The algebra and geometry of networks

Davide Maglia, N. Sabadini, Filippo Schiavio and R.F.C. Walters
University of Insubria, Como, Italy

Networks of components have a compositional description in terms of the algebra of symmetric or braided monoidal categories in which each object has a commutative Frobenius or separable algebra structure compatible with the tensor product. They also have a geometric description [11] - the free such algebra (in the separable symmetric case) is the category of cospans of multigraphs, arrows of which have a pictorial representation; in the Frobenius braided case the geometry is more complicated, capturing not only the connection between components but also their entanglement. These results are in the line introduced by Penrose [1], and Joyal and Street [3], and were obtained by Sabadini and Walters with collaborators Katis and Rosebrugh in earlier work, especially [6–9, 11, 12], beginning with the work on relations with Carboni [2] in 1987. The work has numerous antecedents - we mention just S. Eilenberg, S.L. Bloom, Z. Esik, Gh. Stefanescu. The algebra has connections with quantum field theory ([10]).

The present work presents two developments. The first is some initial work in classifying tangled circuits; the second is a tool for composing cospans of graphs and calculating executions of nets of parallel automata.

1 Tangled circuits

The free braided algebra of the type described above was introduced by Rosebrugh, Sabadini and Walters [14] under the name Tangled Circuits since the geometry captures not only the connection between components but their entanglement.

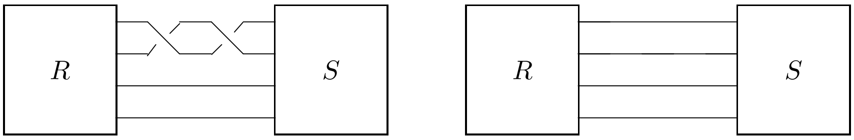
A *commutative Frobenius algebra* in a braided monoidal category with twist τ ([4]) consists of an object G and four arrows $\nabla : G \otimes G \to G$, $\Delta : G \to G \otimes G$, $n : I \to G$ and $e : G \to I$ making (G, ∇, e) a monoid, (G, Δ, n) a comonoid and satisfying the equations

$$(1_G \otimes \nabla)(\Delta \otimes 1_G) = \Delta\nabla = (\nabla \otimes 1_G)(1_G \otimes \Delta) : G \otimes G \to G \otimes G$$
$$\nabla\tau = \nabla : G \otimes G \to G$$
$$\tau\Delta = \Delta : G \to G \otimes G$$

A *multigraph* M consists of two sets M_0 (objects, vertices or wires) and M_1 (arrows, edges or components) and two functions $dom : M_1 \to M_0^*$ and $cod : M_1 \to M_0^*$ where M_0^* is the free monoid on M_0.

Given a multigraph M the free braided strict monoidal category in which the objects of M are equipped with commutative Frobenius algebra structures is called $\mathbf{TCircD}_M$. Its arrows are called *tangled circuit diagrams*, or more briefly circuit diagrams. In the case that M has one vertex and no arrows we will denote $\mathbf{TCircD}_M$ simply as $\mathbf{TCircD}$.

Determining whether two arrows are equal in this category is difficult since it seems to include the problem of classifying knots as a special case. We present some initial work in classifying a special class of arrows which we call blocked braids, that is, an arrow of the form $S \circ B \circ R$ where $R : I \to X^n$ (X^n the tensor power of X), $B : X^n \to X^n$ is a braid on n strings, and $S : X^n \to I$. Here are two examples of distinct blocked braids:



We show that blocked braids on less than four strings are finite in number, whereas with four or more strings the number is infinite. (Notice that there are an infinite number of braids on two strings, whereas there are only two blocked braids on two strings: $\tau = \tau^{-1}$ and 1.)

The method of proof that two tangled circuits are distinct is by associating invariants to tangled circuits - if we manage to find distict invariants then certainly the circuits are distinct. The invariants we use are in fact functors which preserve the braided monoidal and Frobenius structure from the category of tangled circuits to simpler categories such as the following category of *tangled relations*:

Let G be a group. The objects of $\mathbf{TRel}_G$ are the formal powers of G, and the arrows from G^m to G^n are relations R from the set G^m to the set G^n satisfying:

1) if $(x_1, ..., x_m)R(y_1, ...y_n)$ then also for all g in G
$(g^{-1}x_1g, ..., g^{-1}x_mg)R(g^{-1}y_1g, ..., g^{-1}y_mg)$,
2) if $(x_1, ..., x_m)R(y_1, ...y_n)$ then $x_1...x_m(y_1...y_n)^{-1} \in Z(G)$ (the center of G).

Composition and identities are defined to be composition and identity of relations.

It is straightforward to verify that $\mathbf{TRel}_G$ is a category. We introduce some useful notation. Write $x = (x_1, ..., x_m)$, $y = (y_1, ..., y_n)$, and so on. Write $\overline{x} = x_1x_2...x_m$ and for g, h in G, as $g^h = hgh^{-1}$. For g in G write $x^g = (x_1^g, x_2^g, ..., x_m^g)$. Thus, $(\overline{x})^g = \overline{x^g}$, and of course for any x, y in $G^m \times G^n$, $x^g y^g = (xy)^g$ where we write xy for $(x_1, ..., x_m, y_1, ..., y_n)$.

Then $\mathbf{TRel}_G$ is a braided strict monoidal category with tensor defined on objects by $G^m \otimes G^n = G^{m+n}$ and on arrows by product of relations. The twist

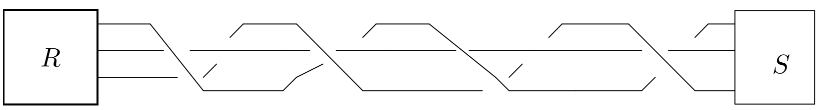
$$\tau_{m,n} : G^m \otimes G^n \to G^n \otimes G^m$$

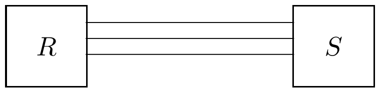
is the functional relation

$$(x, y) \sim (y^{\overline{x}}, x)$$

The results we mention above are obtained by choosing suitable groups and tangled relations to distinguish tangled circuits.

Instead, an interesting equation in the Tangled Circuit category is the so-called Dirac's belt trick, the unwinding of two full twists of a belt without rotating the ends; it amounts to the equality of the following two blocked braids:





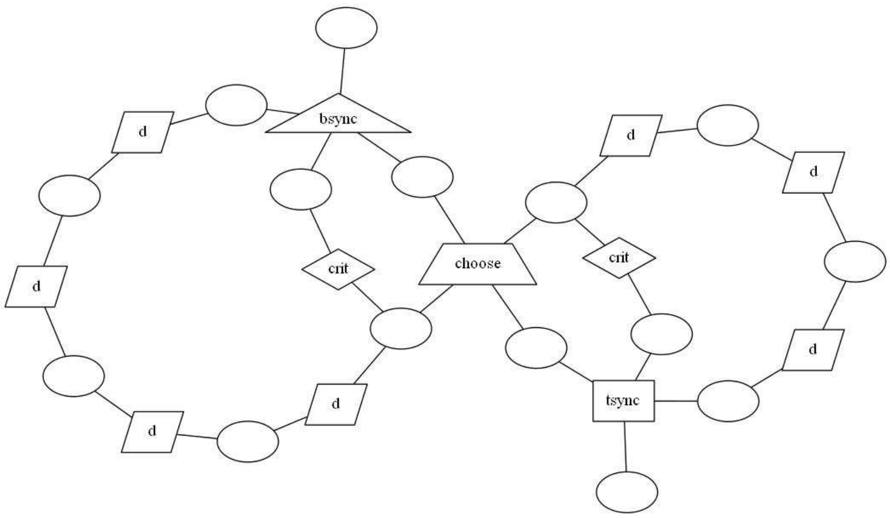
The category of tangled circuits may be relevant to modelling quantum computing with anyons, in a way similar to [13].

2 A graphic tool for executing nets of automata

The second development is a graphic tool for calculating compositionally cospans of multi-graphs and for searching the state space of sequential and parallel networks of automata (the components have state). The state space of a sequential network is a colimit, and of a parallel network is a limit [12]. This has close connections with our paper [5] and recent work of Sobocinski [15, 16] on Petri nets.

The tool defines a language for describing expressions in *Span*(*Graph*) ([6]). However instead of evaluating the expression (with associated explosion of state) it calculates the geometry of the expression by composing in *Cospan*(*Graph*), with a graphic output using GraphViz, and permits (incomplete) exploration of the state space.

The following is an example of the (static) graphic output with the geometry of a system:



As an example of the language for specifying components notice that the component *bsynch* above is defined by

```
bsync=span(3,1){
 00, 01, 10, 11
 00 -> 00 : 0,0,0/0
 00 -> 01 : 0,0,1/0
 00 -> 11 : 0,3,1/0
 10 -> 11 : 5,0,1/0
 10 -> 10 : 5,0,0/0
 01 -> 11 : 5,3,0/0
 01 -> 01 : 5,0,0/0
 11 -> 00 : 0,0,0/2}
```

Further the system as a whole is defined (where D=Δ, ID=identity, U=unit, CU=counit, d=delay, crit=critical delay) by

```
diagonal_delay = crit ; D
expr = (tsync * bsync) ; (diagonal_delay * diagonal_delay) ; (ID * choose * ID) ;
               ( (d;d;d) * ID * ID * (d;d;d;d))
system = (U * ID * ID * U) ; (ID * U * ID * ID * ID * ID * U * ID) ;
         (ID * ID * expr * ID * ID) ; (ID * CU * ID * ID * CU * ID) ; (CU * CU)
```

For more details, and an example of the execution of this (mutual exclusion) system see the post at *http://rfcwalters.blogspot.it/2012/01/petri-nets.html*.

References

[1] R. Penrose, *Applications of negative dimensional tensors*, in Combinatorial mathematics and its applications, Academic Press (1971).

[2] A. Carboni and R.F.C. Walters, *Cartesian bicategories I*, Journal of Pure and Applied Algebra, 49 (1987) 11-32.

[3] A. Joyal, R.H. Street, *The geometry of tensor calculus I*, Adv. Math. 88 (1991) 55-112.

[4] A. Joyal, R.H. Street, *Braided monoidal categories*, Adv. Math. 102 (1993) 20-87.

[5] P. Katis, N. Sabadini, R.F.C. Walters, *Representing P/T nets in Span(Graph)*, Proceedings AMAST '97, SLNCS 1349 (1997) 307-321.

[6] P. Katis, N. Sabadini, R.F.C. Walters, *Span(Graph): an algebra of transition systems*, Proceedings AMAST '97, SLNCS 1349 (1997) 322-336.

[7] P. Katis, N. Sabadini, R.F.C. Walters, *A formalisation of the IWIM Model*, in: Proc. COORDINATION 2000,(Eds.) Porto A., Roman G.-C., LNCS 1906, Springer Verlag, (2000) 267-283.

[8] P. Katis, N. Sabadini, R.F.C. Walters, *On the algebra of systems with feedback and boundary*, Rendiconti del Circolo Matematico di Palermo Serie II, Suppl. 63 (2000), 123-156.

[9] P. Katis, N. Sabadini, R.F.C. Walters, *Compositional minimization in Span(Graph): Some examples*, Electronic Notes in Theoretical Computer Science 104C,(2004) 181-197.

[10] J. Kock, *Frobenius algebras and 2D Topological Quantum Field Theories*, Cambridge University Press (2004).

[11] R. Rosebrugh, N. Sabadini, R.F.C. Walters, *Generic commutative separable algebras and cospans of graphs*, Theory and Applications of Categories, 15, (2005) 264-277.

[12] R. Rosebrugh, N. Sabadini, R.F.C. Walters, *Calculating colimits compositionally*, Montanari Festschrift, LNCS 5065,(2008) 581592

[13] L. de Francesco Albasini, N. Sabadini and R.F.C. Walters, *An algebra of automata that includes both classical and quantum entities*, Electr. Notes Theor. Comput. Sc. 270 (2011) 263-270.

[14] R. Rosebrugh, N. Sabadini, R.F.C. Walters, *Tangled Circuits*, arXiv:1110.0715 (2011).

[15] Jennifer Lantair, Pawel Sobocinski, *WiCcA: LTS generation tool for wire calculus*, Proceeding CALCO'11 Proceedings of the 4th international conference on Algebra and coalgebra in computer science, (2011) 407-412.

[16] Roberto Bruni, Hernan Melgratti, Ugo Montanari and Pawel Sobocinski, *Connector algebras for C/E and P/T nets interactions*, preprint, (2012).