to define the convolution $\otimes$ for two columns, which reads

$$\left|\frac{u_1}{p_1}\right| \bigotimes_c \left|\frac{u_2}{p_2}\right| := \left|\frac{u_1 u_2}{p_1 \times p_2}\right| \tag{2}$$

The constructed structure $\mathcal{T}(S, k)$ for tables is then a semiring (generally without multiplicative unit), the two laws are compatible with the reduction i.e.

Proposition 3 *Let* $T_i;\ i = 1, 2$
i) The reduction of $T_3 := T_1 \otimes T_2$ *depends only of the reduction of* $T_i;\ i = 1, 2$.
ii) More precisely set

$$R_1 = \mathbf{red}(T_1) = \begin{array}{c|c|c|c} u_1 & u_2 & \cdots & u_k \\ \hline p_1 & p_2 & \cdots & p_k \end{array}$$

and

$$R_2 = \mathbf{red}(T_2) = \begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_l \\ \hline q_1 & q_2 & \cdots & q_l \end{array}$$

then there exists a (unique) law $\otimes_p$ *on the reduced tables such that*

$$\mathbf{red}(T_1 \otimes T_2) = R_1 \otimes_c R_2 = R_3 \tag{3}$$

iii) The space of tables with coefficients in k *and monomials in* S, $(\mathcal{T}(S, k), \uplus, \otimes_c)$ *is a semiring with an additive neutral (the empty table). Moreover, this semiring has a multiplicative unit if* S *is a monoid.*

3.3. Total mass

The total mass of a table is just the sum of the coefficients in the bottom row. One can check that

$$\begin{array}{rcl} mass(T1 \oplus T2) & = & mass(T1) + mass(T2); \\ mass(T1 \otimes T2) & = & mass(T1) \cdot mass(T2) \end{array} \tag{4}$$

this allows, if needed, stochastic conditions.

3.4. Algebraic remarks

We have confined in this paragraph some proofs of structural properties concerning the tables. The reader may skip this section with no serious harm.
First, we deal with structures with as little as possible requirements, i.e. *Magmas* and *Semirings*. For formal definitions, see

`http://encyclopedia.thefreedictionary.com/Magma%20category`

`http://mathworld.wolfram.com/Semiring.html`

Proposition 4 *(i) Let* X *be a set,* $(S, +)$ *a commutative and associative semigroup.*
We denote $T(X, S)$ *the set of tables with indices in* X *and values in* S *and define* $\boxplus_p$ *as in (3.2.2). Then*
i) The law $\boxplus$ *is associative and commutative. Moreover the semigroup* $(\mathcal{T}(X, S), \boxplus)$ *always possesses a neutral, the empty table (i.e. with an empty set of indices).*
ii) If $(k, \oplus, \otimes)$ *is a semiring and* X *a semigroup, then the* $(\mathcal{T}(X, k), \oplus, \otimes)$ *is a semiring.*

Proof. (Sketch) i) To each table T let us associate the (finite supported) function $f_T : X \mapsto S$ defined by

$$f_T(w) = \begin{cases} T[w] & \text{if } w \in I(T) \\ 0_S & \text{otherwise} \end{cases} \tag{5}$$

then, check that $f_{T_1 \boxplus_p T_2} = f_{T_1} + f_{T_2}$ and that the correspondence $T \mapsto f_T$ is a isomorphism.
ii) We use a similar technique to show that the correspondence $\mathcal{T}(X, k) \to k[X]$ is one-to-one.

Note 2 *Pointwise product can be considered as being constructed with respect to the (Hadamard) coproduct* $c(w) = w \otimes w$ *whereas convolution is w.r.t. the Cauchy coproduct*

$$c(w) = \sum_{uv=w} u \otimes v \tag{6}$$

(see [5]).

4. APPLICATIONS

4.1. Specializations and images

Multiplicities, Stochastic and Boolean
Whatever the multiplicities, one gets the classical automata by emptying the alphabet (setting $\Sigma = \emptyset$).
Memorized Semiring
We explain here why the memorized semiring, devised at first to perform efficient computations on the shortest path problem with memory (of addresses) can be considered as an image of a "table semiring" (thus proving without computation the central property of [10]).

Let $\mathcal{T}$ be here the table semiring with coefficients in $([0, +\infty], min, +)$. Then a table

$$T = \begin{array}{c|c|c|c|c} u_1 & \cdots & u_k & \cdots & u_n \\ \hline l_1 & \cdots & l_k & \cdots & l_n \end{array} \tag{7}$$

can be written so that $l_1 = \cdots = l_k < l_m$ for $m > k$ (this amounts to say that the set where the minimum is reached is $\{u_1, u_2 \cdots u_k\}$). Then, to such a table, one can associate $\phi(T) := [\{u_1, u_2 \cdots u_k\}, l_1]$ in the memorized semiring. It is easy to check that ϕ transports the laws and the neutrals and obtain the result.

4.2. Application to evolutive systems

Tables are structured as semirings and are flexible enough to recover and amplify the structures of automata with multiplicities and transducers. They give operational tools for modelling agent behaviour for various simulations in the domain of distributed artificial intelligence [2]. The outputs of automata with multiplicities or the values of tables allow to modelize in some cases agent actions or in other cases, probabilities on possible transitions between internal states of agents behaviour. In all cases, the algebraic structures associated with automata outputs or tables values is very interesting to define automatic computations in respect with the evolution of agents behaviour during simulation.

One of our aims is to compute dynamic multiagent systems formations which emerge from a simulation. The use of table operations delivers calculable automata aggregate formation. Thus, when table values are probabilities, we are able to obtain evolutions of these aggregations as adaptive systems do.

With the definition of adapted operators coming from genetic algorithms, we are able to represent evolutive behaviors of agents and so evolutive systems [1]. Thus, tables and memorized semiring are promizing tools for this kind of implementation which leads to model complex systems in many domains.

5. REFERENCES

[1] Bertelle C., Flouret M., Jay V., Olivier D., Ponty J.-L., *Genetic Algorithms on Automata with Multiplicities for Adaptative Agent Behaviour in Emergent Organisations.*

[2] Bertelle C., Flouret M., Jay V., Olivier D., Ponty J.-L., *Automata with Multiplicities as Behaviour Model in Multi-Agent Simulations* SCI 2001.

[3] Champarnaud J.-M., Duchamp G., *Derivatives of rational expressions and related theorems*, T.C.S. **313** 31 (2004).

[4] Duchamp G., Hatem Hadj Kacem, Éric Laugerotte, *On the erasure of several letter-transitions*, JICCSE'04

[5] Duchamp G., Flouret M., Laugerotte E., Luque J-G., *Direct and dual laws for automata with multiplicites*, Theoret. Comput. Sci. 267 (2001) 105-120.

[6] Eilenberg S., *Automata, languages and machines, Vol A*, Acad. Press (1974).

[7] GOLAN, J. S., *Power algebras over semirings with applications in Mathematics and Computer Science*, Kluwer (1999).

[8] Laugerotte E., Abbad H., *Symbolic computation on weighted automata*, JICCSE'04.

[9] Lothaire M., *Combinatorics on words*, Cambridge University Press (new edition), 1997.

[10] Khatatneh K., *Construction of a memorized semiring*, DEA ITA Memoir, University of Rouen (2003).

[11] `http://mathworld.wolfram.com/Multiset.html`