TIMING VIA PINWHEEL DOUBLE CATEGORIES

ELENA DI LAVORE AND MARIO ROMÁN

ABSTRACT. We introduce string diagrams for timed process theories—symmetric monoidal categories graded on a cartesian duoidal poset—built upon the string diagrams of pinwheel double categories.

1. TIMING PROCESS THEORIES

Process theories have an algebra in terms of symmetric monoidal categories; and symmetric monoidal categories have an internal language in terms of string diagrams, which we will use during this text. String diagrams are particularly well-suited to process description (Figure 1), and they appear used, in different levels of formality, across science and engineering.

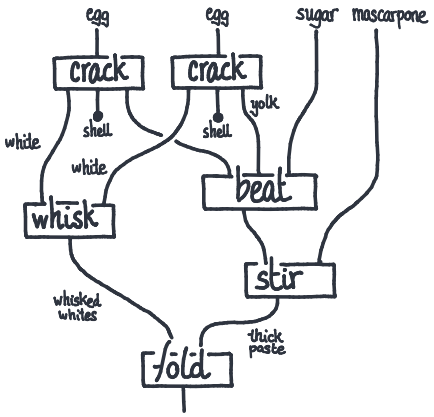

FIGURE 1. Process description of a preparation of *crema di mascarpone*, adapted from Sobocinski

In symmetric monoidal categories, executing two independent processes in parallel is the same as executing any of the two and then executing the other: in fact, string diagrams cannot distinguish between the two. This is a feature, for it reflects that the result must be the same in both cases. However, we may be interested in aspects of the process beyond its result, and these may not be preserved by naive string diagrammatic reasoning: for instance, parallelism certainly matters if we are computing how much time a process takes.

This text proposes a string diagrammatic calculus for timing processes in monoidal categories. It uses duoidal grading and string diagrams for pinwheel double categories. While the underlying category theory needs some care, the resulting string

Date: April 16, 2025.
Key words and phrases. Category theory, categorical semantics.

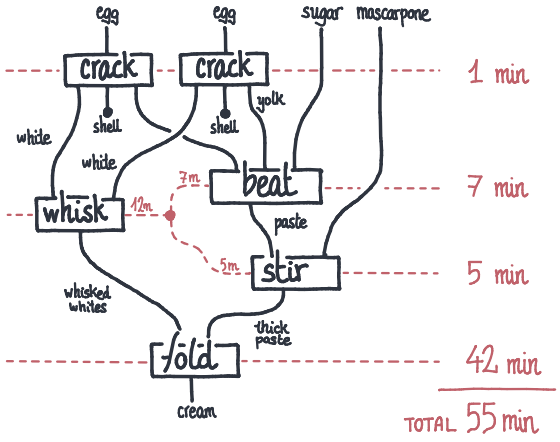

FIGURE 2. Timing of the "crema di mascarpone" process.

diagrams intuitively capture timing: in the vertical direction, we see how resources get transformed, in the horizontal direction, we see how time flows. Let us propose the following slogan.

Process in sequence share *resources*; process in parallel share *time*.

1.1. **Related work.** String diagrams for double categories [Daw95, Mye16] seem to still be missing a proof of initiality; only the single-object case has been partially developed [Nes21, Nes23]. Instead, we take seriously the algebras of Delpeuch's monad on double signatures [Del20] as a definition of pinwheel double categories. Duoidally graded string diagrams are missing, and duoidal string diagrams are only developed for the normal case [Rom24].

2. DUOIDAL TIMING

This section details the semantics of timed string diagrams in terms of duoidally-graded symmetric monoidal categories. The duoids we are interested in are commutative and normal. They represent time durations, and they have two monoid operations: an addition and a maximum. The main idea is that these operations must satisfy the equation

$$\mathsf{max}(a + c; b + d) \leq \mathsf{max}(a; b) + \mathsf{max}(c; d).$$

Interacting maximum and addition are a common motivation behind the development of normal duoidal categories [GF16, SS22].

Definition 1 (Duoid)**.** A *duoid*, $(A, \leq, \oplus, 0, \uparrow, \bot)$, is a poset $(A, \leq)$ endowed with two monoid structures, $(A, \oplus, 0)$ and $(A, \uparrow, \bot)$, that are monotone—meaning that $a \leq a'$ and $b \leq b'$ imply $a \oplus a' \leq b \oplus b'$ and $a \uparrow a' \leq b \uparrow b'$—and such that, additionally, the first laxly distributes over the second,

$$(a \oplus b) \uparrow (c \oplus d) \leq (a \uparrow c) \oplus (b \uparrow d),$$

with $0 \uparrow 0 \leq 0$, and $\bot \leq \bot \oplus \bot$, and $\bot \leq 0$. The duoid is *normal* when $\bot = 0$. It is *commutative* when $a \oplus b = b \oplus a$ and $a \uparrow b = b \uparrow a$.

Example 2 (Natural numbers). The natural numbers, with addition and the maximum, form a normal commutative duoid, $(\mathbb{N}, +, 0, \mathsf{max}, 0)$. In the same sense that every monoidal category forms a duoidal category with the cartesian product when it exists, every monoidal poset forms a duoid with maximum when it exists. We are mostly concerned with these maximum-duoids.

2.1. **Duoidally-graded symmetric monoidal categories.** In a duoidally-graded symmetric monoidal category, every morphism $f \in \mathbb{C}(X;Y)$ has a grade, $a \in A$—we think of it as the time it takes to execute this morphism—and we write $\mathbb{C}_a(X;Y)$ for the set of morphisms with grade $a \in A$. A morphism can be regraded, forced to take more time: for each $f \in \mathbb{C}_a(X;Y)$ and $a \leq b$, there exists $w_b(f) \in \mathbb{C}_b(X;Y)$.

Definition 3 (Duoidally-graded symmetric monoidal category)**.** A duoidally-graded (strict and) symmetric monoidal category[1] on a normal commutative duoid $(A, \leq, \oplus, 0, \uparrow, \bot)$, consists of a monoid of objects, $(\mathbb{C}_{obj}, I, \otimes)$, and, for each element of the duoid, $a \in A$, and each two objects $X, Y \in \mathbb{C}_{obj}$, a set of morphisms $\mathbb{C}_a(X;Y)$.

There must exist an identity morphism, $\mathrm{id}_X \in \mathbb{C}_0(X;X)$, for each object $X \in \mathbb{C}$; but, in particular, there is a special identity morphism for the monoidal unit, $\mathrm{i}_I \in \mathbb{C}_\bot(I;I)$. For each pair of compatible morphisms, $f \in \mathbb{C}_a(X;Y)$ and $g \in \mathbb{C}_b(Y;Z)$, there must exist a composite $(f \fatsemi g) \in \mathbb{C}_{a \oplus b}(X;Z)$. For each pair of morphisms, $f \in \mathbb{C}_a(X;Y)$ and $f' \in \mathbb{C}_{a'}(X';Y')$, there must exist a parallel composite $(f \otimes f') \in \mathbb{C}_{a \uparrow a'}(X \otimes X'; Y \otimes Y')$. For each duoid inequality, $a \leq b$, and each morphism $f \in \mathbb{C}_a(X;Y)$, there must exist a weakened morphism, $w_b(f) \in \mathbb{C}_b(X;Y)$.

A duoidally-graded symmetric monoidal category must satisfy all axioms of symmetric monoidal categories that are well-typed and, additionally, the interchange law (which needs weakening):

$$w_{(a\uparrow c)\oplus(b\uparrow d)}((f \fatsemi h) \otimes (g \fatsemi k)) = (f \otimes g) \fatsemi (h \otimes k), \quad \text{and} \quad w_0(\mathrm{i}) = \mathrm{id}_I.$$

Remark 4. We will be mostly concerned with symmetric monoidal categories graded over a normal and commutative maximum-duoid. This means that the interchange law does not hold automatically, $(f \fatsemi h) \otimes (g \fatsemi k)$ takes equal or less time than $(f \otimes g) \fatsemi (h \otimes k)$; but we can equate both by waiting that difference.

3. Double categories

3.1. **Double categories.** The usual definition of a double category is that of a category internal to the finitely complete category of small categories.

Definition 5 (Double categories)**.** A (strict) *double category* $\mathbb{D}$ consists of (*1*) a set of objects, $\mathbb{D}_{obj}$; (*2*) sets of arrows (or, *horizontal arrows*) $\mathbb{D}_h(X;Y)$ for each pair of objects, $X, Y \in \mathbb{D}_{obj}$; (*3*) sets of arrows (or, *vertical arrows*) $\mathbb{D}_v(X;Y)$ for each pair of objects, $X, Y \in \mathbb{D}_{obj}$; and (*4*) sets of cells, $\mathbb{D}(U; H; K; V)$, for each compatible quadruple of two vertical arrows $U \in \mathbb{D}_v(X;Y)$ and $V \in \mathbb{D}_v(Z;W)$, and two horizontal arrows $H \in \mathbb{D}_h(X;Z)$ and $K \in \mathbb{D}_h(Y;W)$.

Double categories have operations for vertical and horizontal composition of compatible 2-cells along their horizontal and vertical boundaries. These compositions are associative and unital and moreover respect the interchange law.

[1] Strict and symmetric monoidal categories are also known as a *permutative categories*, or *symmetric strict monoidal categories*. The important bit is to not confuse them with commutative monoidal categories, where the braiding morphism is the identity.

Definition 6 (Double signature)**.** A *double signature*, $\Sigma = (\mathcal{O}, \mathcal{H}, \mathcal{V}, \mathcal{D})$, consists of (*1*) a set of objects, $\mathcal{O}$; (*2*) two graphs on these objects, given by sets $\mathcal{H}(X;Y)$ and $\mathcal{V}(X;Y)$ for each pair of objects, $X, Y \in \mathcal{O}$; (*3*) sets of generators,

$$\mathcal{D}_{X,Y,Z,W}(u_1, ..., u_n; h_1, ..., h_p; k_1, ..., k_q; v_1, ..., v_m),$$

for each four objects $X, Y, Z, W \in \mathcal{O}$, and each quadruple of composable paths given by $\boldsymbol{u} = u_1, ..., u_n \in \mathsf{path}(\mathcal{V})(X;Y)$ and $\boldsymbol{h} = h_1, ..., h_p \in \mathsf{path}(\mathcal{H})(Y;W)$, with $\boldsymbol{v} = v_1, ..., v_m \in \mathsf{path}(Z;W)$, and $\boldsymbol{k} = k_1, ..., k_q \in \mathsf{path}(\mathcal{H})(X;Z)$.

Proposition 7 (Double signature morphism)**.** *A* double signature morphism, $\alpha \colon \Sigma \to \Sigma'$*, between two double signatures,* $\Sigma = (\mathcal{O}, \mathcal{H}, \mathcal{V}, \mathcal{D})$ *and* $\Sigma' = (\mathcal{O}', \mathcal{H}', \mathcal{V}', \mathcal{D}')$*, consists of*

1. *a function on objects,* $\alpha_{obj} \colon \mathcal{O} \to \mathcal{O}'$*;*
2. *two graph homomorphisms, with that same underlying function on objects,* $\alpha_H \colon \mathcal{H} \to \mathcal{H}'$ *and* $\alpha_V \colon \mathcal{V} \to \mathcal{V}'$*, which extend to functors,* α_H^* *and* α_V^**; and*
3. *a function on cells, preserving the boundary graphs,*

$$\alpha \colon \mathcal{D}_{X,Y,Z,W}(\boldsymbol{u}; \boldsymbol{h}; \boldsymbol{k}; \boldsymbol{v}) \to \mathcal{D}'_{\alpha(X),\alpha(Y),\alpha(Z),\alpha(W)}(\alpha_V^*(\boldsymbol{u}); \alpha_H^*(\boldsymbol{h}); \alpha_H^*(\boldsymbol{k}); \alpha_V^*(\boldsymbol{v})).$$

Double signatures and double signature morphisms form a category, doubleSig*.*

Proposition 8 (Forgetful double signatures)**.** *The forgetful double signature of a double category,* $\mathbb{D}$*, is a double signature* $\Sigma = (\mathcal{O}, \mathcal{H}, \mathcal{V}, \mathcal{D})$ *given by*

1. *the objects of the double category,* $\mathcal{O} = \mathbb{D}_{obj}$*;*
2. *the horizontal cells of the double category,* $\mathcal{H}(X;Y) = \mathbb{D}_h(X;Y)$*;*
3. *the vertical cells of the double category,* $\mathcal{V}(X;Y) = \mathbb{D}_v(X;Y)$*;*
4. *the 2-cells of the double category, for each quadruple of composable paths given by* $\boldsymbol{u} = u_1, ..., u_n \in \mathsf{path}(\mathcal{V})(X;Y)$ *and* $\boldsymbol{h} = h_1, ..., h_p \in \mathsf{path}(\mathcal{H})(Y;W)$*, with* $\boldsymbol{v} = v_1, ..., v_m \in \mathsf{path}(Z;W)$*, and* $\boldsymbol{k} = k_1, ..., k_q \in \mathsf{path}(\mathcal{H})(X;Z)$*, we have* $\mathcal{D}(\boldsymbol{u}; \boldsymbol{h}; \boldsymbol{k}; \boldsymbol{v}) = \mathbb{D}(u_1 \otimes ... \otimes u_n; h_1 \otimes ... \otimes h_p; k_1 \otimes ... \otimes k_q; v_1 \otimes ... \otimes v_m)$*.*

The construction of the forgetful double signature induces a functor from double categories to double signatures with strict double functors $\mathsf{forget} \colon \mathsf{doubleCat} \to \mathsf{doubleSig}$*.*

3.2. **Double categories from timed polygraphs.** Polygraphs are signatures for monoidal categories: they comprise some generating objects and some basic processes from which all the others are built. If we declare how much time these building blocks take, we can deduce how much every composite process will take. This is the idea behind a timed polygraph: they are signatures for timed process theories.

Definition 9 (Timed polygraph)**.** A *timed polygraph* Ψ consists of a set of objects, $\Psi_o bj$, and for each two lists of objects, $X_1, ..., X_n \in \Psi_{obj}$ and $Y_1, ..., Y_m \in \Psi_{obj}$, and each natural number $t \in \mathbb{N}$, a set of generators, $\Psi_t(X_1, ..., X_n; Y_1, ..., Y_m)$.

Definition 10 (Timed polygraph morphism)**.** A *timed polygraph morphism*, $\alpha \colon \Psi \to \Xi$, consists of a function between the object sets, $\alpha_{obj} \colon \Psi_{obj} \to \Xi_{obj}$, and a function between generators that preserves time, inputs and outputs,

$$\alpha \colon \Psi_t(X_1, ..., X_n; Y_1, ..., Y_m) \to \Xi_t(\alpha(X_1), ..., \alpha(X_n); \alpha(Y_1), ..., \alpha(Y_m)).$$

Proposition 11 (Double signature from a timed polygraph)**.** *Every timed polygraph* Ψ *induces a double signature,* $\mathsf{draw}(\Psi) = (\mathcal{O}, \mathcal{H}, \mathcal{V}, \mathcal{D})$*, containing*

(1) *a single object,* $\mathcal{O} = \{*\}$*;*
(2) *a single horizontal generator,* $\mathcal{H}(*,*) = \{1\}$*;*
(3) *a vertical generator for each object,* $\mathcal{V}(*,*) = \Psi_{obj}$*;*
(4) *and three kinds of cells: a cell for each timed generator,*

$$\mathcal{D}(X_1, ..., X_n; t; t; Y_1, ..., Y_m) = \Psi_t(X_1, ..., X_n; Y_1, ..., Y_m),$$

and, additionally, "waiting" cell, $\mathcal{D}(X; 1; 1; X) = \Psi_1(X; X) + \{\text{wait}\}$*, and a "braiding" cell* $\mathcal{D}(X, Y; 0; 0; Y, X) = \Psi_0(X, Y; Y, X) + \{\sigma\}$ *(see Figure 3).*

This assignment extends to a functor draw: tPolygraph → doubleSig.

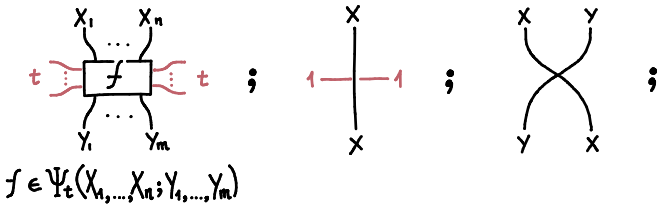

FIGURE 3. Double signature from a timed polygraph.

Proposition 12 (Double category from a timed polygraph)**.** *Every timed polygraph,* Ψ *induces a double category freely generated by its associated double signature,* draw(Ψ)*, and quotiented by the equations in Figure 4.*

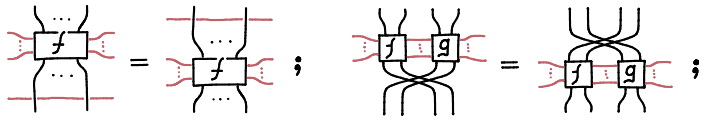

FIGURE 4. Equations for the double category from a timed polygraph.

3.3. **Towards pinwheel double categories.** String diagrams for double categories face a problem: not every plane arrangement of composable cells can be explained in terms of the algebra of double categories [Daw95]. Those that can be explained were described by Dawson as "neat tilings" and characterized as those that do not contain a full *pinwheel* (Figure 5): the only obstruction to composition in double categories are pinwheels.

Example 13. Consider a quintuple of timed processes: f and h take two units of time, while g, k, and a take a single unit. The resources of a and h depend on g, while the resouces of k depend on f and a. We could: *(1)* execute f and g in parallel; *(2)* when g finishes, execute a and h in parallel; *(3)* when both f and a finish, execute k; and *(4)* both h and k will finish at the same time, after 4 units of time. There is no physical limitation for this arrangement, but it is disallowed by the algebra of double categories (Figure 5, left).

However, this means double categories cannot be the algebraic structure that encodes all possible combinations of timing processes. The pinwheel represents a valid timing, and so we should account for it. Instead, we need more algebraic structure: we need double categories *with pinwheels.* Recent work by Delpeuch will open this route [Del20].

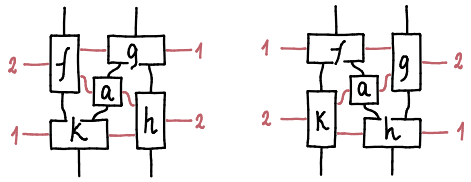

FIGURE 5. Two valid timings that form pinwheels.

4. Double categories with pinwheels

All uses the double categories that do not rely on the exclusion of pinwheels can be recasted for 2-categories, gaining the extra expressivity for pinwheels in the process [Del20]. The main idea is that any 2-cell in a double category can be "tilted" and interpreted as a 2-cell in a corresponding 2-category that has, as generating objects, both the vertical and horizontal cells of the double category (Figure 6).

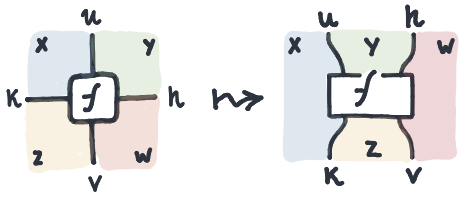

FIGURE 6. Double cells can be tilted to 2-cells.

In formal terms, each double signature can be translated into a 2-graph that generates the free double category with pinwheels.

Definition 14 (Tilted 2-graph)**.** The *tilted 2-graph* of a double signature $\Sigma = (\mathcal{O}, \mathcal{H}, \mathcal{V}, \mathcal{D})$ is the 2-graph, $\mathsf{tilt}(\Sigma)$, defined as having

- the same 0-cells, $\mathsf{tilt}(\Sigma)_{obj} = \mathcal{O}$;
- both vertical and horizontal 1-cells, $\mathsf{tilt}(\Sigma)(X;Y) = \mathcal{V}(X;Y) + \mathcal{H}(X;Y)$;
- the same 2-cells between $X \in \mathcal{O}$ and $W \in \mathcal{O}$ when paths are composed precisely of $\boldsymbol{u} = u_1, ...u_n \in \mathsf{path}(\mathcal{V})(X;Y)$ and $\boldsymbol{h} = h_1, ..., h_p \in \mathsf{path}(\mathcal{H})(Y;W)$, with $\boldsymbol{v} = v_1, ..., v_m \in \mathsf{path}(\mathcal{V})(Z;W)$ and $\boldsymbol{k} = k_1, ..., k_q \in \mathsf{path}(\mathcal{H})(X;Z)$, for any two $Y, Z \in \mathcal{O}$; that is,

  $$\mathsf{tilt}(\Sigma)_{X,W}(u_1, ..., u_n, h_1, ..., h_p; k_1, ..., k_q, v_1, ..., v_n) = \mathcal{D}_{X,Y,Z,W}(\boldsymbol{u}; \boldsymbol{h}; \boldsymbol{k}; \boldsymbol{v});$$

  and empty otherwise, with $\mathsf{tilt}(\Sigma)(x_1, ..., x_n; y_1, ..., y_m) = \varnothing$ for any other two given paths $\boldsymbol{x} = x_1, ..., x_n \in \mathsf{path}(\mathsf{tilt}(\Sigma))(X;W)$ and $\boldsymbol{y} = y_1, ..., y_m \in \mathsf{path}(\mathsf{tilt}(\Sigma))(X;W)$.

From this 2-graph, we can construct its free 2-category, $\mathsf{string}(\mathsf{tilt}(\Sigma))$. String diagrams for this category contain the cells of the pinwheel double category we want to construct, but they also contain many diagrams that cannot be interpreted as double cells: we use those that do to construct the tentative *free pinwheel double category* over a double signature.

Definition 15 (Pinwheel double category)**.** Any double signature, $\Sigma = (\mathcal{O}, \mathcal{H}, \mathcal{V}, \mathcal{D})$, generates a strict double category, $\mathsf{pinwheel}(\Sigma)$, having

(1) objects from its signature $\mathsf{pinwheel}(\Sigma)_{obj} = \mathcal{O}$;

(2) horizontal paths, composing as such, $\mathsf{pinwheel}(\Sigma)_h = \mathsf{path}(\mathcal{H})$;
(3) vertical paths, composing as such, $\mathsf{pinwheel}(\Sigma)_v = \mathsf{path}(\mathcal{V})$;
(4) cells the string diagrams constructed from the tilted 2-graph; that is, for $\boldsymbol{u} = u_1, ...u_n \in \mathsf{path}(\mathcal{V})(X;Y)$ and $\boldsymbol{h} = h_1, ..., h_p \in \mathsf{path}(\mathcal{H})(Y;W)$, with $\boldsymbol{v} = v_1, ..., v_m \in \mathsf{path}(\mathcal{V})(Z;W)$ and $\boldsymbol{k} = k_1, ..., k_q \in \mathsf{path}(\mathcal{H})(X;Z)$, we have that the set of cells $\mathsf{pinwheel}(\Sigma)(\boldsymbol{u};\boldsymbol{h};\boldsymbol{k};\boldsymbol{v})$ is defined by

$$\mathsf{string}(\mathsf{tilt}(\Sigma))(u_1 \otimes ... \otimes u_n \otimes h_1 \otimes ... \otimes h_p; k_1 \otimes ... \otimes k_q \otimes v_1 \otimes ... \otimes v_m).$$

Horizontal and vertical composition for this double category are given by compositions of 2-categorical string diagrams, following Figure 7.

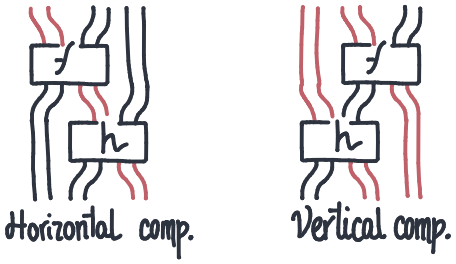

FIGURE 7. Horizontal and vertical compositions in the tilted 2-category.

Proposition 16 (Pinwheel monad)**.** *Pinwheel forms a monad*

$$\mathsf{Pinwheel} = (\mathsf{forget} \fatsemi \mathsf{pinwheel}) : \mathsf{doubleSig} \to \mathsf{doubleSig}.$$

Definition 17 (Pinwheel double category)**.** Pinwheel double categories are the algebras of the pinwheel monad on double signatures.

5. CONCLUSIONS

We have seen how to construct timed string diagrams, like that in Figure 2, using pinwheel double categories. In this example, the timed polygraph is given by the following generators.

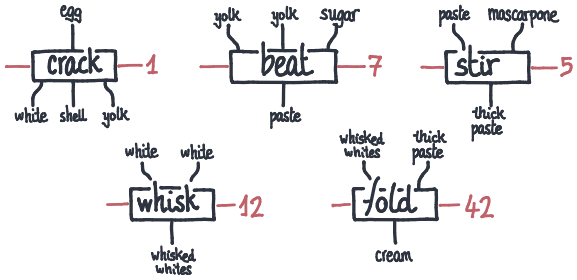

FIGURE 8. Timed polygraph for crema di mascarpone.

5.1. **Further work.** More details for the constructions of this paper may be provided in later versions. Monoidal width, developed by Di Lavore and Sobocinski, may arise from a slight generalization of the construction presented here [DS23]. It may be possible to connect this way of timing processes to time-graded coalgebras of monads, which the authors have recently proposed for continuous dynamical systems [DR25].

Funding. Mario Román was supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0038. Elena Di Lavore and Mario Román were supported by the Advanced Research + Invention Agency (ARIA) Safeguarded AI Programme.

References

[Daw95] Robert Dawson. A forbidden-suborder characterization of binarily-composable diagrams in double categories. *Theory and Applications of Categories*, 1(7):146–155, 1995.

[Del20] Antonin Delpeuch. The word problem for double categories. *Theory and Applications of Categories*, 35(1):1–18, 2020.

[DR25] Elena Di Lavore and Mario Román. Graded coalgebras of monads for continuous dynamics. https://mroman42.github.io/notes/papers/graded-coalgebras-of-monads-for-continuous-dynamics.pdf, 2025.

[DS23] Elena Di Lavore and Pawel Sobocinski. Monoidal width. *Log. Methods Comput. Sci.*, 19(3), 2023.

[GF16] Richard Garner and Ignacio López Franco. Commutativity. *Journal of Pure and Applied Algebra*, 220(5):1707–1751, 2016.

[Mye16] David Jaz Myers. String diagrams for double categories and equipments, 2016.

[Nes21] Chad Nester. The structure of concurrent process histories. In Ferruccio Damiani and Ornela Dardha, editors, *Coordination Models and Languages - 23rd IFIP WG 6.1 International Conference, COORDINATION 2021, Held as Part of the 16th International Federated Conference on Distributed Computing Techniques, DisCoTec 2021, Valletta, Malta, June 14-18, 2021, Proceedings*, volume 12717 of *Lecture Notes in Computer Science*, pages 209–224. Springer, 2021.

[Nes23] Chad Nester. Concurrent Process Histories and Resource Transducers. *Logical Methods in Computer Science*, Volume 19, Issue 1, January 2023.

[Rom24] Mario Román. String diagrams for physical duoidal categories. *arXiv preprint arXiv:2406.19816*, 2024.

[SS22] Brandon T. Shapiro and David I. Spivak. Duoidal structures for compositional dependence. *arXiv preprint arXiv:2210.01962*, 2022.

Appendix A. String Diagrams of Bicategories

Definition 18. A strict *2-category* $\mathbb{B}$ consists of a collection of *objects*, or 0-cells, $\mathbb{B}_{obj}$, and a category of *morphisms* or 1-cells between any two objects, $\mathbb{B}(A;B)$. A strict *2-category* is endowed with operations for the parallel composition of 1-cells,

$$(\fatsemi)\colon \mathbb{B}(A;B) \times \mathbb{B}(B;C) \to \mathbb{B}(A;C),$$
$$(I_A)\colon \mathbb{B}(A;A),$$

that are associative and unital both on objects and morphisms, meaning that $(X \fatsemi Y) \fatsemi Z = X \fatsemi (Y \fatsemi Z)$, and $I_A \fatsemi X = X = X \fatsemi I_B$. Bicategories must satisfy the following axioms, making parallel composition a functor:

(1) parallel composition is unital, $f \fatsemi \mathrm{id} = f$, and $\mathrm{id} \fatsemi f = f$;
(2) parallel composition is associative, $f \fatsemi (g \fatsemi h) = (f \fatsemi g) \fatsemi h$;
(3) compositions are unital, $\mathrm{id} \fatsemi \mathrm{id} = \mathrm{id}$;
(4) compositions interchange, $(f \fatsemi g) \fatsemi (f' \fatsemi g') = (f \fatsemi f') \fatsemi (g \fatsemi g')$.

Remark 19. A single-object strict 2-category is exactly a strict monoidal category.

A.1. **String diagrams of 2-categories.** Let us assume any construction of bicategorical string diagrams. There are multiple options when constructing bicategorical string diagrams: we could propose a combinatorial description or a topological one, we could decide that 0-cells appear as part of the structure, or simply as the property of being well-typed. These constructions are all left adjoints to the same forgetful functor from 2-categories to *2-graphs*.

Definition 20 (2-graph)**.** A *2-graph*, or *bigraph*, $\mathcal{B}$ is given by a set of objects, $\mathcal{B}_{obj}$; a set of arrows between any two objects, $\mathcal{B}(A;B)$; and a set of 2-arrows between any two paths of arrows, $\mathcal{B}(X_0,..,X_n;Y_0,..,Y_m)$.

Definition 21 (2-graph morphism)**.** A *2-graph morphism*, $f\colon \mathcal{A} \to \mathcal{B}$, is a function between their object sets, $f_o\colon \mathcal{A}_{obj} \to \mathcal{B}_{obj}$; a family of functions between their corresponding arrow sets, $f\colon \mathcal{A}(A;B) \to \mathcal{B}(f(A),f(B))$; and a family of functions between their corresponding 2-arrow sets,

$$f\colon \mathcal{A}(X_0,..,X_n;Y_0,..,Y_m) \to \mathcal{B}(f(X_0),..,f(X_n);f(Y_0),..,f(Y_m)).$$

2-Graphs with 2-graph morphisms form a category, 2-graph.

Definition 22 (Forgetful 2-graph)**.** The forgetful 2-graph, $\mathsf{forget}(\mathbb{A})$, of a 2-category, $\mathbb{A}$, has the same set of objects as the 2-category, $\mathsf{forget}(\mathbb{A})_{obj} = \mathbb{A}_{obj}$; and morphism sets given by all the morphisms of the 2-category,

$$\mathsf{forget}(\mathbb{A})(X_1,...,X_n;Y_1,...,Y_n) = \mathbb{A}(X_1 \otimes ... \otimes X_n; Y_1 \otimes ... \otimes Y_n).$$

Theorem 23 (Free bicategories)**.** *The functor* $\mathsf{forget}\colon \mathsf{2\text{-}cat} \to \mathsf{2\text{-}graph}$ *is a right adjoint.*

That is, there is a right adjoint from 2-categories to 2-graphs. A left adjoint can be constructed by bicategorical string diagrams over the bigraph; let us write $\mathsf{string}\colon \mathsf{2\text{-}graph} \to \mathsf{2\text{-}cat}$ for this left adjoint. However, let us still not assume any particular construction for this left adjoint.