Presentations of Premonoidal Categories by Devices

Matt Earnshaw Chad Nester Mario Román

Tallinn University of Technology, Estonia

NWPT, Västerås

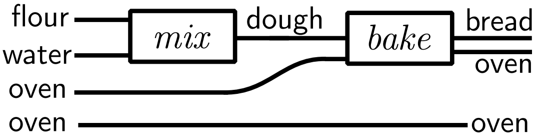
22nd November 2023

String diagrams for process theories

This talk is about a graphical syntax for *processes*, broadly construed.

Monoidal categories are an algebraic formalism for *resource-transforming processes*.

String diagrams are a *sound* and *complete* graphical syntax for monoidal categories.[a]

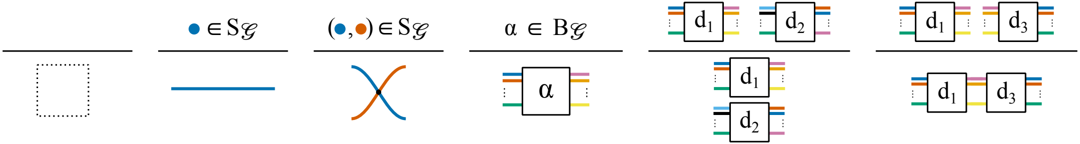


Examples: sets and partial maps with cartesian product, Hilbert spaces and bounded linear maps with tensor product.

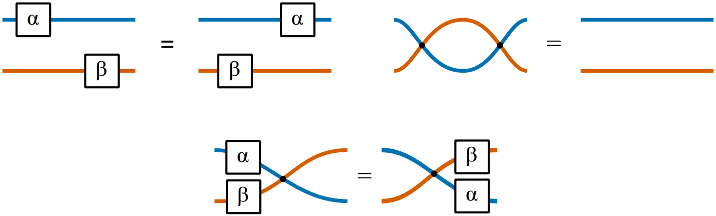
[a] Joyal & Street [JS91]

String diagrams for process theories

Completeness – the free symmetric monoidal category on a set of generating processes is given by string diagrams:



Soundness – equational reasoning is topological:



Premonoidal categories for effectful processes

Interchange is not always obeyed by processes in computer science:

Premonoidal categories[a] refine monoidal categories: interchange does not hold globally.

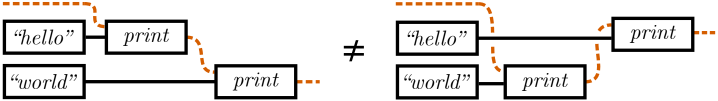
Key example: Kleisli categories of strong monads, or more generally strong promonads.

Interchange holds just when the monad is *commutative*.

[a]Power and Robinson [PR97].

String diagrams for premonoidal categories

Adding a runtime wire presents the free premonoidal category with specified *centre*.[a]



In practice, this global effect limits topological reasoning:

[a] Jeffrey, Román [Jef97, Rom23].

String diagrams with devices

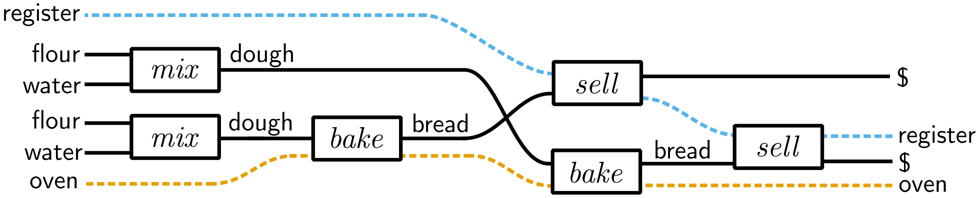
In practice, this global effect limits topological reasoning:

Introduce multiple *device* wires:

Now many natural equations are topological again.

String diagrams with devices

Premonoidal categories are an algebraic foundation for processes that may use both resources and *devices*.



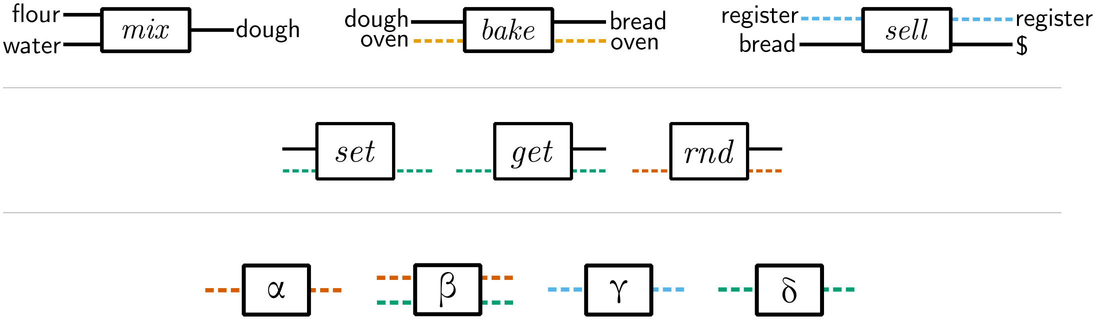
Devices are definite noun phrases: if we only have one oven, we cannot *bake* in parallel.

We introduce a convenient presentation for premonoidal categories based on this idea.

Device signature

Definition. A *device signature* is given by:

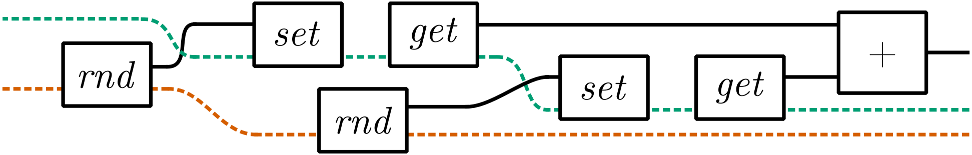
- sets R, P, D of *resources, processes* and *devices*,
- functions $s, t : P \to R^*$ assigning source and target *words* of resources,
- a function $d : P \to \mathscr{P}(D)$ specifying a set of devices used by each process.



Device presentations

A device presentation further specifies some equations between string diagrams:

Proposition. *Device presentations freely generate premonoidal categories.*



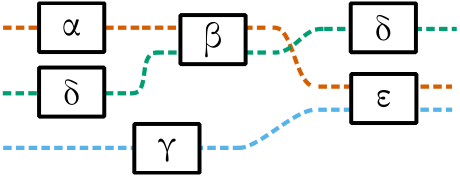
Example adapted from the *functional machine calculus.*[a]

[a]Barrett, Heijltjes & McCusker [Bar23, BHM]

Mazurkiewicz traces by devices

Mazurkiewicz traces [DR95] model the behaviour of concurrent machines.

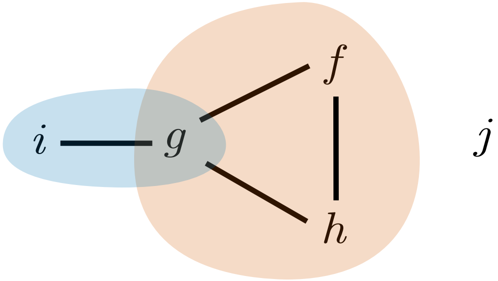
Traces generalize *words*, the behaviour of sequential machines, by allowing specified pairs of actions to commute.

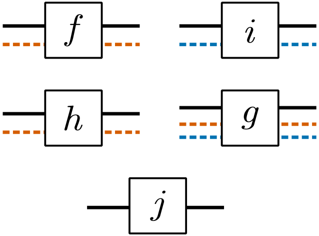


Proposition (E., Sobociński [ES]). *Mazurkiewicz traces arise as the morphisms of premonoidal categories generated by device signatures with no resource wires.*

These devices may be conceived of as shared memory locations.

The canonical device presentation





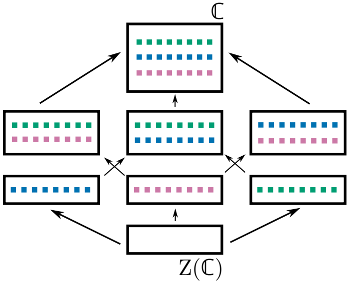
Proposition. *The interference graph of a premonoidal category $\mathbb{C}$ determines a device presentation of $\mathbb{C}$ that contains a device for each non-trivial maximal clique.*

Proposition. *The interference graph of a premonoidal category presented by devices recovers the device presentation.*

The device lattice

The *centralizer* of a set S of processes in a premonoidal category contains all processes interchanging with elements of S.

Proposition. *Centralizers are premonoidal subcategories.*



Proposition. *A premonoidal category $\mathbb{C}$ admits a lattice of premonoidal subcategories, each corresponding to a subset of the devices of $\mathbb{C}$, bounded below by its centre $Z(\mathbb{C})$, and above by $\mathbb{C}$.*

Future directions: Combining effects

Combining categories of effectful processes:

- coproducts and tensor products of algebraic theories [HPP06]
- distributive laws of monads [Bec69]

Given two presentations, we have various ways to combine them.

How do these relate to known constructions for combining effects?

References

[Bar23] Chris Barrett, *On the simply-typed functional machine calculus: Categorical semantics and strong normalisation*, PhD thesis, University of Bath, 2023.

[Bec69] Jon Beck, *Distributive laws*, Seminar on Triples and Categorical Homology Theory (Berlin, Heidelberg) (B. Eckmann, ed.), Springer Berlin Heidelberg, 1969, pp. 119–140.

[BHM] Chris Barrett, Willem Heijltjes, and Guy McCusker, *The Functional Machine Calculus II: Semantics*, 31st EACSL Annual Conference on Computer Science Logic (CSL 2023), pp. 10:1–10:18.

[DR95] V Diekert and G Rozenberg, *The book of traces*, World Scientific, 1995.

[ES] Matthew Earnshaw and Paweł Sobociński, *String Diagrammatic Trace Theory*, MFCS 2023, pp. 43:1–43:15.

[HPP06] Martin Hyland, Gordon Plotkin, and John Power, *Combining effects: Sum and tensor*, Theoretical Computer Science **357** (2006), no. 1, 70–99, Clifford Lectures and the Mathematical Foundations of Programming Semantics.

[Jef97] Alan Jeffrey, *Premonoidal categories and a graphical view of programs*, Preprint (1997).

[JS91] André Joyal and Ross Street, *The geometry of tensor calculus, I*, Advances in Mathematics **88** (1991), no. 1, 55–112.

[PR97] John Power and Edmund Robinson, *Premonoidal categories and notions of computation*, Math. Struct. Comput. Sci. **7** (1997), no. 5, 453–468.

[Rom23] Mario Román, *Promonads and string diagrams for effectful categories*, Proceedings Fifth International Conference on Applied Category Theory (Jade Master and Martha Lewis, eds.), 2023.