Resourceful traces for commuting processes

Matt Earnshaw
University of Tartu, Estonia

joint work with Chad Nester & Mario Román

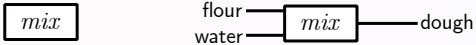
CSL, Paris
26th February 2026

➙ This talk is about a notion of *presentation* for categories of *effectful processes*.

In a program, the order of PRINT statements matters, but the order of assignments doesn't necessarily; or, I may have only one copy of a key that opens multiple doors, so they must be opened in a definite order: we cannot OPEN concurrently.

An established semantics of effectful processes is given by *monads* and their *Kleisli categories*, which often have *premonoidal* structure, modelling such phenomena.

Our work provides a new perspective on these gadgets:
premonoidal categories are trace monoids with types.



This perspective also facilitates the construction of natural combinations of process theories, such as their *commuting tensor product*.

E, Nester, and Román, 2025 – *Resourceful traces for commuting processes*

E., 2025 – *Languages of string diagrams*

Trace monoids

Words denote the behaviour of sequential machines with atomic actions.

Mazurkiewicz traces are words in which **specified pairs** of actions can commute.

A *dependency relation* over Σ is a reflexive, symmetric relation $D \subseteq \Sigma \times \Sigma$.

pairs of actions which may *interfere* or *depend* on each other, i.e. cannot occur concurrently

Traces are elements of the monoid $\langle \Sigma \mid ab = ba,\ \forall (a, b) \notin D \rangle$,

Thus traces are sequences of actions *up to commutation of independent actions*.

$$[wabv] = [wbav]$$

Traces provide a simple non-interleaving model of the behaviour of systems with concurrent atomic actions.

Diekert and Rozenberg, 1995 – *The Book of Traces*

Distributions

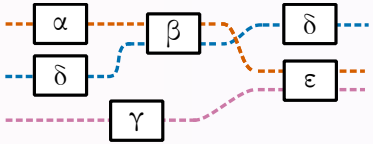
Another construction of traces starts from a *distribution* of the set of actions.

A *distribution* of Σ (across a set of "locations" or "devices") is given by,

a finite set of "devices" $\mathbb{D}$, and a function $\mathrm{dev} : \Sigma \to \mathscr{P}(\mathbb{D})$.

If we represent a distribution as follows,

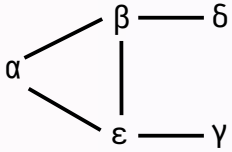
then traces are exactly the *string diagrams* we can build, from these generators (allowing each device exactly once in the boundaries)



E and Sobociński, 2023 – *String diagrammatic trace theory*

Dependence and devices

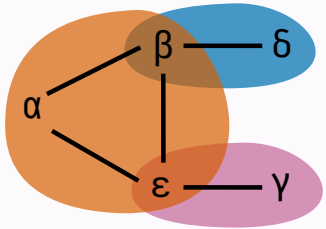
We can go from a dependence relation D to the corresponding distribution taking *maximal cliques* in the *undirected graph* of the relation D



For every (α, β) in D, there is an edge $\alpha - \beta$.

Dependence and devices

We can go from a dependence relation D to the corresponding distribution taking *maximal cliques* in the *undirected graph* of the relation D



Conversely, from a distribution, define $(\alpha, \beta) \in D$ just when $\mathrm{dev}(\alpha) \cap \mathrm{dev}(\beta) \neq \varnothing$.

Diekert and Rozenberg, 1995 – *The Book of Traces*

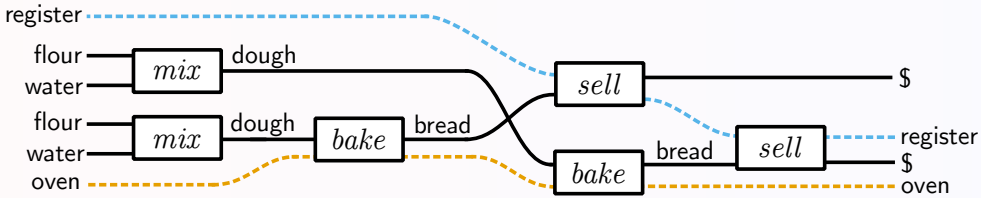
Beyond atomic actions

> *"actions are state transformations of some resources of a system*
> *... actions are independent if they act on disjoint sets of* **resources***"*

Mazurkiewicz, 1977 – *Concurrent program schemes and their interpretations*

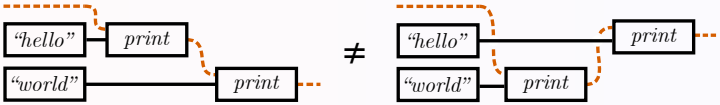
We argue: two notions of resource are conflated here,

- shared resources, definite noun phrases: "*the* database", "*the* printer" ← **devices**
- input/output types, indefinite noun phrases: "*a* query", "*a* string" ← resources



String diagrams for premonoidal categories

String diagrams with a distinguished string are a sound and complete syntax for premonoidal categories.



Jeffrey, 1997 – *Premonoidal categories and a graphical view of programs*

Román and Sobociński, 2025 – *String diagrams for premonoidal categories*

This *global* device prevents interchange of *all* effectful processes:

From a global device to multiple devices

In practice, this global device is too restrictive:

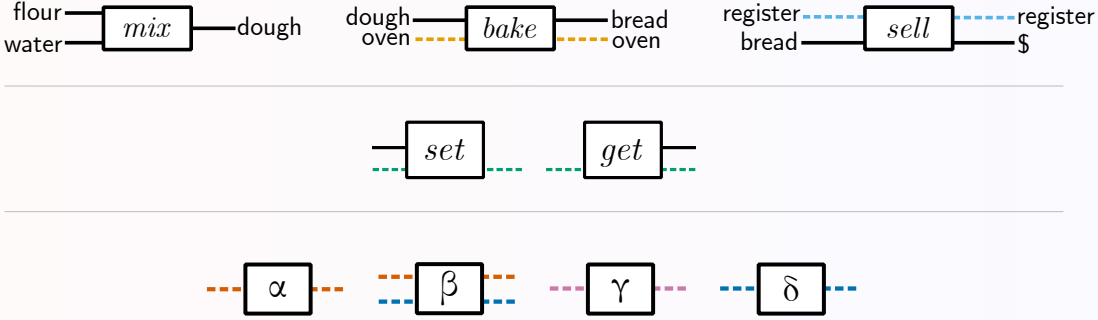
By introducing multiple strings corresponding to *devices*,

equations that we would expect, hold automatically.

Device signature

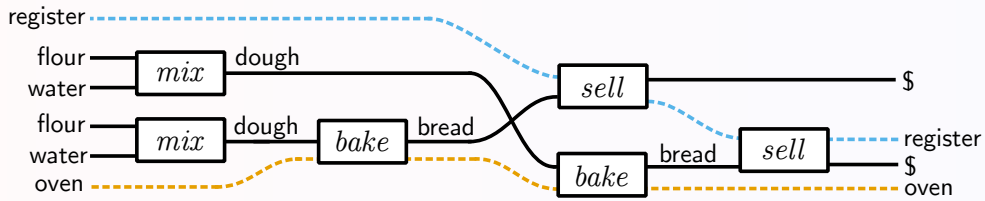
A *device signature* is given by:

- sets $R, P, \mathcal{D}$ of *"resources"*, *"generating processes"* and *"devices"*,
- a monoidal signature $R^* \xleftarrow{s} P \xrightarrow{t} R^*$,
- a function dev : $P \to \mathscr{P}(\mathcal{D})$ specifying the set of devices used by each generator



Device signatures generate premonoidal categories

Morphisms are identities, generating processes in resource context, and compositions.



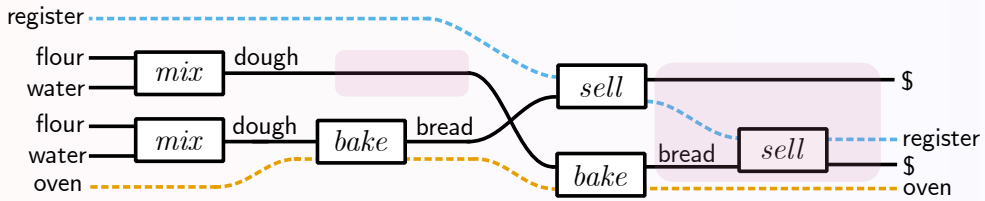
We ask that generating processes in context interchange when they have disjoint sets of devices.

Premonoidal structure is defined recursively on structure of morphisms.

➙ Theorem: this constructs the free premonoidal category on a device signature.

Device signatures generate premonoidal categories

Morphisms are identities, generating processes in resource context, and compositions.



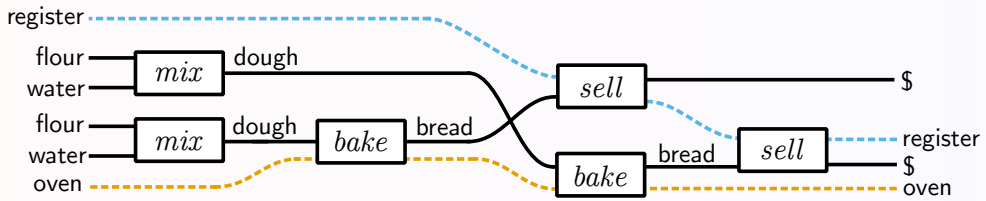
We ask that generating processes in context interchange when they have disjoint sets of devices.

Premonoidal structure is defined recursively on structure of morphisms.

➙ Theorem: this constructs the free premonoidal category on a device signature.

Device signatures generate premonoidal categories

Morphisms are identities, generating processes in resource context, and compositions.



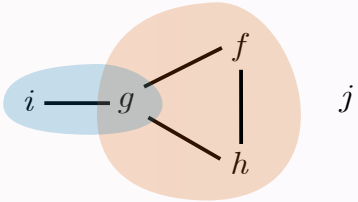
We ask that generating processes in context interchange when they have disjoint sets of devices.

Premonoidal structure is defined recursively on structure of morphisms.

➙ Theorem: this constructs the free premonoidal category on a device signature.

Dependence graph of a premonoidal category

We can extract an *underlying device signature* from any premonoidal category.



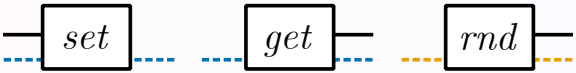
↬ This extends to a right adjoint to the construction of the free premonoidal category on a device signature.

Commuting tensor product of premonoidal categories

Let $\mathbb{A}$ and $\mathbb{B}$ be premonoidal categories. Their *commuting tensor product* $\mathbb{A} \odot \mathbb{B}$ is the *initial cospan* $\mathbb{A} \xrightarrow{F} \mathbb{A} \odot \mathbb{B} \xleftarrow{G} \mathbb{A}$ for which $F(f)$ and $G(g)$ commute $\forall$ f in $\mathbb{A}$, g in $\mathbb{B}$.

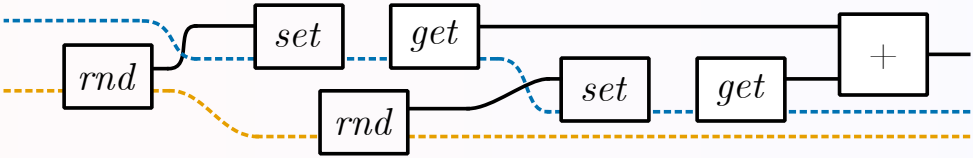
Garner and López Franco, 2016 – *Commutativity*

Given device signatures G and H with a common set of resources, we construct a signature $G \odot_s H$ having the same resources, but disjoint union of devices $\mathcal{D}_G + D_H$,



➙ Theorem: $\mathscr{F}(G \odot_s H) \cong \mathscr{F}(G) \odot \mathscr{F}(H)$

Commuting tensor product of premonoidal categories



Example adapted from Barrett, Heijltjes, and McCusker, 2023 – *The Functional Machine Calculus*.

Devices as grades (work in progress)

We use devices to construct a premonoidal category, but the premonoidal category does not "remember" these devices.

We can capture this extra structure via a grading by the partial monoid $(\mathscr{P}(\mathcal{D}), \oplus, \varnothing)$

$$S \oplus T := \begin{cases} S \cup T & \text{if } S \cap T = \varnothing \\ \uparrow & \text{otherwise.} \end{cases}$$

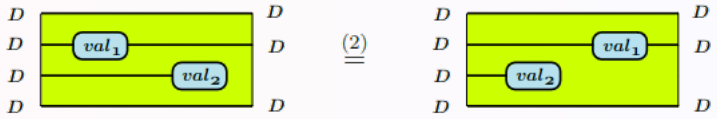
In a $(\mathscr{P}(\mathcal{D}), \oplus, \varnothing)$-graded monoidal category $\mathbb{C}$, we have hom-sets $\mathbb{C}_S(A; B)$ for every $S \in \mathscr{P}(\mathcal{D})$ and monoidal products

$$\otimes_{S,T} : \mathbb{C}_S(X; Y) \times \mathbb{C}_T(X'; Y') \to \mathbb{C}_{S \oplus T}(X \otimes X'; Y \otimes Y')$$

$\oplus$ partial $\Rightarrow$ only exists for morphisms with disjoint sets of devices!

Directions and questions

- *Device categories* in terms of grading by a powerset partial commutative monoid
- Formalizing our construction in terms of string diagrams
- Presentations with equations
- Explicit link to the tensor product of algebraic theories
- Explicit link to the literature on algebraic effects and handlers?
- Relation to Mellies, 2014 – *String diagrams for local state*?



References

- Barrett, Chris, Willem Heijltjes, and Guy McCusker (2023). "The Functional Machine Calculus II: Semantics". In: *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*. Ed. by Bartek Klin and Elaine Pimentel. Vol. 252. ISBN: 978-3-95977-264-8. DOI: 10.4230/LIPIcs.CSL.2023.10.
- Diekert, V and G Rozenberg (1995). *The Book of Traces*. World Scientific. DOI: 10.1142/2563.
- E, Chad Nester, and Mario Román (2025). *Resourceful Traces for Commuting Processes*. arXiv: 2507.18246 [cs.LO]. URL: https://arxiv.org/abs/2507.18246.
- E and Pawel Sobociński (2023). "String Diagrammatic Trace Theory". In: *MFCS*. DOI: 10.4230/LIPIcs.MFCS.2023.43.
- E. (2025). "Languages of String Diagrams". PhD thesis. DOI: 10.23658/TALTECH.2/2025.
- Garner, Richard and Ignacio López Franco (2016). "Commutativity". In: *Journal of Pure and Applied Algebra* 220.5, pp. 1707–1751.
- Jeffrey, Alan (1997). "Premonoidal categories and a graphical view of programs". In: *Preprint*.
- Mazurkiewicz, Antoni (July 1977). "Concurrent Program Schemes and their Interpretations". In: *DAIMI Report Series* 6.78. DOI: 10.7146/dpb.v6i78.7691.
- Mellies, Paul-André (2014). "Local states in string diagrams". In: *Rewriting and Typed Lambda Calculi: Joint International Conference*. Springer, pp. 334–348.
- Power, John and Edmund Robinson (1997). "Premonoidal Categories and Notions of Computation". In: *Math. Struct. Comput. Sci.* 7.5, pp. 453–468. DOI: 10.1017/S0960129597002375.
- Román, Mario and Pawel Sobociński (2025). "String Diagrams for Premonoidal Categories". In: *Logical Methods in Computer Science* Volume 21, Issue 2, 9. ISSN: 1860-5974. DOI: 10.46298/lmcs-21(2:9)2025.