

Thin spans and their associated intersection type system

Pierre Clairambault¹ Simon Forest²

¹LIS, CNRS

²Loria, Université de Lorraine

July 18th, 2026

The starting point: concurrent games

Concurrent games:

- ▶ **games**: sets A with associated **sets of configurations** $\mathcal{C}^g(A) \subseteq \mathcal{P}(A)$ (+ stuffs)
- ▶ **strategies**: sets σ mapped onto games,
with associated **sets of configurations** $\mathcal{C}^s(\sigma) \subseteq \mathcal{P}(\sigma)$ (+ stuffs)

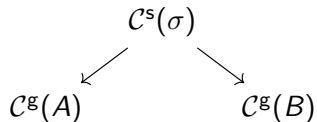
Among all the technicalities, two simple observations:

The starting point: concurrent games

Concurrent games:

- ▶ **games**: sets A with associated **sets of configurations** $\mathcal{C}^g(A) \subseteq \mathcal{P}(A)$ (+ stuffs)
- ▶ **strategies**: sets σ mapped onto games,
with associated **sets of configurations** $\mathcal{C}^s(\sigma) \subseteq \mathcal{P}(\sigma)$ (+ stuffs)

Among all the technicalities, two simple observations:



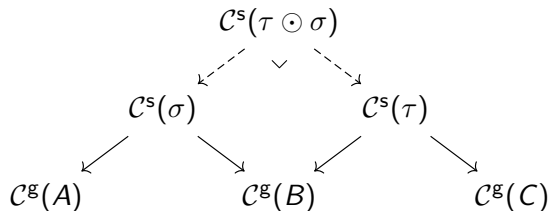
- 1) a strategy σ from game A to game B induces a **span**

The starting point: concurrent games

Concurrent games:

- ▶ **games**: sets A with associated **sets of configurations** $\mathcal{C}^g(A) \subseteq \mathcal{P}(A)$ (+ stuffs)
- ▶ **strategies**: sets σ mapped onto games,
with associated **sets of configurations** $\mathcal{C}^s(\sigma) \subseteq \mathcal{P}(\sigma)$ (+ stuffs)

Among all the technicalities, two simple observations:



2) when looking at configurations, strategy composition is **a pullback**
(under reasonable restrictions)

Thin spans, in a nutshell

What are **thin spans**? Different viewpoints:

- ▶ a **bicategorical span model** extracted from (thin) concurrent games
- ▶ a proof-relevant refinement of the **relational model** of linear logic
- ▶ a semantic model which represents programs through **witnesses of computation**

$$x : A \vdash s : B \quad \rightsquigarrow \quad \begin{array}{c} S \\ \swarrow \quad \searrow \\ !\llbracket A \rrbracket \quad \llbracket B \rrbracket \end{array}$$

Thin spans, in a nutshell

What are **thin spans**? Different viewpoints:

- ▶ a **bicategorical span model** extracted from (thin) concurrent games
- ▶ a proof-relevant refinement of the **relational model** of linear logic
- ▶ a semantic model which represents programs through **witnesses of computation**

$$x : A \vdash s : B \quad \rightsquigarrow \quad \begin{array}{c} S \\ \swarrow \quad \searrow \\ !\llbracket A \rrbracket \quad \llbracket B \rrbracket \end{array}$$

Thin spans, in a nutshell

What are **thin spans**? Different viewpoints:

- ▶ a **bicategorical span model** extracted from (thin) concurrent games
- ▶ a proof-relevant refinement of the **relational model** of linear logic
- ▶ a semantic model which represents programs through **witnesses of computation**

$$x : A \vdash s : B \quad \rightsquigarrow \quad \begin{array}{ccc} & S & \\ \swarrow & & \searrow \\ !\llbracket A \rrbracket & & \llbracket B \rrbracket \end{array}$$

Outline

From relations to thin spans

Interpreting programs in thin spans

An associated rigid intersection type system

Outline

From relations to thin spans

Interpreting programs in thin spans

An associated rigid intersection type system

The model **Rel** of **LL**

Objects: sets A, B, C , etc.

Morphisms $A \rightarrow B$: **relations** $R \subseteq A \times B$, i.e., sets of elements

$$a \multimap b$$

Exponential: $!A$ is $\mathcal{M}_{\text{fin}}(A)$, the set of **finite multisets** on A

(co)Kleisli category **Rel**_!: morphisms $A \rightarrow B$ are morphisms $!A \rightarrow B$ of **Rel**, that is, sets of elements

$$[a_1, \dots, a_n] \multimap b$$

Interpreting programs in $\mathbf{Rel}_I$

Since $\mathbf{Rel}_I$ is **cartesian closed**, one can interpret **programs** inside it.

$x : \mathbf{Bool} \vdash \text{if } x \text{ then ff else tt} : \mathbf{Bool}$

interpreted as

$$\{ [\mathbf{tt}] \multimap \mathbf{ff}, \quad [\mathbf{ff}] \multimap \mathbf{tt} \} \quad (\subseteq \mathcal{M}_{\text{fin}}(\mathbf{Bool}) \times \mathbf{Bool})$$

Interpreting programs in $\mathbf{Rel}_I$

Since $\mathbf{Rel}_I$ is **cartesian closed**, one can interpret **programs** inside it.

$x : \mathbf{Bool} \vdash \mathbf{if } x \mathbf{ then (if } x \mathbf{ then ff else tt) else (if } x \mathbf{ then tt else ff) : Bool}$

interpreted as

$$\{ [\mathbf{tt}, \mathbf{tt}] \multimap \mathbf{ff}, \quad [\mathbf{tt}, \mathbf{ff}] \multimap \mathbf{tt}, \quad [\mathbf{ff}, \mathbf{ff}] \multimap \mathbf{ff} \}$$

Interpreting programs in $\mathbf{Rel}_I$

Since $\mathbf{Rel}_I$ is **cartesian closed**, one can interpret **programs** inside it.

$x : \mathbf{Bool} \vdash \mathbf{if } x \mathbf{ then (if } x \mathbf{ then ff else tt) else (if } x \mathbf{ then tt else ff) : Bool}$

interpreted as

$$\{ [\mathbf{tt}, \mathbf{tt}] \multimap \mathbf{ff}, \quad [\mathbf{tt}, \mathbf{ff}] \multimap \mathbf{tt}, \quad [\mathbf{ff}, \mathbf{ff}] \multimap \mathbf{ff} \}$$

Observe that x is thought to have **multiple assignments**.

This is normal: in the context of CBN reduction and non-determinism, think of $x = \mathbf{tt} \vee \mathbf{ff}$.

Interpreting programs in $\mathbf{Rel}_I$

Since $\mathbf{Rel}_I$ is **cartesian closed**, one can interpret **programs** inside it.

$x : \mathbf{Bool} \vdash \text{if } x \text{ then (if } x \text{ then ff else tt) else (if } x \text{ then tt else ff)} : \mathbf{Bool}$

interpreted as

$$\{ [\text{tt}, \text{tt}] \multimap \text{ff}, \quad [\text{tt}, \text{ff}] \multimap \text{tt}, \quad [\text{ff}, \text{ff}] \multimap \text{ff} \}$$

Also observe: two different executions **get identified** in the interpretation.

Interpreting programs in $\mathbf{Rel}_!$

Since $\mathbf{Rel}_!$ is **cartesian closed**, one can interpret **programs** inside it.

$x: \mathbf{Bool} \vdash \mathbf{if} \ x \ \mathbf{then} \ (\mathbf{if} \ x \ \mathbf{then} \ \mathbf{ff} \ \mathbf{else} \ \mathbf{tt}) \ \mathbf{else} \ (\mathbf{if} \ x \ \mathbf{then} \ \mathbf{tt} \ \mathbf{else} \ \mathbf{ff}) : \mathbf{Bool}$

interpreted as

$$\{ [\mathbf{tt}, \mathbf{tt}] \multimap \mathbf{ff}, \quad [\mathbf{tt}, \mathbf{ff}] \multimap \mathbf{tt}, \quad [\mathbf{ff}, \mathbf{ff}] \multimap \mathbf{ff} \}$$

Also observe: two different executions **get identified** in the interpretation.

Hence, $\mathbf{Rel}_!$ aggregates different executions. Can we change that?

Spans as generalised relations

Relations

$\rightsquigarrow$

Spans

$$R \subseteq A \times B$$

$\rightsquigarrow$

$$\begin{array}{ccc} & S & \\ \partial_A^S \swarrow & & \searrow \partial_B^S \\ A & & B \end{array} \in \mathbf{Set}$$

$$\forall a \in A, b \in B, \quad a R b \text{ or } \neg a R b$$

$\rightsquigarrow$

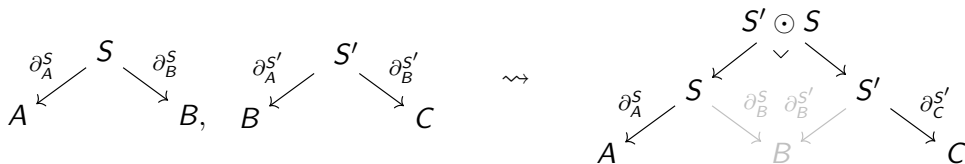
witnesses $S_{a,b} \subseteq S$ above (a, b)

Composition

For relations,

$$R \subseteq A \times B, \quad R' \subseteq B \times C, \quad \rightsquigarrow \quad R' \circ R = \{ (a, c) \in A \times C \mid \exists b \in B, \quad a R b \text{ and } b R' c \}$$

For spans,



where $S' \odot S$ is the **pullback**:

$$(S' \odot S) = \{ (s, s') \in S \times S' \mid \partial_B^S(s) = \partial_B^{S'}(s') \}$$

Identities/units

For relations, given $A \in \mathbf{Set}$,

$$\mathrm{id}_A \hat{=} \Delta_A = \{ (a, a) \in A \times A \}$$

For spans, given $A \in \mathbf{Set}$,

$$u_A = \begin{array}{ccc} & A & \\ \mathrm{id}_A \swarrow & & \searrow \mathrm{id}_A \\ A & & A \end{array}$$

Identities/units

For relations, given $A \in \mathbf{Set}$,

$$\mathrm{id}_A \quad \hat{=} \quad \Delta_A \quad = \quad \{ (a, a) \in A \times A \}$$

For spans, given $A \in \mathbf{Set}$,

$$u_A = \begin{array}{ccc} & A & \\ \mathrm{id}_A \swarrow & & \searrow \mathrm{id}_A \\ A & & A \end{array}$$

This **is not** a strict identity: given a span $A \leftarrow S \rightarrow B$, we have $S \odot u_A \neq S$, i.e.,

$$\begin{array}{ccccc} & S \odot u_A & & & \\ & \swarrow \quad \searrow & & & \\ \mathrm{id}_A \swarrow A & \quad \downarrow \quad & S & \searrow \partial_B^S & \\ A & \quad \swarrow \quad \searrow & & & \\ & \mathrm{id}_A \quad \partial_A^S & & & \end{array} \neq \begin{array}{ccc} & S & \\ \partial_A^S \swarrow & & \searrow \partial_B^S \\ A & & B \end{array}$$

Identities/units

For relations, given $A \in \mathbf{Set}$,

$$\mathrm{id}_A \hat{=} \Delta_A = \{ (a, a) \in A \times A \}$$

For spans, given $A \in \mathbf{Set}$,

$$u_A = \begin{array}{ccc} & A & \\ \mathrm{id}_A \swarrow & & \searrow \mathrm{id}_A \\ A & & A \end{array}$$

This **is not** a strict identity: given a span $A \leftarrow S \rightarrow B$, we have $S \odot u_A \cong S$, i.e.,

$$\begin{array}{ccccc} & & S \odot u_A & & \\ & \swarrow \text{dashed} & \downarrow & \searrow \text{dashed} & \\ \mathrm{id}_A \swarrow A & & \text{dashed } \mathrm{id}_A & \partial_A^S \swarrow S & \searrow \partial_B^S \\ A & & A & & B \end{array} \cong \begin{array}{ccc} & S & \\ \partial_A^S \swarrow & & \searrow \partial_B^S \\ A & & B \end{array}$$

Morphisms of spans

Given two spans $A \leftarrow S \rightarrow B$ and $A \leftarrow S' \rightarrow B$, a **morphism** is $F: S \rightarrow S'$ such that

$$\begin{array}{ccccc} & S & & & \\ \partial_A^S \swarrow & & \searrow \partial_B^S & & \\ A & = & F & = & B \\ \partial_A^{S'} \swarrow & & \searrow \partial_B^{S'} & & \\ & S' & & & \end{array}$$

When F is a bijection, we write $S \cong S'$.

Examples: $S \odot u_A \cong S$, $(S'' \odot S') \odot S \cong S'' \odot (S' \odot S)$, etc.

$\rightsquigarrow$ No category of spans, but a bicategory **Span**, whose 2-morphisms are morphisms of spans.

Towards a semantic model of spans

We already have a generalisation

category of relations $\rightsquigarrow$ bicategory of spans

What about cartesian closure (for higher-order programs)?

We follow the same recipe as in **Rel**

- ▶ equip spans with a structure of linear logic model
- ▶ derive a CCC by considering the associated Kleisli category

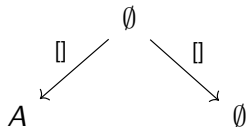
Some structure on **Span**

Just like for **Rel**, the cocartesian structure of **Set** translates to a cartesian structure on **Span**.

Some structure on **Span**

Just like for **Rel**, the cocartesian structure of **Set** translates to a cartesian structure on **Span**.

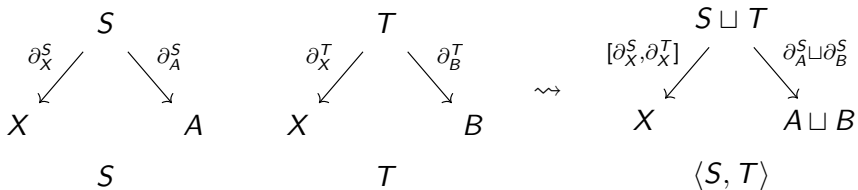
$\top \triangleq \emptyset$ is the terminal object of **Span**.



Some structure on **Span**

Just like for **Rel**, the cocartesian structure of **Set** translates to a cartesian structure on **Span**.

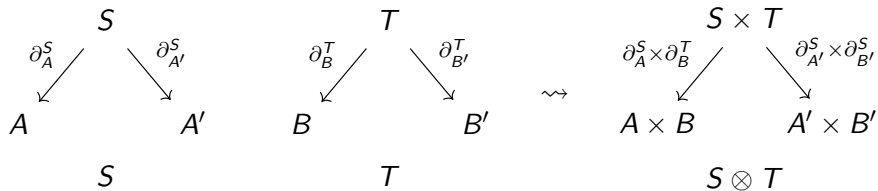
$A \& B \triangleq A \sqcup B$ is the cartesian product on **Span**.



Some structure on **Span**

Just like for **Rel**, the cartesian structure of **Set** translates to a monoidal structure on **Span**.

$A \otimes B \triangleq A \times B$ gives a tensor product on **Span**.



The exponential issue

An ingredient of an **LL** model: the exponential modality

Do we still have an exponential for **Span**?

The exponential issue

An ingredient of an **LL** model: the exponential modality

Do we still have an exponential for **Span**?

- First try: can we use $\mathcal{M}_{\text{fin}}(-)$ of **Rel** as exponential for **Span**?

Given $S \in \mathbf{Span}$, define

$$\mathcal{M}_{\text{fin}} \left(\begin{array}{ccc} & S & \\ \partial_A^S \swarrow & & \searrow \partial_B^S \\ A & & B \end{array} \right) = \begin{array}{ccc} & \mathcal{M}_{\text{fin}}(S) & \\ \mathcal{M}_{\text{fin}}(\partial_A^S) \swarrow & & \searrow \mathcal{M}_{\text{fin}}(\partial_B^S) \\ \mathcal{M}_{\text{fin}}(A) & & \mathcal{M}_{\text{fin}}(B) \end{array}$$

Problem: $\mathcal{M}_{\text{fin}}$ **does not respect composition**, i.e.,

$$\mathcal{M}_{\text{fin}}(S' \odot S) \not\cong \mathcal{M}_{\text{fin}}(S') \odot \mathcal{M}_{\text{fin}}(S).$$

Thus, not a functor **Span** $\rightarrow$ **Span**.

The exponential issue

An ingredient of an **LL** model: the exponential modality

Do we still have an exponential for **Span**?

- ▶ First try: can we use $\mathcal{M}_{\text{fin}}(-)$ of **Rel** as exponential for **Span**? No.
- ▶ Second try: can we use lists as exponential?

$$a_1, \dots, a_n \in A \quad \rightsquigarrow \quad [a_1; \dots ; a_n] \in \mathbf{List}(A)$$

The exponential issue

An ingredient of an **LL** model: the exponential modality

Do we still have an exponential for **Span**?

- First try: can we use $\mathcal{M}_{\text{fin}}(-)$ of **Rel** as exponential for **Span**? No.
- Second try: can we use lists as exponential?

$$a_1, \dots, a_n \in A \rightsquigarrow [a_1; \dots; a_n] \in \mathbf{List}(A)$$
$$\mathbf{List} \left(\begin{array}{ccc} & S & \\ \partial_A^S \swarrow & & \searrow \partial_B^S \\ A & & B \end{array} \right) = \begin{array}{ccc} & \mathbf{List}(S) & \\ \mathbf{List}(\partial_A^S) \swarrow & & \searrow \mathbf{List}(\partial_B^S) \\ \mathbf{List}(A) & & \mathbf{List}(B) \end{array}$$

The exponential issue

An ingredient of an **LL** model: the exponential modality

Do we still have an exponential for **Span**?

- First try: can we use $\mathcal{M}_{\text{fin}}(-)$ of **Rel** as exponential for **Span**? No.
- Second try: can we use lists as exponential?

$$a_1, \dots, a_n \in A \rightsquigarrow [a_1; \dots; a_n] \in \mathbf{List}(A)$$

$$\mathbf{List} \left(\begin{array}{ccc} & S & \\ \partial_A^S \swarrow & & \searrow \partial_B^S \\ A & & B \end{array} \right) = \begin{array}{ccc} & \mathbf{List}(S) & \\ \mathbf{List}(\partial_A^S) \swarrow & & \searrow \mathbf{List}(\partial_B^S) \\ \mathbf{List}(A) & & \mathbf{List}(B) \end{array}$$

We now have a (pseudo)functor, but no **Seely equivalence**

$$\text{see}_{A,B}: \mathbf{List} A \otimes \mathbf{List} B \begin{array}{c} \xrightarrow{\quad} \\ \simeq \\ \xleftarrow{\quad} \end{array} \mathbf{List}(A \& B) : \overline{\text{see}}_{A,B} \in \mathbf{Span}$$

because $[b_1; a_1; a_2; b_2] \neq [a_1; a_2; b_1; b_2]$: **lack of symmetries**

The exponential issue

An ingredient of an **LL** model: the exponential modality

Do we still have an exponential for **Span**?

- ▶ First try: can we use $\mathcal{M}_{\text{fin}}(-)$ of **Rel** as exponential for **Span**? No.
- ▶ Second try: can we use lists as exponential? Probably no.

$$a_1, \dots, a_n \in A \quad \rightsquigarrow \quad [a_1; \dots ; a_n] \in \mathbf{List}(A)$$

Taking into account symmetries

Problem:

- ▶ our spans are **set-based**
- ▶ there is **no adequate Seely equivalence** in this setting, since

$$[b_1; a_1; a_2; b_2] \cong [a_1; a_2; b_1; b_2]$$

makes no sense here.

We need a structure to make sense of these symmetries.

Groupoids

A **groupoid** is a set equipped with symmetries. It is given by:

- ▶ a set G
- ▶ for every $x, y \in G$, a set $G(x, y)$ of **symmetries** $\theta: x \cong y$
- ▶ for every $x \in G$, an **identity symmetry** $\text{id}_x: x \cong x$
- ▶ a **composition operation**:

$$\theta: x \cong x', \quad \theta': x' \cong x'' \quad \rightsquigarrow \quad \theta' \circ \theta: x \cong x''$$

- ▶ an **inverse operation**:

$$\theta: x \cong x' \quad \rightsquigarrow \quad \theta^{-1}: x' \cong x$$

In a nutshell: **a groupoid is a category with invertible morphisms.**

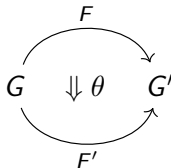
Groupoids

A **(groupoid) functor** $F: G \rightarrow G'$ is a functor between the underlying categories of G and G' .

A **(groupoid) natural transformation** $\theta: F \Rightarrow F': G \rightarrow G'$ is a natural transformation between the underlying functors of categories F and F' , that is,

an (adequately natural) family of symmetries $(\theta_x: F(x) \cong F'(x))_{x \in G}$

Graphically,



Groupoids

Let **Gpd** be the 2-category of groupoids, functors and natural transformations.

$\rightsquigarrow$ with groupoids, we can express symmetries between objects:

$$[b_1; a_1; a_2; b_2] \cong [a_1; a_2; b_1; b_2] \in \mathbf{List}^*(A \& B)$$

We now have a Seely equivalence

$$\text{see}_{A,B}: \mathbf{List}^* A \times \mathbf{List}^* B \begin{array}{c} \xrightarrow{\quad} \\ \simeq \\ \xleftarrow{\quad} \end{array} \mathbf{List}^*(A \sqcup B) : \overline{\text{see}}_{A,B} \in \mathbf{Gpd}$$

because the symmetries allow us to reindex:

$$\begin{array}{lcl} \overline{\text{see}}_{A,B} \circ \text{see}_{A,B} & = & \text{id}_{\mathbf{List}^* A \times \mathbf{List}^* B} \\ \text{see}_{A,B} \circ \overline{\text{see}}_{A,B} & \cong & \text{id}_{\mathbf{List}^*(A \sqcup B)} \end{array}$$

Groupoids

Let **Gpd** be the 2-category of groupoids, functors and natural transformations.

$\rightsquigarrow$ with groupoids, we can express symmetries between objects:

$$[b_1; a_1; a_2; b_2] \cong [a_1; a_2; b_1; b_2] \in \mathbf{List}^*(A \& B)$$

We now have a Seely equivalence

$$\text{see}_{A,B}: \mathbf{List}^* A \times \mathbf{List}^* B \begin{array}{c} \xrightarrow{\quad} \\ \simeq \\ \xleftarrow{\quad} \end{array} \mathbf{List}^*(A \sqcup B) : \overline{\text{see}}_{A,B} \in \mathbf{Gpd}$$

because the symmetries allow us to reindex:

$$\begin{array}{lcl} \overline{\text{see}}_{A,B} \circ \text{see}_{A,B} & = & \text{id}_{\mathbf{List}^* A \times \mathbf{List}^* B} \\ \text{see}_{A,B} \circ \overline{\text{see}}_{A,B} & \cong & \text{id}_{\mathbf{List}^*(A \sqcup B)} \end{array}$$

Span is dead, long live **Span**!

We (re)define **Span** as **Span(Gpd)**

Span is dead, long live **Span**!

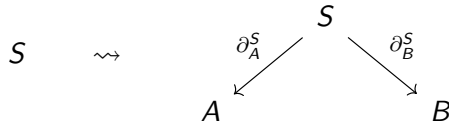
We (re)define **Span** as **Span(Gpd)**

- ▶ objects: groupoids $A, B, \dots$

Span is dead, long live Span!

We (re)define **Span** as **Span(Gpd)**

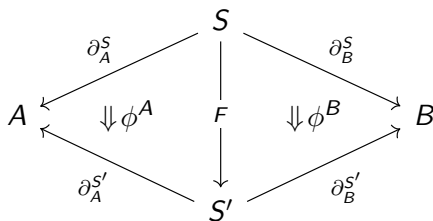
- ▶ objects: groupoids $A, B, \dots$
- ▶ 1-morphisms: spans $S, T, \dots$



Span is dead, long live Span!

We (re)define **Span** as **Span(Gpd)**

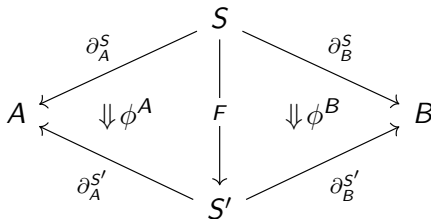
- ▶ objects: groupoids $A, B, \dots$
- ▶ 1-morphisms: spans $S, T, \dots$
- ▶ 2-morphisms: **weak span morphisms** $(F, \phi^A, \phi^B), (G, \psi^A, \psi^B), \dots$ as in



Span is dead, long live Span!

We (re)define **Span** as **Span(Gpd)**

- ▶ objects: groupoids $A, B, \dots$
- ▶ 1-morphisms: spans $S, T, \dots$
- ▶ 2-morphisms: **weak span morphisms** $(F, \phi^A, \phi^B), (G, \psi^A, \psi^B), \dots$ as in

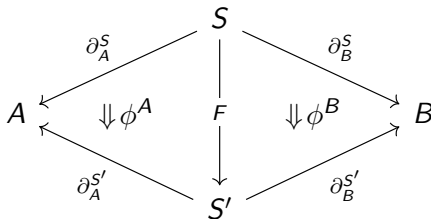


We can now hope that the Seelye equivalence of **Gpd** lifts in **Span(Gpd)**.

Span is dead, long live Span!

We (re)define **Span** as **Span(Gpd)**

- ▶ objects: groupoids $A, B, \dots$
- ▶ 1-morphisms: spans $S, T, \dots$
- ▶ 2-morphisms: **weak span morphisms** $(F, \phi^A, \phi^B), (G, \psi^A, \psi^B), \dots$ as in



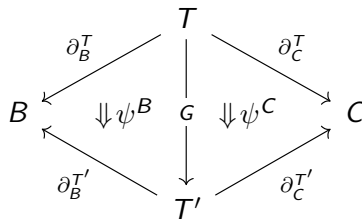
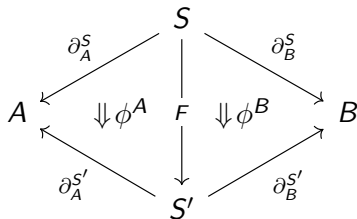
We can now hope that the Seely equivalence of **Gpd** lifts in **Span(Gpd)**.

But, for now: **do we have a bicategory?**

The horizontal composition issue

But is $\mathbf{Span} = \mathbf{Span}(\mathbf{Gpd})$ a bicategory?

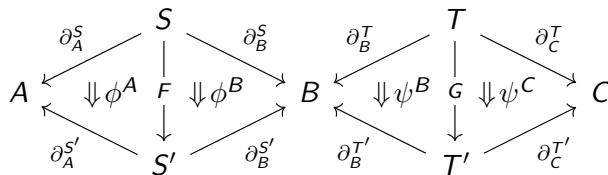
In particular, are we able to define a **horizontal composition**?



The horizontal composition issue

But is $\mathbf{Span} = \mathbf{Span}(\mathbf{Gpd})$ a bicategory?

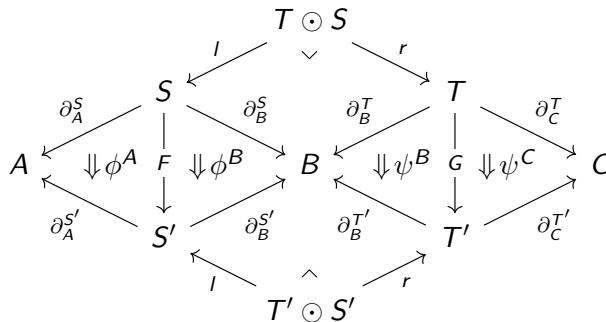
In particular, are we able to define a **horizontal composition**?



The horizontal composition issue

But is $\mathbf{Span} = \mathbf{Span}(\mathbf{Gpd})$ a bicategory?

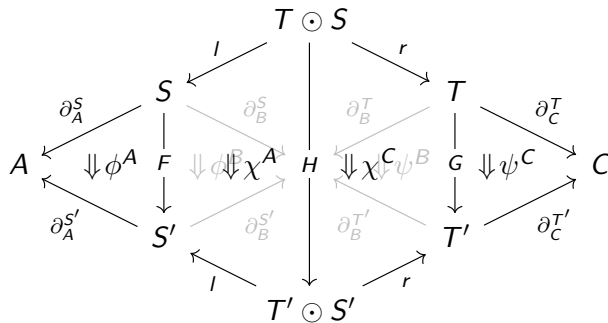
In particular, are we able to define a **horizontal composition**?



The horizontal composition issue

But is $\mathbf{Span} = \mathbf{Span}(\mathbf{Gpd})$ a bicategory?

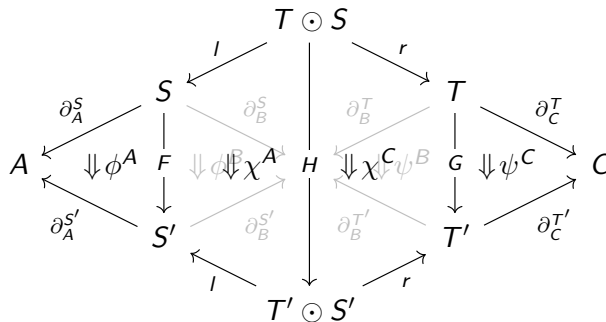
In particular, are we able to define a **horizontal composition**?



The horizontal composition issue

But is $\mathbf{Span} = \mathbf{Span}(\mathbf{Gpd})$ a bicategory?

In particular, are we able to define a **horizontal composition**?

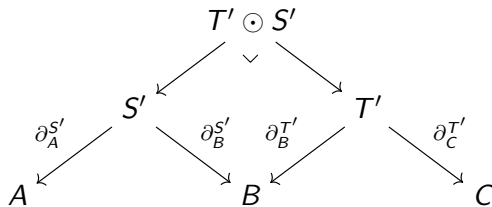


$\rightsquigarrow$ neither **existence** nor **unicity** of H, χ^A, χ^C are guaranteed!

So **Span** is not a bicategory!

Pullbacks and bipullbacks

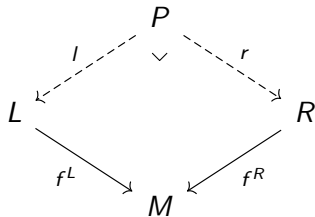
For the horizontal composition **to exist**, we need the pullback of



to be a **bipullback**.

Bipullbacks

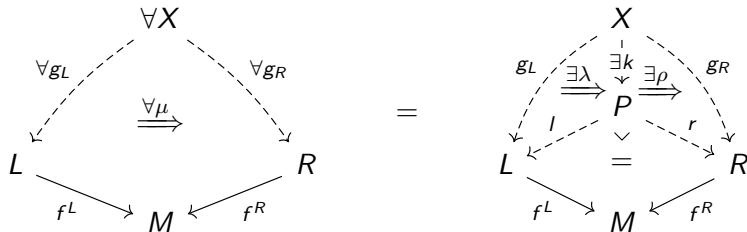
Let a pullback



in **Gpd**.

Bipullbacks

It is a bipullback when every pseudocone can be decomposed along it, i.e.,



and additional conditions.

The factorizations (k, λ, ρ) are **unique but only up to a 2-morphism** $X \begin{array}{c} \xrightarrow{k} \\ \Downarrow \kappa \\ \xrightarrow{k'} \end{array} P$

If P is discrete (meaning: no non-identity symmetries): **unicity is on the nose**
since $\kappa_x = \text{id}_{k(x)}$ for all $x \in X$, necessarily!

Orthogonalities

We enforce that

- ▶ some pullbacks are bipullbacks ($\rightsquigarrow$ existence of composition of 2-morphisms)
- ▶ some pullbacks are discrete ($\rightsquigarrow$ unicity of composition of 2-morphisms)

using standard **biorthogonality techniques** [Hyland,Schalk].

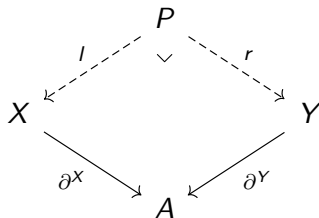
Orthogonalities

Given a groupoid A , a **prestrategy** S on A is a pair $(X \in \mathbf{Gpd}, \partial^X: X \rightarrow A)$.

$$X \xrightarrow{\partial^X} A \quad \in \quad \mathbf{Gpd}$$

Orthogonalities

First orthogonality: two prestrategies X, Y on A are said **uniformly orthogonal**, denoted $X \perp Y$, when the pullback



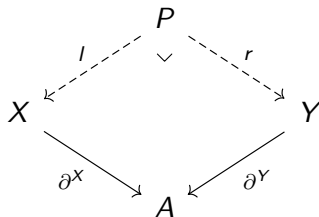
is a **bipullback**.

Given a class $\mathcal{S}$ of prestrategies on A ,

$$\mathcal{S}^\perp \hat{=} \{ Y \in \mathbf{PreStrat}(A) \mid \forall X \in \mathcal{S}, X \perp Y \}.$$

Orthogonalities

Second orthogonality: two prestrategies X, Y on A are said **thinly orthogonal**, denoted $X \perp\!\!\!\perp Y$, when the pullback



is **discrete** (meaning: P only has identity symmetries).

Given a class $\mathcal{S}$ of prestrategies on A ,

$$\mathcal{S}^{\perp\!\!\!\perp} \triangleq \{ Y \in \mathbf{PreStrat}(A) \mid \forall X \in \mathcal{S}, X \perp\!\!\!\perp Y \}.$$

Thin groupoids

A **thin groupoid** is a tuple

$$\mathcal{A} = (A, A_-, A_+, \mathcal{U}_{\mathcal{A}}, T_{\mathcal{A}})$$

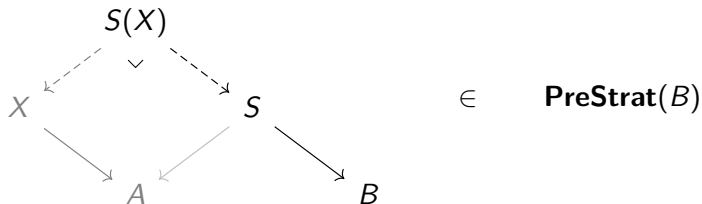
where

- ▶ A is a groupoid
- ▶ $\mathcal{U}_{\mathcal{A}} \subseteq \mathbf{PreStrat}(A)$ is such that $\mathcal{U}_{\mathcal{A}}^{\perp\perp} = \mathcal{U}_{\mathcal{A}}$
- ▶ $T_{\mathcal{A}} \subseteq \mathcal{U}_{\mathcal{A}}$ is such that $T_{\mathcal{A}}^{\parallel\parallel} = T_{\mathcal{A}}$
- ▶ A_- and A_+ are wide subgroupoids of A such that

$$A_- \hookrightarrow A \in T_{\mathcal{A}} \quad \text{and} \quad A_+ \hookrightarrow A \in T_{\mathcal{A}}^{\parallel\parallel}$$

Thin spans

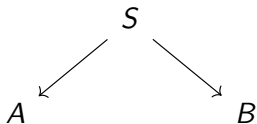
Observation: a span $A \leftarrow S \rightarrow B$ between groupoids A and B maps $X \in \mathbf{PreStrat}(A)$ to some $S(X) \in \mathbf{PreStrat}(B)$, by pullback:



and dually maps elements of $\mathbf{PreStrat}(B)$ to elements of $\mathbf{PreStrat}(A)$.

Thin spans

A **thin span** between two thin groupoids $\mathcal{A}$ and $\mathcal{B}$ is a span



such that

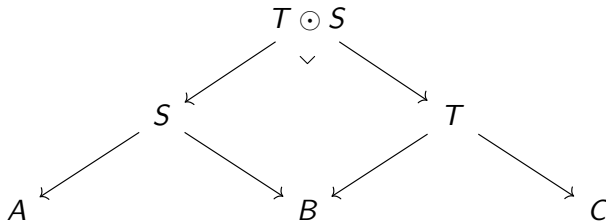
- ▶ S maps $X \in \mathcal{U}_{\mathcal{A}}$ to some $Y \in \mathcal{U}_{\mathcal{B}}$
- ▶ S maps $X \in \mathcal{T}_{\mathcal{A}}$ to some $Y \in \mathcal{T}_{\mathcal{B}}$
- ▶ and dually, from $\mathcal{B}$ to $\mathcal{A}$.

(standard conditions for categories derived from biorthogonalities).

Thin spans

Proposition

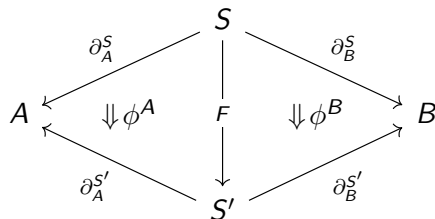
Given thin spans $A \leftarrow S \rightarrow B$ and $B \leftarrow T \rightarrow C$ between thin groupoids $\mathcal{A}, \mathcal{B}, \mathcal{C}$, the pullback



is a bipullback, and the resulting span $A \leftarrow T \odot S \rightarrow C$ is thin.

Positive 2-morphisms

We now restrict the 2-morphisms of spans to **positive 2-morphisms**, that is



such that $\phi_s^A \in A_-$ and $\phi_s^B \in B_+$ for every $s \in S$.

The two orthogonalities $\perp$ and $\perp\!\!\!\perp$ ensure that horizontal composition **exists** and **is unique**.

The bicategory **Thin**

Theorem (Clairambault, F.)

We get a bicategory **Thin**:

- ▶ objects: **thin groupoids** $\mathcal{A}, \mathcal{B}, \dots$
- ▶ 1-morphisms: **thin spans**
- ▶ 2-morphisms: **positive 2-morphisms** of spans

LL operations on thin groupoids: $\mathcal{A} \otimes \mathcal{B}$, $\mathcal{A} \& \mathcal{B}$, etc.

The exponential modality

Recall that the exponential $! : \mathbf{Rel} \rightarrow \mathbf{Rel}$ is derived from the monad $\mathcal{M}_{\text{fin}} : \mathbf{Set} \rightarrow \mathbf{Set}$.

We derive an exponential $! : \mathbf{Thin} \rightarrow \mathbf{Thin}$ from the monad $\mathbf{List}^* : \mathbf{Gpd} \rightarrow \mathbf{Gpd}$.

The exponential modality

The monad $\mathbf{List}^*: \mathbf{Gpd} \rightarrow \mathbf{Gpd}$? The “free strict symmetric monoidal construction”.

To $A \in \mathbf{Gpd}$, associates $\mathbf{List}^*(A) \in \mathbf{Gpd}$:

- ▶ objects: lists $[a_1; \dots; a_n] \in \mathbf{List}(\mathbf{Ob}(A))$;
- ▶ morphisms $[a_1; \dots; a_n] \rightarrow [a'_1; \dots; a'_m]$: pairs $(\pi, (f_i)_{1 \leq i \leq n})$ where
 - ▶ π is a bijection $\{1, \dots, n\} \rightarrow \{1, \dots, m\}$;
 - ▶ f_i is a symmetry $a_i \cong a'_{\pi(i)} \in A$.

The unit $\eta_A: A \rightarrow \mathbf{List}^*(A)$: maps $a \in A$ to $[a]$;

The multiplication $\mu_A: \mathbf{List}^*(\mathbf{List}^*(A)) \rightarrow \mathbf{List}^*(A)$: merges lists of lists into lists.

The exponential modality

The monad $\mathbf{List}^*: \mathbf{Gpd} \rightarrow \mathbf{Gpd}$? The “free strict symmetric monoidal construction”.

To $A \in \mathbf{Gpd}$, associates $\mathbf{List}^*(A) \in \mathbf{Gpd}$:

- ▶ objects: lists $[a_1; \dots; a_n] \in \mathbf{List}(\mathbf{Ob}(A))$;
- ▶ morphisms $[a_1; \dots; a_n] \rightarrow [a'_1; \dots; a'_m]$: pairs $(\pi, (f_i)_{1 \leq i \leq n})$ where
 - ▶ π is a bijection $\{1, \dots, n\} \rightarrow \{1, \dots, m\}$;
 - ▶ f_i is a symmetry $a_i \cong a'_{\pi(i)} \in A$.

The unit $\eta_A: A \rightarrow \mathbf{List}^*(A)$: maps $a \in A$ to $[a]$;

The multiplication $\mu_A: \mathbf{List}^*(\mathbf{List}^*(A)) \rightarrow \mathbf{List}^*(A)$: merges lists of lists into lists.

The exponential modality

The monad $\mathbf{List}^*: \mathbf{Gpd} \rightarrow \mathbf{Gpd}$? The “free strict symmetric monoidal construction”.

To $A \in \mathbf{Gpd}$, associates $\mathbf{List}^*(A) \in \mathbf{Gpd}$:

- ▶ objects: lists $[a_1; \dots; a_n] \in \mathbf{List}(\mathbf{Ob}(A))$;
- ▶ morphisms $[a_1; \dots; a_n] \rightarrow [a'_1; \dots; a'_m]$: pairs $(\pi, (f_i)_{1 \leq i \leq n})$ where
 - ▶ π is a bijection $\{1, \dots, n\} \rightarrow \{1, \dots, m\}$;
 - ▶ f_i is a symmetry $a_i \cong a'_{\pi(i)} \in A$.

The unit $\eta_A: A \rightarrow \mathbf{List}^*(A)$: maps $a \in A$ to $[a]$;

The multiplication $\mu_A: \mathbf{List}^*(\mathbf{List}^*(A)) \rightarrow \mathbf{List}^*(A)$: merges lists of lists into lists.

The exponential modality

The monad $\mathbf{List}^*: \mathbf{Gpd} \rightarrow \mathbf{Gpd}$? The “free strict symmetric monoidal construction”.

To $A \in \mathbf{Gpd}$, associates $\mathbf{List}^*(A) \in \mathbf{Gpd}$:

- ▶ objects: lists $[a_1; \dots; a_n] \in \mathbf{List}(\mathbf{Ob}(A))$;
- ▶ morphisms $[a_1; \dots; a_n] \rightarrow [a'_1; \dots; a'_m]$: pairs $(\pi, (f_i)_{1 \leq i \leq n})$ where
 - ▶ π is a bijection $\{1, \dots, n\} \rightarrow \{1, \dots, m\}$;
 - ▶ f_i is a symmetry $a_i \cong a'_{\pi(i)} \in A$.

The unit $\eta_A: A \rightarrow \mathbf{List}^*(A)$: maps $a \in A$ to $[a]$;

The multiplication $\mu_A: \mathbf{List}^*(\mathbf{List}^*(A)) \rightarrow \mathbf{List}^*(A)$: merges lists of lists into lists.

The exponential modality

The monad $\mathbf{List}^*: \mathbf{Gpd} \rightarrow \mathbf{Gpd}$? The “free strict symmetric monoidal construction”.

To $A \in \mathbf{Gpd}$, associates $\mathbf{List}^*(A) \in \mathbf{Gpd}$:

- ▶ objects: lists $[a_1; \dots; a_n] \in \mathbf{List}(\mathrm{Ob}(A))$;
- ▶ morphisms $[a_1; \dots; a_n] \rightarrow [a'_1; \dots; a'_m]$: pairs $(\pi, (f_i)_{1 \leq i \leq n})$ where
 - ▶ π is a bijection $\{1, \dots, n\} \rightarrow \{1, \dots, m\}$;
 - ▶ f_i is a symmetry $a_i \cong a'_{\pi(i)} \in A$.

The unit $\eta_A: A \rightarrow \mathbf{List}^*(A)$: maps $a \in A$ to $[a]$;

The multiplication $\mu_A: \mathbf{List}^*(\mathbf{List}^*(A)) \rightarrow \mathbf{List}^*(A)$: merges lists of lists into lists.

The exponential modality

We get an exponential

$$!: \mathbf{Thin} \rightarrow \mathbf{Thin}$$

where

$$!\mathcal{A} \quad \hat{=} \quad (\mathbf{List}^*(A), \dots)$$

for every thin groupoid $\mathcal{A}$ and

$$! \left(\begin{array}{ccc} & S & \\ \partial_A^S \swarrow & & \searrow \partial_B^S \\ A & & B \end{array} \right) \quad \hat{=} \quad \begin{array}{ccc} & \mathbf{List}^*(S) & \\ \mathbf{List}^*(\partial_A^S) \swarrow & & \searrow \mathbf{List}^*(\partial_B^S) \\ \mathbf{List}^*(A) & & \mathbf{List}^*(B) \end{array}$$

for every thin span $S: \mathcal{A} \rightarrow \mathcal{B}$.

The exponential modality

The structure of comonad of $!$ is derived from the monad structure of $\mathbf{List}^*$.

Given a thin groupoid $\mathcal{A}$,

$$\breve{\eta}_{\mathcal{A}} = \begin{array}{ccc} & A & \\ \eta_A \swarrow & & \searrow \text{id}_A \\ \mathbf{List}^*(A) & & A \end{array}$$

$$\breve{\mu}_{\mathcal{A}} = \begin{array}{ccc} & \mathbf{List}^*(\mathbf{List}^*(A)) & \\ \mu_A \swarrow & & \searrow \text{id}_{!!A} \\ \mathbf{List}^*(A) & & \mathbf{List}^*(\mathbf{List}^*(A)) \end{array} .$$

The Kleisli bicategory

We thus get a Kleisli bicategory $\mathbf{Thin}_!$ with $! = \mathbf{List}^*$, whose 1-morphisms $\mathcal{A} \rightarrow \mathcal{B}$ are of the form

$$\begin{array}{ccc} & S & \\ \partial_{!A}^S \swarrow & & \searrow \partial_B^S \\ !A & & B \end{array} .$$

The Kleisli bicategory

We thus get a Kleisli bicategory $\mathbf{Thin}_!$ with $! = \mathbf{List}^*$, whose 1-morphisms $\mathcal{A} \rightarrow \mathcal{B}$ are of the form

$$\begin{array}{ccc} & S & \\ \partial_{!A}^S \swarrow & & \searrow \partial_B^S \\ !A & & B \end{array} .$$

In categorical models of $\mathbf{LL}$, the Kleisli category is cartesian closed.

The Kleisli bicategory

We thus get a Kleisli bicategory $\mathbf{Thin}_!$ with $! = \mathbf{List}^*$, whose 1-morphisms $\mathcal{A} \rightarrow \mathcal{B}$ are of the form

$$\begin{array}{ccc} & S & \\ \partial_{!A}^S \swarrow & & \searrow \partial_B^S \\ !A & & B \end{array} .$$

In categorical models of $\mathbf{LL}$, the Kleisli category is cartesian closed.

Theorem (Clairambault, F.)

The bicategory $\mathbf{Thin}_!$ is cartesian closed.

Outline

From relations to thin spans

Interpreting programs in thin spans

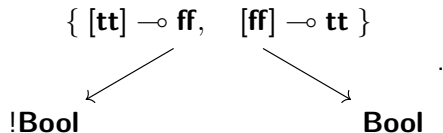
An associated rigid intersection type system

Examples of interpretations

Example 1:

$x : \mathbf{Bool} \vdash \text{if } x \text{ then ff else tt} : \mathbf{Bool}$

interpreted as the span (which happens to be a relation)



Examples of interpretations

Example 2:

$x : \mathbf{Bool} \vdash \mathbf{if } x \mathbf{ then (if } x \mathbf{ then ff else tt) else (if } x \mathbf{ then tt else ff) : Bool}$

interpreted as the span (which happens to be a relation)

$\{ [\mathbf{tt}; \mathbf{tt}] \multimap \mathbf{ff}, [\mathbf{tt}; \mathbf{ff}] \multimap \mathbf{tt}, [\mathbf{ff}; \mathbf{tt}] \multimap \mathbf{tt}, [\mathbf{ff}; \mathbf{ff}] \multimap \mathbf{ff} \}$

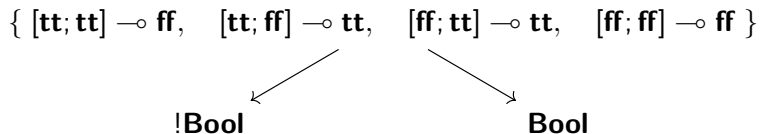


Examples of interpretations

Example 2:

$x : \mathbf{Bool} \vdash \mathbf{if } x \mathbf{ then (if } x \mathbf{ then ff else tt) else (if } x \mathbf{ then tt else ff) : Bool}$

interpreted as the span (which happens to be a relation)



to compare with the interpretation in $\mathbf{Rel}_!$:

$$\{ [\mathbf{tt}, \mathbf{tt}] \multimap \mathbf{ff}, [\mathbf{tt}, \mathbf{ff}] \multimap \mathbf{tt}, [\mathbf{ff}, \mathbf{ff}] \multimap \mathbf{tt} \}.$$

Interpretation of types

Considered calculus: STLC

$$s, t, \dots ::= x \in \text{Var} \quad | \quad \lambda x. t \quad | \quad t u$$

Considered types:

$$A, B, \dots ::= \mathbf{Bool} \quad | \quad A \rightarrow B$$

Interpretation $\llbracket A \rrbracket$ of a type A (only the groupoid):

$$\llbracket \mathbf{Bool} \rrbracket = 1 \sqcup 1 \qquad \llbracket A \rightarrow B \rrbracket = !\llbracket A \rrbracket \times \llbracket B \rrbracket$$

Interpretation of types

Considered calculus: STLC

$$s, t, \dots ::= x \in \text{Var} \quad | \quad \lambda x. t \quad | \quad t \ u$$

Considered types:

$$A, B, \dots ::= \mathbf{Bool} \quad | \quad A \rightarrow B$$

Interpretation $\llbracket A \rrbracket$ of a type A (only the groupoid):

$$\llbracket \mathbf{Bool} \rrbracket = 1 \sqcup 1 \qquad \llbracket A \rightarrow B \rrbracket = !\llbracket A \rrbracket \times \llbracket B \rrbracket$$

Interpretation of types

Considered calculus: STLC

$$s, t, \dots ::= x \in \text{Var} \quad | \quad \lambda x. t \quad | \quad t \ u$$

Considered types:

$$A, B, \dots ::= \mathbf{Bool} \quad | \quad A \rightarrow B$$

Interpretation $\llbracket A \rrbracket$ of a type A (only the groupoid):

$$\llbracket \mathbf{Bool} \rrbracket = 1 \sqcup 1 \qquad \llbracket A \rightarrow B \rrbracket = !\llbracket A \rrbracket \times \llbracket B \rrbracket$$

Interpretation of contexts

Interpretation $\llbracket \Gamma \rrbracket$ of a context $\Gamma = (x_1 : A_1, \dots, x_n : A_n)$:

$$\llbracket \Gamma \rrbracket = !\llbracket A_1 \rrbracket \times \dots \times !\llbracket A_n \rrbracket$$

Proposition

The underlying groupoid of $\llbracket \Gamma \rrbracket$ is equipped with a structure of monoid.

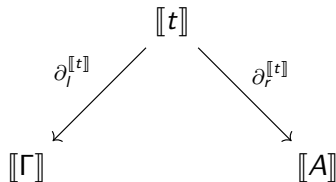
- multiplication: given $\gamma = (\gamma_1, \dots, \gamma_n)$ and $\delta = (\delta_1, \dots, \delta_n)$ in $\llbracket \Gamma \rrbracket$,

$$\gamma \vec{\oplus} \delta = (\gamma_1 ++ \delta_1, \dots, \gamma_n ++ \delta_n)$$

where $\gamma_i ++ \delta_i$ is the concatenation of lists in $!\llbracket A_i \rrbracket$

Interpretation of terms

Given $\Gamma = x_1 : A_1, \dots, x_n : A_n$ and a derivation $\Gamma \vdash t : A$, we define a thin span



by induction on the derivation.

Interpretation of terms

Variable case: for $\Gamma = x_1 : A_1, \dots, x_n : A_n$ and $i \in \{ 1, \dots, n \}$,

$$\begin{array}{c}
 \llbracket x_i \rrbracket = \begin{array}{c} \begin{array}{c} \llbracket A_i \rrbracket \\ \swarrow \\ \langle \square, \dots, \eta_{\llbracket A_i \rrbracket}, \dots, \square \rangle \\ \nwarrow \\ \square \end{array} \begin{array}{c} \searrow \text{id}_{\llbracket A_i \rrbracket} \\ \llbracket A_i \rrbracket \end{array} \\ \uparrow \\ \begin{array}{c} \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \end{array} \end{array}
 \end{array}$$

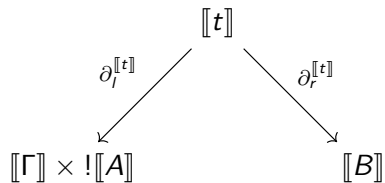
$\rightsquigarrow$ “for every $a \in \llbracket A_i \rrbracket$, there is one computation consuming $(\square, \dots, [a], \dots, \square)$ and producing a ”

Interpretation of terms

λ -abstraction case: given a derivation of $(\Gamma, x : A) \vdash t : B$,

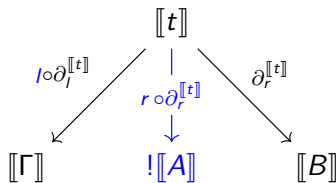
Interpretation of terms

λ -abstraction case: given a derivation of $(\Gamma, x : A) \vdash t : B$,



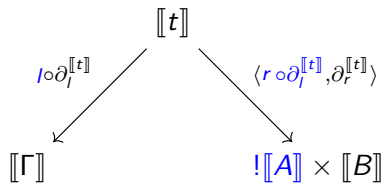
Interpretation of terms

λ -abstraction case: given a derivation of $(\Gamma, x : A) \vdash t : B$,



Interpretation of terms

λ -abstraction case: given a derivation of $(\Gamma, x : A) \vdash t : B$,



Interpretation of terms

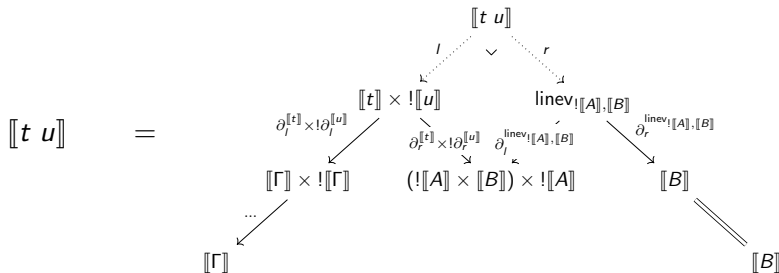
λ -abstraction case: given a derivation of $(\Gamma, x : A) \vdash t : B$,

$$\llbracket \lambda x. t \rrbracket = \begin{array}{c} \llbracket t \rrbracket \\ \swarrow \scriptstyle l \circ \partial_l^{\llbracket t \rrbracket} \quad \searrow \scriptstyle \langle r \circ \partial_l^{\llbracket t \rrbracket}, \partial_r^{\llbracket t \rrbracket} \rangle \\ \llbracket \Gamma \rrbracket \qquad \qquad \qquad !\llbracket A \rrbracket \times \llbracket B \rrbracket \end{array}$$

$\rightsquigarrow$ “if t consumes $(\gamma, [a_i]_i)$ to produce b , then $\lambda x. t$ consumes γ to produce $[a_i]_i \multimap b$ ”

Interpretation of terms

Application case: given a derivation of $\Gamma \vdash t : A \rightarrow B$ and of $\Gamma \vdash u : A$,



$\rightsquigarrow$ “if t consumes γ to produce $[a_i]_i \multimap b$ and u consumes δ to produce $[a_i]_i$, then $t \ u$ consumes $\gamma \vec{\oplus} \delta$ to produce b ”

Outline

From relations to thin spans

Interpreting programs in thin spans

An associated rigid intersection type system

ITS for **Thin**

[De 18]: interpretations of programs in **Rel** can be presented syntactically through an **intersection type system**.

Can we have a similar presentation for **Thin**?

Yes, by shifting from bags as **multisets** to bags as **lists**.

ITS for Thin

Simple types considered:

$$A, B, \dots ::= \mathbf{Bool} \mid A \rightarrow B$$

Refinement types **values** and intersection types **multivalues**:

$$\begin{aligned}\alpha, \beta, \dots &::= \mathbf{tt} \mid \mathbf{ff} \mid \kappa \multimap \alpha \\ \kappa, \lambda, \dots &::= [\alpha_1, \dots, \alpha_n] \quad (n \in \mathbb{N})\end{aligned}$$

where $[\alpha_1, \dots, \alpha_n]$ is a **list** of elements.

Refinement judgements $\alpha \triangleleft A$ and $\kappa \triangleleft_m A$ and their rules:

$$\begin{array}{c} \frac{}{\mathbf{ff} \triangleleft \mathbf{Bool}} \quad \frac{}{\mathbf{tt} \triangleleft \mathbf{Bool}} \quad \frac{\kappa \triangleleft_m A \quad \beta \triangleleft B}{\kappa \multimap \alpha \triangleleft A \rightarrow B} \quad \frac{\forall i \in \{1, \dots, n\} \quad \alpha_i \triangleleft A}{[\alpha_1, \dots, \alpha_n] \triangleleft_m A}\end{array}$$

Resource contexts: sequences Θ of bindings of the form

$$\Theta, \Sigma, \dots \quad ::= \quad (x_i : [a_{i,1}, \dots, a_{i,n_i}] \triangleleft A_i)_{1 \leq i \leq n} \quad (n \in \mathbb{N})$$

Addition of resource contexts: given

$$\Theta = (x_i : \kappa_i \triangleleft A_i)_{1 \leq i \leq n} \quad \Sigma = (x_i : \lambda_i \triangleleft A_i)_{1 \leq i \leq n}$$

we put

$$\Theta \vec{\oplus} \Sigma = (x_i : (\kappa_i ++ \lambda_i) \triangleleft A_i)_{1 \leq i \leq n}$$

where $\kappa_i ++ \lambda_i$ stands for the **concatenation of lists**.

ITS for **Thin**

Intersection type judgements $\Theta \vdash t : \alpha \triangleleft A$ and $\Theta \vdash_{\text{m}} t : \kappa \triangleleft A$ and their rules:

$$\text{(IT-VAR)} \quad \frac{\alpha \triangleleft A_i}{(x_1 : [] \triangleleft A_1, \dots, x_i : [\alpha] \triangleleft A_i, \dots, x_n : [] \triangleleft A_n) \vdash x_i : \alpha \triangleleft A_i}$$

$$\text{(IT-APP)} \quad \frac{\Theta \vdash t : \kappa \multimap \beta \triangleleft A \rightarrow B \quad \Theta' \vdash_{\text{m}} u : \kappa \triangleleft A}{\Theta \vec{\oplus} \Theta' \vdash t \ u : \beta \triangleleft B}$$

$$\text{(IT-LAM)} \quad \frac{(\Theta, x : \kappa \triangleleft A) \vdash t : \beta \triangleleft B}{\Theta \vdash \lambda x. t : \kappa \multimap \beta \triangleleft A \rightarrow B}$$

$$\text{(IT-MULT)} \quad \frac{\Theta_i \vdash t : \alpha_i \triangleleft A \quad \text{for } i \in \{1, \dots, n\}}{\Theta_1 \vec{\oplus} \dots \vec{\oplus} \Theta_n \vdash_{\text{m}} t : [\alpha_1, \dots, \alpha_n] \triangleleft A}$$

We could add **if**'s to our language and get the following rules:

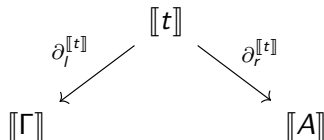
$$(IT-IF-THEN) \quad \frac{\Theta \vdash t : \mathbf{tt} \triangleleft \mathbf{Bool} \quad \Theta_{\text{then}} \vdash u : \alpha \triangleleft A}{\Theta \oplus \vec{\Theta}_{\text{then}} \vdash_m \mathbf{if } t \mathbf{ then } u \mathbf{ else } v : \alpha \triangleleft A}$$

$$(IT-IF-ELSE) \quad \frac{\Theta \vdash t : \mathbf{tt} \triangleleft \mathbf{Bool} \quad \Theta_{\text{else}} \vdash v : \alpha \triangleleft A}{\Theta \oplus \vec{\Theta}_{\text{else}} \vdash_m \mathbf{if } t \mathbf{ then } u \mathbf{ else } v : \alpha \triangleleft A}$$

ITS for **Thin**

Theorem (Clairambault, F.)

Given a derivation $\Gamma \vdash t : A$ and its interpretation



we have a bijection

$$\text{Ob}(\llbracket t \rrbracket) \cong \{ p \mid p \text{ derivation of } \Theta \vdash t : \alpha \triangleleft A \text{ for } \Theta \triangleleft \Gamma \}.$$

- Contrary to **Rel**, **no quotient is required** here for the bijection.

A known broken system

This system is the ITS of **Rel** where commutativity is dropped.

It is known for **not** satisfying subject reduction! [Kfoury,Wells]

A known broken system

In the context $f: \mathbf{Bool} \rightarrow \mathbf{Bool} \rightarrow \mathbf{Bool}, x: \mathbf{Bool}$ consider

$$t_1 = (\lambda y. \lambda z. f \ z \ y) \ x \ x \qquad t_2 = f \ x \ x$$

A known broken system

In the context $f: \mathbf{Bool} \rightarrow \mathbf{Bool} \rightarrow \mathbf{Bool}, x: \mathbf{Bool}$ consider

$$t_1 = (\lambda y. \lambda z. f \ z \ y) \ x \ x \quad t_2 = f \ x \ x$$

Given the resource context

$$\Theta_{\mathbf{tt}, \mathbf{ff}} = (f: [[\mathbf{ff}]] \multimap [\mathbf{tt}] \multimap \mathbf{tt}] \triangleleft \mathbf{Bool} \rightarrow \mathbf{Bool}, x: [\mathbf{tt}; \mathbf{ff}] \triangleleft \mathbf{Bool})$$

we have

$$\Theta_{\mathbf{tt}, \mathbf{ff}} \vdash t_1: \mathbf{tt} \triangleleft \mathbf{Bool}$$

$$\Theta_{\mathbf{tt}, \mathbf{ff}} \not\vdash t_2: \mathbf{tt} \triangleleft \mathbf{Bool}$$

while $t_1 \rightarrow^* t_2 \rightsquigarrow$ **subject reduction not satisfied**

A known broken system

But in **Thin**, the reduction $t_1 \rightarrow^* t_2$ is interpreted as a reindexing

$$\llbracket t_1 \rightarrow^* t_2 \rrbracket : \llbracket t_1 \rrbracket \rightarrow \llbracket t_2 \rrbracket$$

so that subject reduction is weakly recovered:

$$\Theta_{\mathbf{tt}, \mathbf{ff}} \vdash t_1 : \mathbf{tt} \triangleleft \mathbf{Bool} \quad \cong \quad \Theta_{\mathbf{ff}, \mathbf{tt}} \vdash t_2 : \mathbf{tt} \triangleleft \mathbf{Bool}$$

(The above is informal!)

ITS for symmetries

Let's remember that thin spans are spans of groupoids, **with symmetries**.

Can we describe the ones in the interpretation of λ -terms?

$$\begin{array}{ccc} & \llbracket t \rrbracket & \\ \partial_l^{\llbracket t \rrbracket} \swarrow & & \searrow \partial_r^{\llbracket t \rrbracket} \\ \llbracket \Gamma \rrbracket & & \llbracket A \rrbracket \end{array} \in \mathbf{Gpd}$$

ITS for symmetries

Refinement types **value symmetries** and intersection types **multivalue symmetries**:

$$\alpha, \beta, \dots ::= \mathbf{tt} \mid \mathbf{ff} \mid \kappa \multimap \alpha$$

$$\kappa, \lambda, \dots ::= [\alpha_1, \dots, \alpha_n] \quad (n \in \mathbb{N})$$

$$\phi, \psi, \dots ::= \mathbf{id}_{\mathbf{tt}} \mid \mathbf{id}_{\mathbf{ff}} \mid \theta \multimap \phi$$

$$\theta, \zeta, \dots ::= (\pi, [\phi_1, \dots, \phi_n]) \quad (n \in \mathbb{N}, \pi \in \mathcal{S}_n)$$

ITS for symmetries

Resource symmetry contexts:

$$\Xi, \Xi', \dots \quad ::= \quad (x_i : \theta_i :: \kappa_i \Rightarrow \kappa'_i \triangleleft A_i)_{1 \leq i \leq n}$$

ITS for symmetries

Intersection type symmetry judgements

$$\Xi \vdash t : \phi :: \alpha \Rightarrow \alpha' \triangleleft A \quad \text{and} \quad \Xi \vdash_{\text{m}} t : \theta :: \kappa \Rightarrow \kappa' \triangleleft A$$

and their rules:

$$\text{(ITM-VAR)} \quad \frac{\phi :: \alpha \Rightarrow \alpha' \triangleleft A_i}{(\dots, x_i : (\text{id}_{\{1\}}, [\phi]) :: [\alpha] \Rightarrow [\alpha'] \triangleleft A_i, \dots) \vdash x_i : \phi :: \alpha \Rightarrow \alpha' \triangleleft A_i}$$

$$\text{(ITM-APP)} \quad \frac{\Xi \vdash t : (\theta \multimap \phi) :: (\kappa \multimap \beta) \Rightarrow (\kappa' \multimap \beta') \triangleleft A \rightarrow B \quad \Xi' \vdash_{\text{m}} u : \theta :: \kappa \Rightarrow \kappa'}{\Xi \oplus \Xi' \vdash t u : \phi :: \beta \Rightarrow \beta' \triangleleft B}$$

$$\text{(ITM-LAM)} \quad \frac{(\Xi, x : \theta :: \kappa \Rightarrow \kappa' \triangleleft A) \vdash t : \phi :: \beta \Rightarrow \beta' \triangleleft B}{\Xi \vdash \lambda x. t : (\theta \multimap \phi) :: (\kappa \multimap \beta) \Rightarrow (\kappa' \multimap \beta') \triangleleft A \rightarrow B}$$

$$\text{(ITM-INT)} \quad \frac{n \in \mathbb{N} \quad \sigma \in \mathcal{S}_n \quad \forall i \in \{1, \dots, n\}, \Xi_i \vdash t : \phi_i :: \alpha_i \Rightarrow \alpha'_i \triangleleft A}{(\sigma \otimes (\Xi_i)_{i \in [n]}) \vdash_{\text{m}} t : (\sigma, [\phi_1, \dots, \phi_n]) :: [\alpha_i]_{1 \leq i \leq n} \Rightarrow [\alpha'_{\sigma^{-1}(i)}]_{1 \leq i \leq n} \triangleleft A}$$

ITS for symmetries

Given a derivation $\Gamma \vdash t : A$, we get a groupoid $\llbracket t \rrbracket^{\mathbf{IT}}$:

- ▶ objects: derivations $\Theta \vdash t : \alpha \triangleleft A$
- ▶ symmetries: derivations $\Xi \vdash t : \phi :: \alpha \Rightarrow \alpha' \triangleleft A$
- ▶ composition of derivations as composition

Theorem (Clairambault, F.)

*The interpretation in **Thin** is described by the presented ITS, i.e.,*

$$\llbracket t \rrbracket \cong \llbracket t \rrbracket^{\mathbf{IT}}.$$

Conclusion

Thin spans

- ▶ a **quantitative semantic model** based on spans of groupoids
- ▶ provide a **proof-relevant extension** of the relational model **Rel**
- ▶ **syntactic description** given by a **rigid** intersection type system

Related works

Other quantitative and/or proof-relevant models:

- ▶ **Stable spans**

[Saunders-Evans and Winskel](#). “Event Structure Spans for Nondeterministic Dataflow”. 2007

- ▶ **Generalized species of structures**

[Fiore, Gambino, et al.](#) “The cartesian closed bicategory of generalised species of structures”. 2008

- ▶ **Template games**

[Melliès](#). “Template games and differential linear logic”. 2019

Syntactic descriptions through intersection type systems:

- ▶ [Tsukada, Asada, and Ong](#). “Generalised species of rigid resource terms”. 2017

- ▶ [Olimpieri](#). “Intersection type distributors”. 2021

The end

Any questions?