Project in Rich Composition of Systems

Elemér E. Rosinger

Department of Mathematics
and Applied Mathematics
University of Pretoria
Pretoria
0002 South Africa
eerosinger@hotmail.com

Dedicated to Marie-Louise Nykamp

Abstract

Presently - and surprisingly - only *two* ways to compose systems are known : classical systems have their state spaces composed by Cartesian product, while in the case of quantum systems, by tensor product. Reasons, as well as hints are presented why and how more *rich* ways of composition of systems should be considered. Improving computational power is one such reason, why a hint may come from DNA computation. Mathematically, the issue comes down to choosing a proper way to compose certain classes of finite graphs which include as a particular case the juxtaposition of finite string of words in an alphabet.

1. Motivation

One of the curious and strange facts in the history of science is the long ongoing lack of awareness of the issue of the *composition of systems*, which only came to awareness in the second part of the 20th century.
And still today, in the realms of non-living systems, we only happen to know of *two* ways of such compositions : the classical, and the

quantum, the former given by Cartesian products, and the latter, by tensor products.

Take the example of wheels which - since time immemorial - have been known and composed in various ways into systems, among them carts with two, three or four wheels, for instance. Yet it took a few centuries even after the introduction of Cartesian coordinates in the 17th century, and thus in general, of Cartesian products of sets, until modern Control Theory introduced the concept of *state space* of a system. And then it turned out that the state space S of a classical mechanical system composed of two such systems, with the respective state spaces S' and S'', is given by their *Cartesian product* $S = S' \times S''$.

Amusingly however, even today, few people happen to know that - in Physics - we only know about one single other way to compose state spaces of systems, namely, by *tensor product*. Indeed, given two quantum systems with their respective state space being the Hilbert spaces H' and H'', the composite quantum system which is obtained from them will have the state space given by the Hilbert space resulted from the *tensor product* $H = H' \bigotimes H''$.

So much for all what we happen to know so far in Physics about the ways to *compose* systems, more precisely, to compose the *state spaces* of systems.
Not to mention that most of those involved in various branches of science, engineering, including the mathematicians specialized in tensor products, are not quite aware of the above ...

And then, in that situation, a few decades back, came the growing interest in quantum computation ...

Now, just like usual electronic digital computation, quantum computation is also essentially built upon the composition of a large number of simple systems. In the former case, the simple systems are *bits* which are composed into a *register* where the effective computations take place. In the case of a quantum computer the simple systems are *qubits*, which is a shortened form of "quantum bits". And they also have to be composed - this time as quantum systems - into *quan-*

tum registers. Furthermore, in order to be able to compete with usual electronic digital computers, such quantum registers may have to be the quantum composition of, say, at least two to three hundred qubits.

For more clarity, let us get into some details.

2. The Simple Case of Computers

Let us suppose that the register R_E of a usual electronic digital computer is the classical mechanical composition of $n > 1$ bits $B_1, \ldots, B_n$. Then as is well known, the state space of S_E of R_E is given by the *finite* set

$$(1) \quad S_E = B_1 \times \ldots \times B_n = \{0,1\}^n$$

which has 2^n elements.

In a dramatically different manner, the state space S_Q of a quantum register R_Q which is the quantum composition of the same number $n > 1$ of qubits $Q_1, \ldots, Q_n$, is an *infinite* subset of a 2^n dimensional complex Hilbert space, namely

$$(2) \quad S_Q \underset{\neq}{\subsetneq} \mathbb{C}^{(2^n)} = \mathbb{C}^2 \bigotimes \ldots \bigotimes \mathbb{C}^2$$

where the tensor product has n factors. Thus clearly, if in (2) the usual Cartesian product would be used instead of the tensor product, then we would obtain the space $\mathbb{C}^{(2n)} = \mathbb{C}^2 \times \ldots \times \mathbb{C}^2$, which starting already with $n \geq 3$, and as n grows, is becoming considerably *smaller* that the quantum state space in (2).

The relations (1), (2) can already give a good idea about the considerably increased computational power of quantum computers. And the *essential* fact is that the surprising difference between (1) and (2) comes in a most simple and direct way from the difference between the Cartesian, and on the other hand, the tensor products which are involved.

3. Cartesian versus Tensor Products, and Entanglement as the Gap between them

That considerable difference between (1) and (2) can more generally be seen, for instance, as follows, and it is purely a matter of linear algebra. Let X and Y be two finite dimensional vector spaces over a field $\mathbb{K}$, having the respective dimensions $dim(X) = h$ and $dim(Y) = k$. Then as is well known

(3) $\quad dim(X \times Y) = h + k, \quad dim(X \bigotimes Y) = hk$

therefore, typically, that is, unless $h, k \leq 2$, one obviously has

(4) $\quad dim(X \times Y) < dim(X \bigotimes Y)$

And here comes in one of the *specific* quantum phenomena which is *entanglement* and which turns out to be one of the *basic resources* in quantum computation. Namely, in view of (4), the injective linear mapping

(5) $\quad X \times Y \ni (x, y) \longmapsto x \otimes y \in X \bigotimes Y$

is typically *not* surjective. For convenience, this mapping is identified with the simple *strict inclusion*

(6) $\quad X \times Y \subsetneqq X \bigotimes Y$

and in view of (4), the vast majority of elements in the tensor product $X \bigotimes Y$ are *not* in the Cartesian product $X \times Y$. Their set, namely

(7) $\quad (X \bigotimes Y) \setminus (X \times Y)$

gives the set of all *entangled* elements of the tensor product $X \bigotimes Y$. And as seen from (3), typically, there are *many more* of the entangled elements in a tensor product $X \bigotimes Y$, than there are of the non-entangled elements in the Cartesian product $X \times Y$.

4. "Quantum Like" Systems and their Alternative Ways of Composition ...

One of the major difficulties in the effective construction of quantum computers is the well known quantum phenomenon of *decoherence*. In view of that, in certain quantum computer related circles, the idea arose to find and use so called "quantum like" systems, that is systems which may have some computationally useful properties of the usual quantum system, on the other hand however, do not suffer from the phenomenon of decoherence.
As it happens however, so far, no such systems of sufficient practical relevance have seemingly been presented ...

Nevertheless, the mere idea of considering such "quantum like" systems suggests the consideration of ways of *composing* systems, more precisely, their state spaces, which are *different* from both the classical one given by Cartesian products, and the quantum one, given by tensor products.

Now, a rather a priori requirement for such new ways of composing state spaces of systems is that the respective ways should *not* be much simpler than the tensor products which are involved in the composition of quantum systems. Indeed, we are interested in possibly not losing much from the power of quantum computers. Thus these new compositions should on the scale

(8) Cartesian product − − − − − − − − − − − − tensor product

be rather nearer to the tensor product, than to the Cartesian one.

And then, why not, and in view of (1), (2) above, these new compositions may as well be *more* rich, than the tensor product as well ...

In this regard, preliminary studies can be found in [1-7]. There, the most general and purely algebraic essence of tensor products is attempted to be found, so as to be able to pursue various of its possible generalizations, in this way exploring the part of the above scale (8)

to the right of tensor products.
Further related details can be found in [8,9].

Needless to say, there may as well be ways of composing systems far more rich, and to a good extent outside of the realms grasped by (8) or its more obvious extensions. And as seen in the next subsection 5, such a possibility is suggested by what is called the "DNA computing", [10].

Coming back to tensor products and it mentioned extensions in [1-7]. It is simply to note that the essence of it is in the *free semigroup* of one dimensional finite strings of words over some alphabet, endowed with the simple algebraic operation of concatenation of such strings.

Thus as a rather natural further and much more *rich* extension is suggested by the considering free semigroups of *finite graphs* in which the algebraic operation is some *suitable* composition of graphs.

And here, one is faced with the :

Problem :

Which may be the suitable ways to compose graphs, when it comes to model compositions of state spaces of systems used as registers in computation ?

□

5. A Hint : DNA Computing ?

So far, a distinctly *third* way of computation, namely, *DNA computation* has been considered and studied to some extent. This computation is indeed markedly different from the usual electronic digital one, and also from what quantum computation is supposed to be. And it is based on some of the ways information is processed on the DNA level in *living* organisms. Consequently, it is in principle already effectively implementable, and it may not need more than the use of several living cells, be they each on their own, or being parts of some living organism.

Now, the possible hint for finding more rich ways to compose systems may, indeed, happen to come from DNA computing due to the following.

The structure itself of DNA is the well known so called *double helix* which obviously has the following features :

- it is a rather minor generalization of one dimensional finite strings of words in an alphabet of four letters,

- with a corresponding minor generalization of concatenation as the algebraic operation.

References

[1] Rosinger E E : arXiv:physics/0701116, A General Scheme of Entanglement

[2] Rosinger E E : arXiv:physics/0701246, Further de-Entangling Entanglement

[3] Rosinger E E : arXiv:0807.1436, Two Generalizations of Tensor Products, Beyond Vector Spaces

[4] Rosinger E E : arXiv:0808.0095, Group Invariant Entanglements in Generalized Tensor Products

[5] Rosinger E E : arXiv:0810.3196, The Necessary Structure of Congruences in Free Semigroups

[6] Rosinger E E : arXiv:0812.2533, Magmatic "Quantum-Like" Systems

[7] Khrennikov A, Rosinger E E : arXiv:1203.0412 Extended Tensor Products and Generalization of the Notion of Entanglement

[8] Rosinger E E : Non-Cartesian Systems : an Open Problem. http://hal.archives-ouvertes.fr/hal-00544928, http://vixra.org/abs/1012.0020

[9] Rosinger E E : Further on Non-Cartesian Systems.
http://hal.archives-ouvertes.fr/hal-00546909,
http://vixra.org/abs/1012.0031

[10] WIKIPEDIA : DNA Computing