Comprehending stuff- and structure-types

Jürgen Koslowski

Department of Theoretical Computer Science
Technical University Braunschweig

CT 2007, Carvoeiro, Portugal, 2007-06-19

http://www.iti.cs.tu-bs.de/~koslowj/RESEARCH

01. Motivation: References

The initial motivation comes from two papers:

[BD] John C. Baez, James Dolan: From Finite Sets to Feynman Diagrams (April 2000), arXiv:math/0004133

[SB] Simon Byrne: On Groupoids and Stuff, Honors Thesis, MacQuarie University (November 2005), www.maths.mq.edu.au/~street/ByrneHons.pdf

as well as from various issues of John Baez's semi-regular column

"This Week's Find in Mathematical Physics"

available at http://math.ucr.edu/home/baez/weekXYZ.html, where intriguing applicatins of spans of groupoids are mentioned.

The main underlying reference is

[CE] André Joyal: Une théorie combinatoire des séries formelles, *Adv. Math.* **42** (1981), 1–82

The initial motivation comes from two papers:

[BD] John C. Baez, James Dolan: From Finite Sets to Feynman Diagrams (April 2000), arXiv:math/0004133

[SB] Simon Byrne: On Groupoids and Stuff, Honors Thesis, MacQuarie University (November 2005), www.maths.mq.edu.au/~street/ByrneHons.pdf

as well as from various issues of John Baez's semi-regular column

"This Week's Find in Mathematical Physics"

available at http://math.ucr.edu/home/baez/weekXYZ.html, where intriguing applicatins of spans of groupoids are mentioned.

The main underlying reference is

[CE] André Joyal: Une théorie combinatoire des séries formelles, *Adv. Math.* **42** (1981), 1–82

01. Motivation: References

The initial motivation comes from two papers:

[BD] John C. Baez, James Dolan: From Finite Sets to Feynman Diagrams (April 2000), arXiv:math/0004133

[SB] Simon Byrne: On Groupoids and Stuff, Honors Thesis, MacQuarie University (November 2005), www.maths.mq.edu.au/~street/ByrneHons.pdf

as well as from various issues of John Baez's semi-regular column

"This Week's Find in Mathematical Physics"

available at http://math.ucr.edu/home/baez/weekXYZ.html, where intriguing applicatins of spans of groupoids are mentioned.

The main underlying reference is

[CE] André Joyal: Une théorie combinatoire des séries formelles, *Adv. Math.* **42** (1981), 1–82

The initial motivation comes from two papers:

[BD] John C. Baez, James Dolan: From Finite Sets to Feynman Diagrams (April 2000), `arXiv:math/0004133`

[SB] Simon Byrne: On Groupoids and Stuff, Honors Thesis, MacQuarie University (November 2005), `www.maths.mq.edu.au/~street/ByrneHons.pdf`

as well as from various issues of John Baez's semi-regular column

"This Week's Find in Mathematical Physics"

available at `http://math.ucr.edu/home/baez/weekXYZ.html`, where intriguing applicatins of spans of groupoids are mentioned.

The main underlying reference is

[CE] André Joyal: Une théorie combinatoire des séries formelles, *Adv. Math.* **42** (1981), 1–82

The initial motivation comes from two papers:

[BD] John C. Baez, James Dolan: From Finite Sets to Feynman Diagrams (April 2000), `arXiv:math/0004133`

[SB] Simon Byrne: On Groupoids and Stuff, Honors Thesis, MacQuarie University (November 2005), `www.maths.mq.edu.au/~street/ByrneHons.pdf`

as well as from various issues of John Baez's semi-regular column

"This Week's Find in Mathematical Physics"

available at `http://math.ucr.edu/home/baez/weekXYZ.html`, where intriguing applicatins of spans of groupoids are mentioned.

The main underlying reference is

[CE] André Joyal: Une théorie combinatoire des séries formelles, *Adv. Math.* **42** (1981), 1–82

The initial motivation comes from two papers:

[BD] John C. Baez, James Dolan: From Finite Sets to Feynman Diagrams (April 2000), `arXiv:math/0004133`

[SB] Simon Byrne: On Groupoids and Stuff, Honors Thesis, MacQuarie University (November 2005), `www.maths.mq.edu.au/~street/ByrneHons.pdf`

as well as from various issues of John Baez's semi-regular column

"This Week's Find in Mathematical Physics"

available at `http://math.ucr.edu/home/baez/weekXYZ.html`, where intriguing applicatins of spans of groupoids are mentioned.

The main underlying reference is

[CE] André Joyal: Une théorie combinatoire des séries formelles, *Adv. Math.* **42** (1981), 1–82

- In order to "categorify" combinatorics, Joyal begins defining a species of structures F by assigning to each finite set n the set nF of F-structures that can live on n.
- Which types F of structures and which functions $n \xrightarrow{f} m$ between finite sets should be taken into consideration?
- To obtain well-behaved liftings $nF \xrightarrow{fF} mF$ for all types F of structures, restricting to bijections $n \xrightarrow{f} m$ seems appropriate. (Also, no need to consider contravariance!)

Definition

Thus a species of structures is just a functor from the groupoid $\mathbb{E}$ of finite sets and bijections to set.

- In order to “categorify” combinatorics, Joyal begins defining a species of structures F by assigning to each finite set n the set nF of F-structures that can live on n.
- Which types F of structures and which functions $n \xrightarrow{f} m$ between finite sets should be taken into consideration?
- To obtain well-behaved liftings $nF \xrightarrow{fF} mF$ for all types F of structures, restricting to bijections $n \xrightarrow{f} m$ seems appropriate. (Also, no need to consider contravariance!)

Definition

Thus a species of structures is just a functor from the groupoid $\mathbb{E}$ of finite sets and bijections to set.

- In order to "categorify" combinatorics, Joyal begins defining a species of structures F by assigning to each finite set n the set nF of F-structures that can live on n.
- Which types F of structures and which functions $n \xrightarrow{f} m$ between finite sets should be taken into consideration?
- To obtain well-behaved liftings $nF \xrightarrow{fF} mF$ for all types F of structures, restricting to bijections $n \xrightarrow{f} m$ seems appropriate. (Also, no need to consider contravariance!)

Definition

Thus a species of structures is just a functor from the groupoid $\mathbb{E}$ of finite sets and bijections to set.

- In order to "categorify" combinatorics, Joyal begins defining a species of structures F by assigning to each finite set n the set nF of F-structures that can live on n.
- Which types F of structures and which functions $n \xrightarrow{f} m$ between finite sets should be taken into consideration?
- To obtain well-behaved liftings $nF \xrightarrow{fF} mF$ for all types F of structures, restricting to bijections $n \xrightarrow{f} m$ seems appropriate. (Also, no need to consider contravariance!)

Definition

Thus a species of structures is just a functor from the groupoid $\boldsymbol{E}$ of finite sets and bijections to set.

- In order to “categorify” combinatorics, Joyal begins defining a species of structures F by assigning to each finite set n the set nF of F-structures that can live on n.
- Which types F of structures and which functions $n \xrightarrow{f} m$ between finite sets should be taken into consideration?
- To obtain well-behaved liftings $nF \xrightarrow{fF} mF$ for all types F of structures, restricting to bijections $n \xrightarrow{f} m$ seems appropriate. (Also, no need to consider contravariance!)

Definition

Thus a species of structures is just a functor from the groupoid $\boldsymbol{E}$ of finite sets and bijections to set.

For a species $\boldsymbol{E} \xrightarrow{F} \boldsymbol{set}$ Baez and Dolan construct a $\boldsymbol{gpd}$-morphism into $\boldsymbol{E}$ that "contains all the information in the [species]" F :

- Its domain is the value of a certain functor $\boldsymbol{set} \longrightarrow \boldsymbol{gpd}$ at 1 ;
- This functor is modeled on the analytic functor F^{a} associated with F , *i.e.*, the left Kan-extension of F along $\boldsymbol{E} \overset{J}{\hookrightarrow} \boldsymbol{set}$:

$$\begin{tikzcd} \boldsymbol{E} \arrow[r, "F"] \arrow[d, hook, "J"'] \arrow[r, phantom, shift right=4ex, "\Downarrow"] & \boldsymbol{set} \\ \boldsymbol{set} \arrow[ur, dashed, "F^{\mathrm{a}}"'] & \end{tikzcd}$$

suggests consideration of

where $\boldsymbol{set} \overset{I}{\hookrightarrow} \boldsymbol{gpd}$ is the other obvious inclusion.

For a species $\boldsymbol{E} \xrightarrow{F} \boldsymbol{set}$ Baez and Dolan construct a $\boldsymbol{gpd}$-morphism into $\boldsymbol{E}$ that "contains all the information in the [species]" F :

- Its domain is the value of a certain functor $\boldsymbol{set} \longrightarrow \boldsymbol{gpd}$ at 1 ;
- This functor is modeled on the analytic functor F^{a} associated with F , *i.e.*, the left Kan-extension of F along $\boldsymbol{E} \xhookrightarrow{J} \boldsymbol{set}$:

$$\begin{tikzcd} \boldsymbol{E} \arrow[r, "F"] \arrow[d, hook, "J"'] \arrow[r, phantom, shift right=4ex, "\Downarrow"] & \boldsymbol{set} \\ \boldsymbol{set} \arrow[ur, dashed, "F^{\mathrm{a}}"'] & \end{tikzcd}$$

suggests consideration of

$$\begin{tikzcd} \boldsymbol{E} \arrow[r, "F"] \arrow[d, hook, "J"'] & \boldsymbol{set} \arrow[d, hook, "I"] \\ \boldsymbol{set} \arrow[r] & \boldsymbol{gpd} \end{tikzcd}$$

where $\boldsymbol{set} \xhookrightarrow{I} \boldsymbol{gpd}$ is the other obvious inclusion.

For a species $\boldsymbol{E} \xrightarrow{F} \boldsymbol{set}$ Baez and Dolan construct a $\boldsymbol{gpd}$-morphism into $\boldsymbol{E}$ that "contains all the information in the [species]" F:

- Its domain is the value of a certain functor $\boldsymbol{set} \longrightarrow \boldsymbol{gpd}$ at 1;
- This functor is modeled on the analytic functor F^{a} associated with F, *i.e.*, the left Kan-extension of F along $\boldsymbol{E} \xhookrightarrow{J} \boldsymbol{set}$:

$$\begin{tikzcd} \boldsymbol{E} \arrow[r, "F"] \arrow[d, hook, "J"'] \arrow[dr, phantom, "\Downarrow"] & \boldsymbol{set} \\ \boldsymbol{set} \arrow[ur, dotted, "F^{\mathrm{a}}"'] & \end{tikzcd}$$

suggests consideration of

$$\begin{tikzcd} \boldsymbol{E} \arrow[r, "F"] \arrow[d, hook, "J"'] \arrow[dr, phantom, "\swarrow"] & \boldsymbol{set} \arrow[d, hook, "I"] \\ \boldsymbol{set} \arrow[r, dotted, "(FI)^{\mathrm{a}}"'] & \boldsymbol{gpd} \end{tikzcd}$$

where $\boldsymbol{set} \xhookrightarrow{I} \boldsymbol{gpd}$ is the other obvious inclusion.

03. The Baez-Dolan approach

For a species $\boldsymbol{E} \xrightarrow{F} \boldsymbol{set}$ Baez and Dolan construct a $\boldsymbol{gpd}$-morphism into $\boldsymbol{E}$ that "contains all the information in the [species]" F:

- Its domain is the value of a certain functor $\boldsymbol{set} \longrightarrow \boldsymbol{gpd}$ at 1;
- This functor is modeled on the analytic functor F^{a} associated with F, *i.e.*, the left Kan-extension of F along $\boldsymbol{E} \overset{J}{\hookrightarrow} \boldsymbol{set}$:

$$\begin{tikzcd}
\boldsymbol{E} \arrow[rr, "F"] \arrow[dr, hook, "J"'] & \arrow[d, phantom, "\Downarrow"] & \boldsymbol{set} \\
& \boldsymbol{set} \arrow[ur, dotted, "F^{\mathrm{a}}"'] &
\end{tikzcd}$$

suggests consideration of

$$\begin{tikzcd}
\boldsymbol{E} \arrow[rr, "F"] \arrow[d, hook, "J"'] & & \boldsymbol{set} \arrow[d, hook, "I"] \\
\boldsymbol{set} \arrow[rr, dotted, "(FI)^{\mathrm{a}}"'] & \arrow[u, phantom, "\Downarrow"] & \boldsymbol{gpd}
\end{tikzcd}$$

where $\boldsymbol{set} \overset{I}{\hookrightarrow} \boldsymbol{gpd}$ is the other obvious inclusion.

For a species $\boldsymbol{E} \xrightarrow{F} \boldsymbol{set}$ Baez and Dolan construct a $\boldsymbol{gpd}$-morphism into $\boldsymbol{E}$ that "contains all the information in the [species]" F:

- Its domain is the value of a certain functor $\boldsymbol{set} \longrightarrow \boldsymbol{gpd}$ at 1;
- This functor is modeled on the analytic functor F^{a} associated with F, *i.e.*, the left Kan-extension of F along $\boldsymbol{E} \xhookrightarrow{J} \boldsymbol{set}$:

$$\begin{tikzcd} \boldsymbol{E} \arrow[r, "F"] \arrow[dr, hook, "J"'] & \boldsymbol{set} \arrow[d, phantom, "\Downarrow"] \\ & \boldsymbol{set} \arrow[u, dotted, "F^{\mathrm{a}}"'] \end{tikzcd}$$

suggests consideration of

$$\begin{tikzcd} \boldsymbol{E} \arrow[r, "F"] \arrow[d, hook, "J"'] & \boldsymbol{set} \arrow[d, hook, "I"] \arrow[dl, phantom, "\Swarrow"] \\ \boldsymbol{set} \arrow[r, dotted, "(FI)^{\mathrm{a}}"'] & \boldsymbol{gpd} \end{tikzcd}$$

where $\boldsymbol{set} \xhookrightarrow{I} \boldsymbol{gpd}$ is the other obvious inclusion.

03. The Baez-Dolan approach

For a species $\boldsymbol{E} \xrightarrow{F} \boldsymbol{set}$ Baez and Dolan construct a $\boldsymbol{gpd}$-morphism into $\boldsymbol{E}$ that "contains all the information in the [species]" F:

- Its domain is the value of a certain functor $\boldsymbol{set} \longrightarrow \boldsymbol{gpd}$ at 1;
- This functor is modeled on the analytic functor F^{a} associated with F, *i.e.*, the left Kan-extension of F along $\boldsymbol{E} \overset{J}{\hookrightarrow} \boldsymbol{set}$:

$$\begin{tikzcd}
\boldsymbol{E} \arrow[rr, "F"] \arrow[dr, hook, "J"'] \arrow[rr, phantom, shift right=6, "\Downarrow"] & & \boldsymbol{set} \\
& \boldsymbol{set} \arrow[ur, dotted, "F^{\mathrm{a}}"'] &
\end{tikzcd}$$

suggests consideration of

$$\begin{tikzcd}
\boldsymbol{E} \arrow[r, "F"] \arrow[d, hook, "J"'] \arrow[dr, phantom, "\Swarrow"] & \boldsymbol{set} \arrow[d, hook, "I"] \\
\boldsymbol{set} \arrow[r, dotted, "(FI)^{\mathrm{a}}"'] & \boldsymbol{gpd}
\end{tikzcd}$$

where $\boldsymbol{set} \overset{I}{\hookrightarrow} \boldsymbol{gpd}$ is the other obvious inclusion.

The exponential generating function for F

$$N \xrightarrow{|F|} N, \quad x \mapsto \sum_{n \in N} (|nF| \cdot x^n)/n!$$

provides the template for both analytic functors F^{a} and $(FI)^{\mathrm{a}}$ (now we think of $n!$ as the permutation group of n):

$$\mathbf{set} \xrightarrow{F^{\mathrm{a}}} \mathbf{set}, \quad X \mapsto \int^{u \in \mathbf{E}} uF \times X^u \cong \sum_{n \in N} (nF \times X^n)/n!$$

$$\mathbf{set} \xrightarrow{(FI)^{\mathrm{a}}} \mathbf{gpd}, \quad X \mapsto \int^{u \in \mathbf{E}} uF \times X^u \cong \sum_{n \in N} (nF \times X^n)/\!\!/n!$$

The “weak quotient” $/\!\!/$ is just a glueing construction!
$(FI)^{\mathrm{a}}$ maps X to the groupoid of X-colored F-structured sets.

The exponential generating function for F

$$N \xrightarrow{|F|} N, \quad x \mapsto \sum_{n \in N} (|nF| \cdot x^n)/n!$$

provides the template for both analytic functors F^{a} and $(FI)^{\mathrm{a}}$ (now we think of $n!$ as the permutation group of n):

$$\boldsymbol{set} \xrightarrow{F^{\mathrm{a}}} \boldsymbol{set}, \quad X \mapsto \int^{u \in \boldsymbol{E}} uF \times X^u \cong \sum_{n \in N} (nF \times X^n)/n!$$

$$\boldsymbol{set} \xrightarrow{(FI)^{\mathrm{a}}} \boldsymbol{gpd}, \quad X \mapsto \int^{u \in \boldsymbol{E}} uF \times X^u \cong \sum_{n \in N} (nF \times X^n)/\!\!/n!$$

The "weak quotient" $/\!\!/$ is just a glueing construction!
$(FI)^{\mathrm{a}}$ maps X to the groupoid of X-colored F-structured sets.

The exponential generating function for F

$$\boldsymbol{N} \xrightarrow{|F|} \boldsymbol{N}, \quad x \mapsto \sum_{n \in \boldsymbol{N}} (|nF| \cdot x^n)/n!$$

provides the template for both analytic functors F^{a} and $(FI)^{\mathrm{a}}$ (now we think of $n!$ as the permutation group of n):

$$\boldsymbol{set} \xrightarrow{F^{\mathrm{a}}} \boldsymbol{set}, \quad X \mapsto \int^{u \in \boldsymbol{E}} uF \times X^u \cong \sum_{n \in \boldsymbol{N}} (nF \times X^n)/n!$$

$$\boldsymbol{set} \xrightarrow{(FI)^{\mathrm{a}}} \boldsymbol{gpd}, \quad X \mapsto \int^{u \in \boldsymbol{E}} uF \times X^u \cong \sum_{n \in \boldsymbol{N}} (nF \times X^n)/\!\!/n!$$

The "weak quotient" $/\!\!/$ is just a glueing construction!
$(FI)^{\mathrm{a}}$ maps X to the groupoid of X-colored F-structured sets.

04. The Baez-Dolan approach, continued

The exponential generating function for F

$$\mathbf{N} \xrightarrow{|F|} \mathbf{N}, \quad x \mapsto \sum_{n \in \mathbf{N}} (|nF| \cdot x^n)/n!$$

provides the template for both analytic functors F^{a} and $(FI)^{\mathrm{a}}$ (now we think of $n!$ as the permutation group of n):

$$\boldsymbol{set} \xrightarrow{F^{\mathrm{a}}} \boldsymbol{set}, \quad X \mapsto \int^{u \in \boldsymbol{E}} uF \times X^u \cong \sum_{n \in \mathbf{N}} (nF \times X^n)/n!$$

$$\boldsymbol{set} \xrightarrow{(FI)^{\mathrm{a}}} \boldsymbol{gpd}, \quad X \mapsto \int^{u \in \boldsymbol{E}} uF \times X^u \cong \sum_{n \in \mathbf{N}} (nF \times X^n) /\!\!/ n!$$

The "weak quotient" $/\!\!/$ is just a glueing construction!

$(FI)^{\mathrm{a}}$ maps X to the groupoid of X-colored F-structured sets.

The exponential generating function for F

$$\boldsymbol{N} \xrightarrow{|F|} \boldsymbol{N}, \quad x \mapsto \sum_{n \in \boldsymbol{N}} (|nF| \cdot x^n)/n!$$

provides the template for both analytic functors F^{a} and $(FI)^{\mathrm{a}}$ (now we think of $n!$ as the permutation group of n):

$$\boldsymbol{set} \xrightarrow{F^{\mathrm{a}}} \boldsymbol{set}, \quad X \mapsto \int^{u \in \boldsymbol{E}} uF \times X^u \cong \sum_{n \in \boldsymbol{N}} (nF \times X^n)/n!$$

$$\boldsymbol{set} \xrightarrow{(FI)^{\mathrm{a}}} \boldsymbol{gpd}, \quad X \mapsto \int^{u \in \boldsymbol{E}} uF \times X^u \cong \sum_{n \in \boldsymbol{N}} (nF \times X^n) /\!\!/ n!$$

The "weak quotient" $/\!\!/$ is just a glueing construction!
$(FI)^{\mathrm{a}}$ maps X to the groupoid of X-colored F-structured sets.

- Baez and Dolan then generalize the obvious forgetful functor $1(FI)^{\mathrm{a}} \longrightarrow \boldsymbol{E}$ to arbitrary $\boldsymbol{gpd}$-morphisms into $\boldsymbol{E}$, subject to an unspecified finiteness condition; these are called stuff types.
- Byrne [SB] describes stuff (and structure) types as "working in the opposite direction" of species, by forgetting the structure.
- He then proceeds to construct a stuff type from a species by a glueing construction.
- Conversely, from a stuff type he constructs a "fibre functor" $\boldsymbol{E} \longrightarrow \boldsymbol{gpd}$, and then characterizes those stuff types that will indeed produce a species in this fashion as the faithful ones:

$$\begin{array}{rccl} \text{functor} & \mathcal{G} \longrightarrow \boldsymbol{E} & : & \text{stuff type} \\ \text{faithful functor} & \mathcal{G} \longrightarrow \boldsymbol{E} & : & \text{structure type} \\ \text{full and faithful functor} & \mathcal{G} \longrightarrow \boldsymbol{E} & : & \text{property type} \end{array}$$

- Baez and Dolan then generalize the obvious forgetful functor $1(FI)^{\mathrm{a}} \longrightarrow \boldsymbol{E}$ to arbitrary $\boldsymbol{gpd}$-morphisms into $\boldsymbol{E}$, subject to an unspecified finiteness condition; these are called stuff types.
- Byrne [SB] describes stuff (and structure) types as "working in the opposite direction" of species, by forgetting the structure.
- He then proceeds to construct a stuff type from a species by a glueing construction.
- Conversely, from a stuff type he constructs a "fibre functor" $\boldsymbol{E} \longrightarrow \boldsymbol{gpd}$, and then characterizes those stuff types that will indeed produce a species in this fashion as the faithful ones:

functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	stuff type
faithful functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	structure type
full and faithful functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	property type

05. The Baez-Dolan approach and Byrne's view

- Baez and Dolan then generalize the obvious forgetful functor $1(FI)^{\mathrm{a}} \longrightarrow \boldsymbol{E}$ to arbitrary $\boldsymbol{gpd}$-morphisms into $\boldsymbol{E}$, subject to an unspecified finiteness condition; these are called stuff types.
- Byrne [SB] describes stuff (and structure) types as "working in the opposite direction" of species, by forgetting the structure.
- He then proceeds to construct a stuff type from a species by a glueing construction.
- Conversely, from a stuff type he constructs a "fibre functor" $\boldsymbol{E} \longrightarrow \boldsymbol{gpd}$, and then characterizes those stuff types that will indeed produce a species in this fashion as the faithful ones:

functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	stuff type
faithful functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	structure type
full and faithful functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	property type

- Baez and Dolan then generalize the obvious forgetful functor $1(FI)^{\mathrm{a}} \longrightarrow \boldsymbol{E}$ to arbitrary $\boldsymbol{gpd}$-morphisms into $\boldsymbol{E}$, subject to an unspecified finiteness condition; these are called stuff types.
- Byrne [SB] describes stuff (and structure) types as "working in the opposite direction" of species, by forgetting the structure.
- He then proceeds to construct a stuff type from a species by a glueing construction.
- Conversely, from a stuff type he constructs a "fibre functor" $\boldsymbol{E} \longrightarrow \boldsymbol{gpd}$, and then characterizes those stuff types that will indeed produce a species in this fashion as the faithful ones:

functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	stuff type
faithful functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	structure type
full and faithful functor	$\mathcal{G} \longrightarrow \boldsymbol{E}$	:	property type

05. The Baez-Dolan approach and Byrne's view

- Baez and Dolan then generalize the obvious forgetful functor $1(FI)^{\mathrm{a}} \longrightarrow \boldsymbol{E}$ to arbitrary $\boldsymbol{gpd}$-morphisms into $\boldsymbol{E}$, subject to an unspecified finiteness condition; these are called stuff types.
- Byrne [SB] describes stuff (and structure) types as "working in the opposite direction" of species, by forgetting the structure.
- He then proceeds to construct a stuff type from a species by a glueing construction.
- Conversely, from a stuff type he constructs a "fibre functor" $\boldsymbol{E} \longrightarrow \boldsymbol{gpd}$, and then characterizes those stuff types that will indeed produce a species in this fashion as the faithful ones:

$$\begin{array}{rccl} \text{functor} & \mathcal{G} \longrightarrow \boldsymbol{E} & : & \text{stuff type} \\ \text{faithful functor} & \mathcal{G} \longrightarrow \boldsymbol{E} & : & \text{structure type} \\ \text{full and faithful functor} & \mathcal{G} \longrightarrow \boldsymbol{E} & : & \text{property type} \end{array}$$

06. Labeled transition systems (LTSs): recap

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}\,(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$
\begin{array}{cl}
\boldsymbol{Q} \xrightarrow{L := \langle !, \ell \rangle} \boldsymbol{X} & \text{(graph morphism)} \\
\hline\hline
X \xleftarrow{\ell} Q_1 \overset{s}{\underset{t}{\rightrightarrows}} Q_0 & \\
\hline\hline
X \xleftarrow{\ell} Q_1 \xrightarrow{\langle s, t \rangle} Q_0 \times Q_0 & \\
\hline\hline
X \xrightarrow{L_\circ} \langle Q_0, Q_0 \rangle \boldsymbol{spn} & \\
\hline\hline
\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} & \text{(graph morphism)}
\end{array}
$$

This also works for general graphs $\boldsymbol{X}$.

06. Labeled transition systems (LTSs): recap

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$
\begin{array}{cl}
Q \xrightarrow{L:=\langle !,\ell\rangle} \boldsymbol{X} & \text{(graph morphism)} \\
\hline\hline
X \xleftarrow{\ell} Q_1 \overset{s}{\underset{t}{\rightrightarrows}} Q_0 & \\
\hline\hline
X \xleftarrow{\ell} Q_1 \xrightarrow{\langle s,t\rangle} Q_0 \times Q_0 & \\
\hline\hline
X \xrightarrow{L_\circ} \langle Q_0, Q_0\rangle \boldsymbol{spn} & \\
\hline\hline
\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} & \text{(graph morphism)}
\end{array}
$$

This also works for general graphs $\boldsymbol{X}$.

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$\boldsymbol{Q} \xrightarrow{L := \langle !, \ell \rangle} \boldsymbol{X} \qquad \text{(graph morphism)}$$

$$\overline{\overline{\quad X \xleftarrow{\ell} Q_1 \overset{s}{\underset{t}{\rightrightarrows}} Q_0 \quad}}$$

$$\overline{\overline{\quad X \xleftarrow{\ell} Q_1 \xrightarrow{\langle s,t \rangle} Q_0 \times Q_0 \quad}}$$

$$\overline{\overline{\quad X \xrightarrow{L_\circ} \langle Q_0, Q_0 \rangle \boldsymbol{spn} \quad}}$$

$$\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} \qquad \text{(graph morphism)}$$

This also works for general graphs $\boldsymbol{X}$.

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}\,(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$\begin{array}{c} \boldsymbol{Q} \xrightarrow{L := \langle !, \ell \rangle} \boldsymbol{X} \qquad \text{(graph morphism)} \\ \hline\hline X \xleftarrow{\ell} Q_1 \overset{s}{\underset{t}{\rightrightarrows}} Q_0 \\ \hline\hline X \xleftarrow{\ell} Q_1 \xrightarrow{\langle s, t \rangle} Q_0 \times Q_0 \\ \hline\hline X \xrightarrow{L_\circ} \langle Q_0, Q_0 \rangle \boldsymbol{spn} \\ \hline\hline \boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} \qquad \text{(graph morphism)} \end{array}$$

This also works for general graphs $\boldsymbol{X}$.

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}\,(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$\boldsymbol{Q} \xrightarrow{L := \langle !, \ell \rangle} \boldsymbol{X} \qquad \text{(graph morphism)}$$
$$\overline{\overline{\qquad\qquad}}$$
$$X \xleftarrow{\ \ell\ } Q_1 \overset{s}{\underset{t}{\rightrightarrows}} Q_0$$
$$\overline{\overline{\qquad\qquad}}$$
$$X \xleftarrow{\ \ell\ } Q_1 \xrightarrow{\langle s, t \rangle} Q_0 \times Q_0$$
$$\overline{\overline{\qquad\qquad}}$$
$$X \xrightarrow{L_\circ} \langle Q_0, Q_0 \rangle\, \boldsymbol{spn}$$
$$\overline{\overline{\qquad\qquad}}$$
$$\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} \qquad \text{(graph morphism)}$$

This also works for general graphs $\boldsymbol{X}$.

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$\boldsymbol{Q} \xrightarrow{L := \langle !, \ell \rangle} \boldsymbol{X} \qquad \text{(graph morphism)}$$

$$\overline{\overline{\quad X \xleftarrow{\ \ell\ } Q_1 \overset{s}{\underset{t}{\rightrightarrows}} Q_0 \quad}}$$

$$\overline{\overline{\quad X \xleftarrow{\ \ell\ } Q_1 \xrightarrow{\langle s,t \rangle} Q_0 \times Q_0 \quad}}$$

$$\overline{\overline{\quad X \xrightarrow{L_\circ} \langle Q_0, Q_0 \rangle \boldsymbol{spn} \quad}}$$

$$\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} \qquad \text{(graph morphism)}$$

This also works for general graphs $\boldsymbol{X}$.

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$\boldsymbol{Q} \xrightarrow{L:=\langle !,\ell\rangle} \boldsymbol{X} \qquad \text{(graph morphism)}$$

$$\overline{\overline{\qquad\qquad}}$$

$$X \xleftarrow{\ell} Q_1 \underset{t}{\overset{s}{\rightrightarrows}} Q_0$$

$$\overline{\overline{\qquad\qquad}}$$

$$X \xleftarrow{\ell} Q_1 \xrightarrow{\langle s,t\rangle} Q_0 \times Q_0$$

$$\overline{\overline{\qquad\qquad}}$$

$$X \xrightarrow{L_\circ} \langle Q_0, Q_0\rangle \boldsymbol{spn}$$

$$\overline{\overline{\qquad\qquad}}$$

$$\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} \qquad \text{(graph morphism)}$$

This also works for general graphs $\boldsymbol{X}$.

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}\,(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$\boldsymbol{Q} \xrightarrow{L:=\langle !,\ell\rangle} \boldsymbol{X} \qquad \text{(graph morphism)}$$
$$\overline{\overline{}}$$
$$X \xleftarrow{\ell} Q_1 \underset{t}{\overset{s}{\rightrightarrows}} Q_0$$
$$\overline{\overline{}}$$
$$X \xleftarrow{\ell} Q_1 \xrightarrow{\langle s,t\rangle} Q_0 \times Q_0$$
$$\overline{\overline{}}$$
$$X \xrightarrow{L_\circ} \langle Q_0, Q_0\rangle\, \boldsymbol{spn}$$
$$\overline{\overline{}}$$
$$\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} \qquad \text{(graph morphism)}$$

This also works for general graphs $\boldsymbol{X}$.

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}\,(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$\boldsymbol{Q} \xrightarrow{L:=\langle !,\ell\rangle} \boldsymbol{X} \qquad \text{(graph morphism)}$$

$$\overline{\overline{}}$$

$$X \xleftarrow{\ell} Q_1 \overset{s}{\underset{t}{\rightrightarrows}} Q_0$$

$$\overline{\overline{}}$$

$$X \xleftarrow{\ell} Q_1 \xrightarrow{\langle s,t\rangle} Q_0 \times Q_0$$

$$\overline{\overline{}}$$

$$X \xrightarrow{L_\circ} \langle Q_0, Q_0\rangle\,\boldsymbol{spn}$$

$$\overline{\overline{}}$$

$$\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn} \qquad \text{(graph morphism)}$$

This also works for general graphs $\boldsymbol{X}$.

The picture emerging so far bears striking resemblance to the different views of (traditional) LTSs over a label graph $\boldsymbol{X}$,

- as processes, *i.e.*, (faithful) graph-morphisms into $\boldsymbol{X}$, or
- as systems, *i.e.*, graph-morphisms $\boldsymbol{X} \longrightarrow \boldsymbol{spn}(\boldsymbol{rel})$.

Viewing labels in a *set* X as arrows of a single-node graph we get

$$\dfrac{\dfrac{\dfrac{\dfrac{\boldsymbol{Q} \xrightarrow{L := \langle !, \ell \rangle} \boldsymbol{X}}{X \xleftarrow{\ell} Q_1 \overset{s}{\underset{t}{\rightrightarrows}} Q_0}}{X \xleftarrow{\ell} Q_1 \xrightarrow{\langle s,t \rangle} Q_0 \times Q_0}}{X \xrightarrow{L_\circ} \langle Q_0, Q_0 \rangle \boldsymbol{spn}}}{\boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn}}$$

(graph morphism) — top line; (graph morphism) — bottom line

This also works for general graphs $\boldsymbol{X}$.

Definition

$grph$ and $Grph$ denote the (bi)categories of small, respectively, locally small graphs and graph morphisms. These have non-full sub(bi)categories cat and Cat, respectively.

We call a $Grph$-morphism fiber-small, if each fibre is small ;-)

Theorem

Every graph X induces an essentially bijective correspondence between fiber-small processes $Q \longrightarrow X$ and systems $X \longrightarrow spn$.

If X is a category, this restricts to fiber-small functors $Q \longrightarrow X$, respectively, lax functors $X \longrightarrow spn$.

07. Graph comprehension at the object level

Definition

$\boldsymbol{grph}$ and $\boldsymbol{Grph}$ denote the (bi)categories of small, respectively, locally small graphs and graph morphisms. These have non-full sub(bi)categories $\boldsymbol{cat}$ and $\boldsymbol{Cat}$, respectively.

We call a $\boldsymbol{Grph}$-morphism fiber-small, if each fibre is small ;-)

Theorem

Every graph $\boldsymbol{X}$ induces an essentially bijective correspondence between fiber-small processes $\boldsymbol{Q} \longrightarrow \boldsymbol{X}$ and systems $\boldsymbol{X} \longrightarrow \boldsymbol{spn}$.

If $\boldsymbol{X}$ is a category, this restricts to fiber-small functors $\boldsymbol{Q} \longrightarrow \boldsymbol{X}$, respectively, lax functors $\boldsymbol{X} \longrightarrow \boldsymbol{spn}$.

Definition

$\boldsymbol{grph}$ and $\boldsymbol{Grph}$ denote the (bi)categories of small, respectively, locally small graphs and graph morphisms. These have non-full sub(bi)categories $\boldsymbol{cat}$ and $\boldsymbol{Cat}$, respectively.

We call a $\boldsymbol{Grph}$-morphism fiber-small, if each fibre is small ;-)

Theorem

Every graph $\boldsymbol{X}$ induces an essentially bijective correspondence between fiber-small processes $\boldsymbol{Q} \longrightarrow \boldsymbol{X}$ and systems $\boldsymbol{X} \longrightarrow \boldsymbol{spn}$.

If $\boldsymbol{X}$ is a category, this restricts to fiber-small functors $\boldsymbol{Q} \longrightarrow \boldsymbol{X}$, respectively, lax functors $\boldsymbol{X} \longrightarrow \boldsymbol{spn}$.

07. Graph comprehension at the object level

Definition

$grph$ and $Grph$ denote the (bi)categories of small, respectively, locally small graphs and graph morphisms. These have non-full sub(bi)categories cat and Cat, respectively.

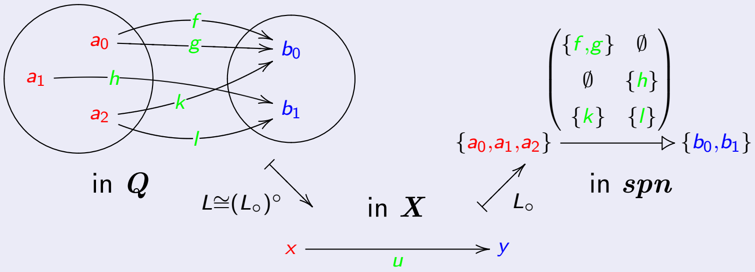
We call a $Grph$-morphism fiber-small, if each fibre is small ;-)

Theorem

Every graph X induces an essentially bijective correspondence between fiber-small processes $Q \longrightarrow X$ and systems $X \longrightarrow spn$.

If X is a category, this restricts to fiber-small functors $Q \longrightarrow X$, respectively, lax functors $X \longrightarrow spn$.

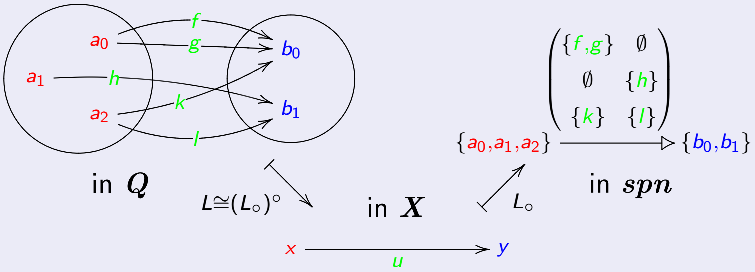
Sketch of proof



- $L_\circ$ is obtained from L by taking inverse images.
- In the other direction one employs disjoint unions.
- If $\boldsymbol{X}$ is a category, laxness of $L_\circ$ equips $\boldsymbol{Q}$ with units and a composition that are preserved by L, and vica versa. □

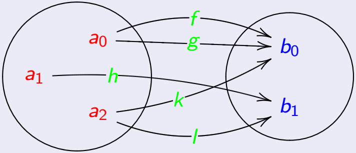
08. Graph comprehension at the object level

Sketch of proof



- $L_\circ$ is obtained from L by taking inverse images.
- In the other direction one employs disjoint unions.
- If $\boldsymbol{X}$ is a category, laxness of $L_\circ$ equips $\boldsymbol{Q}$ with units and a composition that are preserved by L, and vica versa.

Sketch of proof



in $\boldsymbol{Q}$

$$\{a_0,a_1,a_2\} \xrightarrow{\begin{pmatrix} \{f,g\} & \emptyset \\ \emptyset & \{h\} \\ \{k\} & \{l\} \end{pmatrix}} \{b_0,b_1\}$$

in $\boldsymbol{spn}$

$$\boldsymbol{Q} \xrightarrow{L \cong (L_\circ)^\circ} \boldsymbol{X} \xrightarrow{L_\circ} \boldsymbol{spn}$$

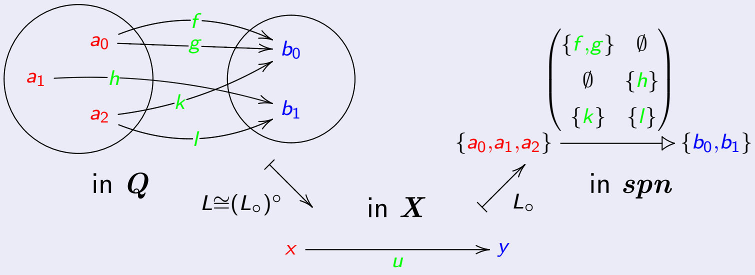
in $\boldsymbol{X}$

$$x \xrightarrow{u} y$$

- $L_\circ$ is obtained from L by taking inverse images.
- In the other direction one employs disjoint unions.
- If $\boldsymbol{X}$ is a category, laxness of $L_\circ$ equips $\boldsymbol{Q}$ with units and a composition that are preserved by L, and vica versa.

08. Graph comprehension at the object level

Sketch of proof



- $L_\circ$ is obtained from L by taking inverse images.
- In the other direction one employs disjoint unions.
- If X is a category, laxness of $L_\circ$ equips Q with units and a composition that are preserved by L, and vica versa. $\square$

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in spn

$$\begin{tikzcd} xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "\Nearrow u\varphi"] & xL \arrow[d, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL \end{tikzcd}$$

Modules (mixed assoc.)

$L \overset{\pi}{\Longrightarrow} M$: in spn

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \\ yL \arrow[r, "v\pi"'] & zM \end{tikzcd}$$

Lax transforms

$M \overset{\sigma}{\Longrightarrow} N$: in spn

$$\begin{tikzcd} yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\Swarrow v\sigma"] & yN \arrow[d, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \overset{\varphi\#}{\Longrightarrow} L$ and $M \overset{\sigma^\#}{\Longrightarrow} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in spn

$$\begin{tikzcd} xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "\Nearrow u\varphi"] & xL \arrow[d, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL \end{tikzcd}$$

Modules (mixed assoc.)

$L \overset{\pi}{\Longrightarrow} M$: in spn

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi"] & yM \arrow[d, "vM"] \arrow[dl, phantom, "\Swarrow"] \\ yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Nearrow"] & zM \end{tikzcd}$$

Lax transforms

$M \overset{\sigma}{\Longrightarrow} N$: in spn

$$\begin{tikzcd} yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\Swarrow v\sigma"] & yN \arrow[d, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \overset{\varphi\#}{\Longrightarrow} L$ and $M \overset{\sigma^{\#}}{\Longrightarrow} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in spn

$$\begin{tikzcd}
xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "\overset{u\varphi}{\Nearrow}"] & xL \arrow[d, "uL"] \\
yK \arrow[r, "y\varphi"'] & yL
\end{tikzcd}$$

Modules (mixed assoc.)

$L \overset{\pi}{\Longrightarrow} M$: in spn

$$\begin{tikzcd}
& xL \arrow[r, "u\pi"] \arrow[dl, "uL"'] \arrow[d, "(u;v)\pi"] & yM \arrow[d, "vM"] \arrow[dl, phantom, "\Swarrow"] \\
yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Nearrow"] & zM &
\end{tikzcd}$$

Lax transforms

$M \overset{\sigma}{\Longrightarrow} N$: in spn

$$\begin{tikzcd}
yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\overset{v\sigma}{\Swarrow}"] & yN \arrow[d, "vN"] \\
zM \arrow[r, "z\sigma"'] & zN
\end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \overset{\varphi\#}{\Longrightarrow} L$ and $M \overset{\sigma^{\#}}{\Longrightarrow} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

09. Three important types of 1-cells for systems $X \longrightarrow \boldsymbol{spn}$

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in $\boldsymbol{spn}$

$$\begin{tikzcd} xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "\nearrow u\varphi"] & xL \arrow[d, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL \end{tikzcd}$$

Modules (mixed assoc.)

$L \overset{\pi}{\Longrightarrow} M$: in $\boldsymbol{spn}$

$$\begin{tikzcd} & xL \arrow[r, "u\pi"] \arrow[dl, "uL"'] \arrow[d, "(u;v)\pi"] & yM \arrow[dl, "vM"] \\ yL \arrow[r, "v\pi"'] & zM & \end{tikzcd}$$

Lax transforms

$M \overset{\sigma}{\Longrightarrow} N$: in $\boldsymbol{spn}$

$$\begin{tikzcd} yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\swarrow v\sigma"] & yN \arrow[d, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \overset{\varphi\#}{\Longrightarrow} L$ and $M \overset{\sigma^{\#}}{\Longrightarrow} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

Oplax map-transforms

$K \xRightarrow{\varphi} L$: in spn

$$\begin{tikzcd} xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "u\varphi \Nearrow"] & xL \arrow[d, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL \end{tikzcd}$$

Modules (mixed assoc.)

$L \xRightarrow{\pi} M$: in spn

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \arrow[dl, phantom, "\Swarrow"] \\ yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Nearrow"] & zM \end{tikzcd}$$

Lax transforms

$M \xRightarrow{\sigma} N$: in spn

$$\begin{tikzcd} yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\Swarrow v\sigma"] & yN \arrow[d, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \xRightarrow{\varphi\#} L$ and $M \xRightarrow{\sigma^{\#}} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in spn

$$\begin{tikzcd} & xK \arrow[r, "x\varphi"] \arrow[dl, "uK"'] & xL \arrow[dl, "uL"] \\ yK \arrow[r, "y\varphi"'] \arrow[ur, phantom, "\Nearrow u\varphi"] & yL & \end{tikzcd}$$

Modules (mixed assoc.)

$L \overset{\pi}{\Longrightarrow} M$: in spn

$$\begin{tikzcd} & xL \arrow[r, "u\pi"] \arrow[dl, "uL"'] \arrow[d, "(u;v)\pi"] & yM \arrow[dl, "vM"] \arrow[l, phantom, "\Swarrow"] \\ yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Nearrow"] & zM & \end{tikzcd}$$

Lax transforms

$M \overset{\sigma}{\Longrightarrow} N$: in spn

$$\begin{tikzcd} & yM \arrow[r, "y\sigma"] \arrow[dl, "vM"'] & yN \arrow[dl, "vN"] \\ zM \arrow[r, "z\sigma"'] \arrow[ur, phantom, "\Swarrow v\sigma"] & zN & \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \overset{\varphi\#}{\Longrightarrow} L$ and $M \overset{\sigma^{\#}}{\Longrightarrow} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

09. Three important types of 1-cells for systems $X \longrightarrow \boldsymbol{spn}$

Oplax map-transforms

$K \xRightarrow{\varphi} L$: in $\boldsymbol{spn}$

$$\begin{tikzcd} & xK \arrow[r, "x\varphi"] \arrow[dl, "uK"'] \arrow[d, phantom, "\Nearrow u\varphi"] & xL \arrow[dl, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL & \end{tikzcd}$$

Modules (mixed assoc.)

$L \xRightarrow{\pi} M$: in $\boldsymbol{spn}$

$$\begin{tikzcd} & xL \arrow[r, "u\pi"] \arrow[dl, "uL"'] \arrow[d, "(u;v)\pi"] \arrow[dl, phantom, shift right=4, "\Nearrow"] & yM \arrow[dl, "vM"] \arrow[dl, phantom, shift left=4, "\Swarrow"] \\ yL \arrow[r, "v\pi"'] & zM & \end{tikzcd}$$

Lax transforms

$M \xRightarrow{\sigma} N$: in $\boldsymbol{spn}$

$$\begin{tikzcd} & yM \arrow[r, "y\sigma"] \arrow[dl, "vM"'] \arrow[d, phantom, "\Swarrow v\sigma"] & yN \arrow[dl, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN & \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \xRightarrow{\varphi\#} L$ and $M \xRightarrow{\sigma^{\#}} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in $\boldsymbol{spn}$

$$\begin{tikzcd} xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "u\varphi \Nearrow"] & xL \arrow[d, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL \end{tikzcd}$$

Modules (mixed assoc.)

$L \overset{\pi}{\Longrightarrow} M$: in $\boldsymbol{spn}$

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \arrow[dl, phantom, "\Swarrow"] \\ yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Nearrow"] & zM \end{tikzcd}$$

Lax transforms

$M \overset{\sigma}{\Longrightarrow} N$: in $\boldsymbol{spn}$

$$\begin{tikzcd} yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\Swarrow v\sigma"] & yN \arrow[d, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \overset{\varphi\#}{\Longrightarrow} L$ and $M \overset{\sigma^{\#}}{\Longrightarrow} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

09. Three important types of 1-cells for systems $\boldsymbol{X} \longrightarrow \boldsymbol{spn}$

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in $\boldsymbol{spn}$

$$\begin{tikzcd} xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "u\varphi\Nearrow"] & xL \arrow[d, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL \end{tikzcd}$$

Modules (mixed assoc.)

$L \overset{\pi}{\Longrightarrow} M$: in $\boldsymbol{spn}$

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \arrow[dl, phantom, "\Swarrow", near start] \\ yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Nearrow", near start] & zM \end{tikzcd}$$

Lax transforms

$M \overset{\sigma}{\Longrightarrow} N$: in $\boldsymbol{spn}$

$$\begin{tikzcd} yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\Swarrow v\sigma"] & yN \arrow[d, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category $\boldsymbol{X}$ such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \overset{\varphi\#}{\Longrightarrow} L$ and $M \overset{\sigma^\#}{\Longrightarrow} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

09. Three important types of 1-cells for systems $X \longrightarrow spn$

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in spn

$$\begin{tikzcd}
xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "u\varphi \Nearrow"] & xL \arrow[d, "uL"] \\
yK \arrow[r, "y\varphi"'] & yL
\end{tikzcd}$$

Modules (mixed assoc.)

$L \xRightarrow{\pi} M$: in spn

$$\begin{tikzcd}
xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \\
yL \arrow[r, "v\pi"'] & zM
\end{tikzcd}$$

Lax transforms

$M \xRightarrow{\sigma} N$: in spn

$$\begin{tikzcd}
yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\Swarrow v\sigma"] & yN \arrow[d, "vN"] \\
zM \arrow[r, "z\sigma"'] & zN
\end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \xRightarrow{\varphi\#} L$ and $M \xRightarrow{\sigma^{\#}} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

09. Three important types of 1-cells for systems $X \longrightarrow \mathbf{spn}$

Oplax map-transforms

$K \xRightarrow{\varphi} L$: in $\mathbf{spn}$

$$\begin{tikzcd} xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "u\varphi\,\Nearrow"] & xL \arrow[d, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL \end{tikzcd}$$

Modules (mixed assoc.)

$L \xRightarrow{\pi} M$: in $\mathbf{spn}$

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \arrow[dl, phantom, shift left=3, "\Swarrow"] \\ yL \arrow[r, "v\pi"'] \arrow[ur, phantom, shift left=3, "\Nearrow"] & zM \end{tikzcd}$$

Lax transforms

$M \xRightarrow{\sigma} N$: in $\mathbf{spn}$

$$\begin{tikzcd} yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\Swarrow v\sigma"] & yN \arrow[d, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category X such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \xRightarrow{\varphi\#} L$ and $M \xRightarrow{\sigma^{\#}} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

09. Three important types of 1-cells for systems $\boldsymbol{X} \longrightarrow \boldsymbol{spn}$

Oplax map-transforms

$K \overset{\varphi}{\Longrightarrow} L$: in $\boldsymbol{spn}$

$$\begin{tikzcd} xK \arrow[r, "x\varphi"] \arrow[d, "uK"'] \arrow[dr, phantom, "u\varphi \Nearrow"] & xL \arrow[d, "uL"] \\ yK \arrow[r, "y\varphi"'] & yL \end{tikzcd}$$

Modules (mixed assoc.)

$L \overset{\pi}{\Longrightarrow} M$: in $\boldsymbol{spn}$

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi"] & yM \arrow[d, "vM"] \arrow[dl, phantom, "\Swarrow"] \\ yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Nearrow"] & zM \end{tikzcd}$$

Lax transforms

$M \overset{\sigma}{\Longrightarrow} N$: in $\boldsymbol{spn}$

$$\begin{tikzcd} yM \arrow[r, "y\sigma"] \arrow[d, "vM"'] \arrow[dr, phantom, "\Swarrow v\sigma"] & yN \arrow[d, "vN"] \\ zM \arrow[r, "z\sigma"'] & zN \end{tikzcd}$$

- Lax transforms are closely related to simulations (of N by M).
- For lax functors from a category $\boldsymbol{X}$ such 1-cells also must be compatible with the lax structures of their domain and codomain.
- We obtain modules $K \overset{\varphi\#}{\Longrightarrow} L$ and $M \overset{\sigma^{\#}}{\Longrightarrow} N$ by pasting with the identity module on the codomain, respectively, domain.
- Modulations provide 2-cells for modules. Modifications between transforms lift faithfully, but in general not fully.

The process 1-cell corresponding to a module $L \overset{\pi}{\Longrightarrow} M$ is a span of fibre-small functors over $\boldsymbol{X}$ with a natural transformation:

π° in diagram form

$$\begin{tikzcd}
& \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{\mathcal{P}}{\Rightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
& \boldsymbol{X} &
\end{tikzcd}$$

Interpretation

Combine $\boldsymbol{Q}$ and $\boldsymbol{R}$ into a new category over $\boldsymbol{X}$ with $\boldsymbol{P}$-objects serving as *new arrows* linking $\boldsymbol{Q}$-with $\boldsymbol{R}$-objects, and $\boldsymbol{P}$-arrows serving as new commutative squares linking $\boldsymbol{Q}$- with $\boldsymbol{R}$-arrows.

Old and new arrows are composed according to the module 2-cells:

$$\begin{tikzcd}
xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] \arrow[dr, phantom, shift left=3ex, "\swarrow"] \arrow[dr, phantom, shift right=3ex, "\nearrow"] & yM \arrow[d, "vM"] \\
yL \arrow[r, "v\pi"'] & zM
\end{tikzcd}$$

In particular, $\boldsymbol{P}$ encompasses *all* new arrows.

09. 1-cells for processes over X : the module case

The process 1-cell corresponding to a module $L \overset{\pi}{\Longrightarrow\!\!\triangleright} M$ is a span of fibre-small functors over X with a natural transformation:

π° in diagram form

$$\begin{tikzcd} & P \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\ Q \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{\mathcal{P}}{\Rightarrow}"] & & R \arrow[dl, "M^\circ"] \\ & X & \end{tikzcd}$$

Interpretation

Combine Q and R into a new category over X with P-objects serving as *new arrows* linking Q-with R-objects, and P-arrows serving as new commutative squares linking Q- with R-arrows.

Old and new arrows are composed according to the module 2-cells:

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \arrow[dl, phantom, "\Leftarrow"] \\ yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Rightarrow"] & zM \end{tikzcd}$$

In particular, P encompasses *all* new arrows.

09. 1-cells for processes over $\boldsymbol{X}$: the module case

The process 1-cell corresponding to a module $L \overset{\pi}{\Longrightarrow} M$ is a span of fibre-small functors over $\boldsymbol{X}$ with a natural transformation:

π° in diagram form

$$\begin{tikzcd} & \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\ \boldsymbol{Q} \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{\mathcal{P}}{\Rightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\ & \boldsymbol{X} & \end{tikzcd}$$

Interpretation

Combine $\boldsymbol{Q}$ and $\boldsymbol{R}$ into a new category over $\boldsymbol{X}$ with $\boldsymbol{P}$-objects serving as *new arrows* linking $\boldsymbol{Q}$-with $\boldsymbol{R}$-objects, and $\boldsymbol{P}$-arrows serving as new commutative squares linking $\boldsymbol{Q}$- with $\boldsymbol{R}$-arrows.

Old and new arrows are composed according to the module 2-cells:

$$\begin{tikzcd} xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \\ yL \arrow[r, "v\pi"'] & zM \end{tikzcd}$$

In particular, $\boldsymbol{P}$ encompasses *all* new arrows.

The process 1-cell corresponding to a module $L \overset{\pi}{\Longrightarrow} M$ is a span of fibre-small functors over $\boldsymbol{X}$ with a natural transformation:

π° in diagram form

$$\begin{tikzcd}
& \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] \arrow[d, phantom, "\overset{\mathcal{P}}{\Rightarrow}"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] & \, & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
& \boldsymbol{X} &
\end{tikzcd}$$

Interpretation

Combine $\boldsymbol{Q}$ and $\boldsymbol{R}$ into a new category over $\boldsymbol{X}$ with $\boldsymbol{P}$-objects serving as *new arrows* linking $\boldsymbol{Q}$-with $\boldsymbol{R}$-objects, and $\boldsymbol{P}$-arrows serving as new commutative squares linking $\boldsymbol{Q}$- with $\boldsymbol{R}$-arrows.

Old and new arrows are composed according to the module 2-cells:

$$\begin{tikzcd}
xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \arrow[dl, phantom, "\Leftarrow" near start] \\
yL \arrow[r, "v\pi"'] \arrow[ur, phantom, "\Rightarrow" near start] & zM
\end{tikzcd}$$

In particular, $\boldsymbol{P}$ encompasses *all* new arrows.

09. 1-cells for processes over $\boldsymbol{X}$: the module case

The process 1-cell corresponding to a module $L \overset{\pi}{\Longrightarrow} M$ is a span of fibre-small functors over $\boldsymbol{X}$ with a natural transformation:

π° in diagram form

$$\begin{tikzcd}
& \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] & \overset{\mathcal{P}}{\Rightarrow} & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
& \boldsymbol{X} &
\end{tikzcd}$$

Interpretation

Combine $\boldsymbol{Q}$ and $\boldsymbol{R}$ into a new category over $\boldsymbol{X}$ with $\boldsymbol{P}$-objects serving as *new arrows* linking $\boldsymbol{Q}$-with $\boldsymbol{R}$-objects, and $\boldsymbol{P}$-arrows serving as new commutative squares linking $\boldsymbol{Q}$- with $\boldsymbol{R}$-arrows.

Old and new arrows are composed according to the module 2-cells:

$$\begin{tikzcd}
xL \arrow[r, "u\pi"] \arrow[d, "uL"'] \arrow[dr, "(u;v)\pi" description] & yM \arrow[d, "vM"] \\
yL \arrow[r, "v\pi"'] & zM
\end{tikzcd}$$

In particular, $\boldsymbol{P}$ encompasses *all* new arrows.

Differences compared to the module case

For $L \xRightarrow{\pi} M$ or $L \xRightarrow{\pi} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$
\begin{tikzcd}
 & \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\
\boldsymbol{Q} \arrow[rr, phantom, "\xRightarrow{id}"] \arrow[dr, "L^\circ"'] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
 & \boldsymbol{X} &
\end{tikzcd}
$$

Identificaton of new composites

$$
\begin{tikzcd}
yL \arrow[r, "y\pi"] \arrow[d, "vL"'] \arrow[dr, phantom, "\Leftarrow v\pi"] & yM \arrow[d, "vM"] \\
zL \arrow[r, "z\pi"'] & zM
\end{tikzcd}
\quad \text{resp.} \quad
\begin{tikzcd}
xL \arrow[r, "x\pi"] \arrow[d, "uL"'] \arrow[dr, phantom, "u\pi \Rightarrow"] & xM \arrow[d, "uM"] \\
yL \arrow[r, "y\pi"'] & yM
\end{tikzcd}
$$

- For $L \xRightarrow{\pi} M$ this makes P_1 $(1 \xrightarrow{0} 2)$-orthogonal * , which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \xRightarrow{\pi} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

10. 1-cells for processes over $\boldsymbol{X}$: the transform case

Differences compared to the module case

For $L \xRightarrow{\pi} M$ or $L \xRightarrow{\pi} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$\begin{tikzcd}
& \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{id}{\Longrightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
& \boldsymbol{X} &
\end{tikzcd}$$

Identificaton of new composites

$$\begin{tikzcd}
yL \arrow[r, "y\pi"] \arrow[d, "vL"'] \arrow[dr, phantom, "\Leftarrow v\pi"] & yM \arrow[d, "vM"] \\
zL \arrow[r, "z\pi"'] & zM
\end{tikzcd}
\quad \text{resp.} \quad
\begin{tikzcd}
xL \arrow[r, "x\pi"] \arrow[d, "uL"'] \arrow[dr, phantom, "u\pi \Rightarrow"] & xM \arrow[d, "uM"] \\
yL \arrow[r, "y\pi"'] & yM
\end{tikzcd}$$

- For $L \xRightarrow{\pi} M$ this makes P_1 $(1 \xrightarrow{0} 2)$-orthogonal * , which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \xRightarrow{\pi} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

10. 1-cells for processes over $\boldsymbol{X}$: the transform case

Differences compared to the module case

For $L \overset{\pi}{=\!\!=\!\!\rhd} M$ or $L \xRightarrow{\pi} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$
\begin{tikzcd}
& \boldsymbol{P} \arrow[dl, red, "P_0"'] \arrow[dr, red, "P_1"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{id}{\Longrightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
& \boldsymbol{X} &
\end{tikzcd}
$$

Identificaton of new composites

$$
\begin{tikzcd}
yL \arrow[r, "y\pi"] \arrow[d, "vL"'] \arrow[dr, phantom, "\Swarrow v\pi"] & yM \arrow[d, "vM"] \\
zL \arrow[r, "z\pi"'] & zM
\end{tikzcd}
\quad \text{resp.} \quad
\begin{tikzcd}
xL \arrow[r, "x\pi"] \arrow[d, "uL"'] \arrow[dr, phantom, "\Nearrow u\pi"] & xM \arrow[d, "uM"] \\
yL \arrow[r, "y\pi"'] & yM
\end{tikzcd}
$$

- For $L \overset{\pi}{=\!\!=\!\!\rhd} M$ this makes P_1 $(1 \xrightarrow{0} 2)$-orthogonal*, which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \xRightarrow{\pi} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

10. 1-cells for processes over $\boldsymbol{X}$: the transform case

Differences compared to the module case

For $L \overset{\pi}{\Longrightarrow\!\!\shortmid} M$ or $L \xRightarrow{\pi} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$\begin{tikzcd} & \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\ \boldsymbol{Q} \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{id}{\Longrightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\ & \boldsymbol{X} & \end{tikzcd}$$

Identificaton of new composites

$$\begin{tikzcd} yL \arrow[r, "y\pi"] \arrow[d, "vL"'] \arrow[dr, phantom, "\Leftarrow v\pi"] & yM \arrow[d, "vM"] \\ zL \arrow[r, "z\pi"'] & zM \end{tikzcd} \quad \text{resp.} \quad \begin{tikzcd} xL \arrow[r, "x\pi"] \arrow[d, "uL"'] \arrow[dr, phantom, "u\pi \Rightarrow"] & xM \arrow[d, "uM"] \\ yL \arrow[r, "y\pi"'] & yM \end{tikzcd}$$

- For $L \overset{\pi}{\Longrightarrow\!\!\shortmid} M$ this makes P_1 $(1 \xrightarrow{0} 2)$-orthogonal * , which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \xRightarrow{\pi} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

10. 1-cells for processes over $\boldsymbol{X}$: the transform case

Differences compared to the module case

For $L \overset{\pi}{=\!\!=\!\!\!\shortmid\!\!\!\rhd} M$ or $L \overset{\pi}{\Longrightarrow} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$\begin{tikzcd} & \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\ \boldsymbol{Q} \arrow[dr, "L^{\circ}"'] \arrow[rr, phantom, "\overset{id}{\Longrightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^{\circ}"] \\ & \boldsymbol{X} & \end{tikzcd}$$

Identificaton of new composites

$$\begin{tikzcd} yL \arrow[r, "y\pi"] \arrow[d, "vL"'] \arrow[dr, phantom, "\swarrow v\pi"] & yM \arrow[d, "vM"] \\ zL \arrow[r, "z\pi"'] & zM \end{tikzcd} \quad \text{resp.} \quad \begin{tikzcd} xL \arrow[r, "x\pi"] \arrow[d, "uL"'] \arrow[dr, phantom, "u\pi \nearrow"] & xM \arrow[d, "uM"] \\ yL \arrow[r, "y\pi"'] & yM \end{tikzcd}$$

- For $L \overset{\pi}{=\!\!=\!\!\!\shortmid\!\!\!\rhd} M$ this makes P_1 $(1 \overset{0}{\longrightarrow} 2)$-orthogonal * , which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \overset{\pi}{\Longrightarrow} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

Differences compared to the module case

For $L \overset{\pi}{=\!=\!\triangleright} M$ or $L \overset{\pi}{\Longrightarrow} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$
\begin{tikzcd}
 & \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{id}{\Longrightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
 & \boldsymbol{X} &
\end{tikzcd}
$$

Identificaton of new composites

$$
\begin{tikzcd}
yL \arrow[r, "y\pi"] \arrow[d, "vL"'] \arrow[dr, phantom, "\Swarrow v\pi"] & yM \arrow[d, "vM"] \\
zL \arrow[r, "z\pi"'] & zM
\end{tikzcd}
\quad \text{resp.} \quad
\begin{tikzcd}
xL \arrow[r, "x\pi"] \arrow[d, "uL"'] \arrow[dr, phantom, "u\pi \Nearrow"] & xM \arrow[d, "uM"] \\
yL \arrow[r, "y\pi"'] & yM
\end{tikzcd}
$$

- For $L \overset{\pi}{=\!=\!\triangleright} M$ this makes P_1 $(1 \xrightarrow{0} 2)$-orthogonal* , which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \overset{\pi}{\Longrightarrow} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

10. 1-cells for processes over $\boldsymbol{X}$: the transform case

Differences compared to the module case

For $L \overset{\pi}{=\!=\!\rhd} M$ or $L \overset{\pi}{\Longrightarrow} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$
\begin{tikzcd}
 & \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] & \overset{id}{\Longrightarrow} & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
 & \boldsymbol{X} &
\end{tikzcd}
$$

Identificaton of new composites

$$
\begin{tikzcd}
yL \arrow[r, "y\pi"] \arrow[d, "vL"'] \arrow[dr, phantom, "\Swarrow v\pi"] & yM \arrow[d, "vM"] \\
zL \arrow[r, "z\pi"'] & zM
\end{tikzcd}
\quad \text{resp.} \quad
\begin{tikzcd}
xL \arrow[r, "x\pi"] \arrow[d, "uL"'] \arrow[dr, phantom, "u\pi \Nearrow"] & xM \arrow[d, "uM"] \\
yL \arrow[r, "y\pi"'] & yM
\end{tikzcd}
$$

- For $L \overset{\pi}{=\!=\!\rhd} M$ this makes P_1 $(1 \xrightarrow{0} 2)$-orthogonal* , which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \overset{\pi}{\Longrightarrow} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

Differences compared to the module case

For $L \overset{\pi}{=\!\!=\!\!\!\rhd} M$ or $L \overset{\pi}{\Longrightarrow} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$\begin{tikzcd}
 & \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{id}{\Longrightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
 & \boldsymbol{X} &
\end{tikzcd}$$

Identificaton of new composites

$$\begin{tikzcd}
 & yL \arrow[r, "y\pi"] \arrow[dl, "vL"'] & yM \arrow[dl, "vM"] \\
zL \arrow[r, "z\pi"'] \arrow[ur, phantom, "\Leftarrow v\pi"] & zM &
\end{tikzcd}
\quad \text{resp.} \quad
\begin{tikzcd}
 & xL \arrow[r, "x\pi"] \arrow[dl, "uL"'] & xM \arrow[dl, "uM"] \\
yL \arrow[r, "y\pi"'] \arrow[ur, phantom, "u\pi \Rightarrow"] & yM &
\end{tikzcd}$$

- For $L \overset{\pi}{=\!\!=\!\!\!\rhd} M$ this makes P_1 $(1 \xrightarrow{0} 2)$-orthogonal* , which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \overset{\pi}{\Longrightarrow} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

10. 1-cells for processes over $\boldsymbol{X}$: the transform case

Differences compared to the module case

For $L \overset{\pi}{=\!=\!\!\rightharpoondown} M$ or $L \overset{\pi}{\Longrightarrow} M$ only those new arrows show up in $\boldsymbol{P}$ that live over identities in $\boldsymbol{X}$; old and new arrows now compose freely and are then identified according to the 2-cells of π :

π° in diagram form

$$\begin{tikzcd}
 & \boldsymbol{P} \arrow[dl, "P_0"'] \arrow[dr, "P_1"] & \\
\boldsymbol{Q} \arrow[dr, "L^\circ"'] \arrow[rr, phantom, "\overset{id}{\Longrightarrow}"] & & \boldsymbol{R} \arrow[dl, "M^\circ"] \\
 & \boldsymbol{X} &
\end{tikzcd}$$

Identificaton of new composites

$$\begin{tikzcd}
yL \arrow[r, "y\pi"] \arrow[d, "vL"'] \arrow[dr, phantom, "\Swarrow v\pi"] & yM \arrow[d, "vM"] \\
zL \arrow[r, "z\pi"'] & zM
\end{tikzcd}
\quad \text{resp.} \quad
\begin{tikzcd}
xL \arrow[r, "x\pi"] \arrow[d, "uL"'] \arrow[dr, phantom, "u\pi \Nearrow"] & xM \arrow[d, "uM"] \\
yL \arrow[r, "y\pi"'] & yM
\end{tikzcd}$$

- For $L \overset{\pi}{=\!=\!\!\rightharpoondown} M$ this makes P_1 $(1 \xrightarrow{0} 2)$-orthogonal* , which turns π° into a simulation of M° by L° over $\boldsymbol{X}$.
- For $L \overset{\pi}{\Longrightarrow} M$ this makes P_0 iso, which turns π° into a functor over $\boldsymbol{X}$.

With modulations as 2-cells between modules and modfications as 2-cells between transforms, we obtain equivalences

$$\llbracket \boldsymbol{X}, \boldsymbol{spn} \rrbracket_{@} \simeq \boldsymbol{Cat}^{@}\!/\!\!/\boldsymbol{X} \qquad \text{with } @ \in \{\mathrm{md}, \mathrm{lx}, \mathrm{mp}, \dots\}$$

where for fibre-small functors $\boldsymbol{Q} \xrightarrow{L} \boldsymbol{X} \xleftarrow{M} \boldsymbol{R}$ the hom-categories are given by

$$\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{md}}\!/\!\!/\boldsymbol{X} = \boldsymbol{Cat}/(L/M)$$
$$\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{lx}}\!/\!\!/\boldsymbol{X} = \{\, P \xrightarrow{P} L/M : P\alpha = \boldsymbol{id} \wedge (1 \xrightarrow{0} 2) \perp P\partial_1 \,\}$$
$$\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{mp}}\!/\!\!/\boldsymbol{X} = \{\, Q \xrightarrow{Q} L/M : Q\alpha = \boldsymbol{id} \,\}$$

where
$$\begin{tikzcd} & L/M \arrow[dl, "\partial_0"'] \arrow[dr, "\partial_1"] & \\ \boldsymbol{Q} \arrow[dr, "L"'] \arrow[rr, phantom, "\overset{\alpha}{\Rightarrow}"] & & \boldsymbol{R} \arrow[dl, "M"] \\ & \boldsymbol{X} & \end{tikzcd}$$
denotes the comma square.

With modulations as 2-cells between modules and modfications as 2-cells between transforms, we obtain equivalences

$$\llbracket \boldsymbol{X}, \boldsymbol{spn} \rrbracket_{@} \cong \boldsymbol{Cat}^{@}/\!\!/ \boldsymbol{X} \qquad \text{with } @ \in \{\mathrm{md}, \mathrm{lx}, \mathrm{mp}, \dots\}$$

where for fibre-small functors $\boldsymbol{Q} \xrightarrow{L} \boldsymbol{X} \xleftarrow{M} \boldsymbol{R}$ the hom-categories are given by

$$\langle L, M \rangle \boldsymbol{Cat}^{\mathrm{md}}/\!\!/ \boldsymbol{X} = \boldsymbol{Cat}/(L/M)$$

$$\langle L, M \rangle \boldsymbol{Cat}^{\mathrm{lx}}/\!\!/ \boldsymbol{X} = \{ \boldsymbol{P} \xrightarrow{P} L/M : P\alpha = \boldsymbol{id} \wedge (1 \xrightarrow{0} 2) \perp P\partial_1 \}$$

$$\langle L, M \rangle \boldsymbol{Cat}^{\mathrm{mp}}/\!\!/ \boldsymbol{X} = \{ \boldsymbol{Q} \xrightarrow{Q} L/M : Q\alpha = \boldsymbol{id} \}$$

where
$$\begin{tikzcd} & L/M \arrow[dl, "\partial_0"'] \arrow[dr, "\partial_1"] & \\ \boldsymbol{Q} \arrow[dr, "L"'] \arrow[rr, phantom, "\overset{\alpha}{\Rightarrow}"] & & \boldsymbol{R} \arrow[dl, "M"] \\ & \boldsymbol{X} & \end{tikzcd}$$
denotes the comma square.

11. The main equivalencess

With modulations as 2-cells between modules and modfications as 2-cells between transforms, we obtain equivalences

$$\llbracket \boldsymbol{X}, \boldsymbol{spn} \rrbracket_{@} \cong \boldsymbol{Cat}^{@}\!/\!\!/\boldsymbol{X} \qquad \text{with } @ \in \{\mathrm{md}, \mathrm{lx}, \mathrm{mp}, \dots\}$$

where for fibre-small functors $\boldsymbol{Q} \xrightarrow{L} \boldsymbol{X} \xleftarrow{M} \boldsymbol{R}$ the hom-categories are given by

$$\begin{aligned}
\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{md}}\!/\!\!/\boldsymbol{X} &= \boldsymbol{Cat}/(L/M) \\
\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{lx}}\!/\!\!/\boldsymbol{X} &= \{\, \boldsymbol{P} \xrightarrow{P} L/M : P\alpha = \boldsymbol{id} \wedge (1 \xrightarrow{0} 2) \perp P\partial_1 \,\} \\
\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{mp}}\!/\!\!/\boldsymbol{X} &= \{\, \boldsymbol{Q} \xrightarrow{Q} L/M : Q\alpha = \boldsymbol{id} \,\}
\end{aligned}$$

where
$$\begin{tikzcd}
 & L/M \arrow[dl, "\partial_0"'] \arrow[dr, "\partial_1"] \arrow[dd, phantom, "\overset{\alpha}{\Rightarrow}"] & \\
\boldsymbol{Q} \arrow[dr, "L"'] & & \boldsymbol{R} \arrow[dl, "M"] \\
 & \boldsymbol{X} &
\end{tikzcd}$$
denotes the comma square.

11. The main equivalencess

With modulations as 2-cells between modules and modfications as 2-cells between transforms, we obtain equivalences

$$\llbracket \boldsymbol{X}, \boldsymbol{spn} \rrbracket_{@} \cong \boldsymbol{Cat}^{@}/\!\!/ \boldsymbol{X} \qquad \text{with } @ \in \{\mathrm{md}, \mathrm{lx}, \mathrm{mp}, \dots\}$$

where for fibre-small functors $\boldsymbol{Q} \xrightarrow{L} \boldsymbol{X} \xleftarrow{M} \boldsymbol{R}$ the hom-categories are given by

$$\begin{aligned}
\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{md}}/\!\!/ \boldsymbol{X} &= \boldsymbol{Cat}/(L/M) \\
\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{lx}}/\!\!/ \boldsymbol{X} &= \{\, \boldsymbol{P} \xrightarrow{P} L/M : P\alpha = \boldsymbol{id} \wedge (1 \xrightarrow{0} 2) \perp P\partial_1 \,\} \\
\langle L, M\rangle \boldsymbol{Cat}^{\mathrm{mp}}/\!\!/ \boldsymbol{X} &= \{\, \boldsymbol{Q} \xrightarrow{Q} L/M : Q\alpha = \boldsymbol{id} \,\}
\end{aligned}$$

where
$$\begin{tikzcd}
 & L/M \arrow[dl, "\partial_0"'] \arrow[dr, "\partial_1"] \arrow[dd, phantom, "\overset{\alpha}{\Rightarrow}"] & \\
\boldsymbol{Q} \arrow[dr, "L"'] & & \boldsymbol{R} \arrow[dl, "M"] \\
 & \boldsymbol{X} &
\end{tikzcd}$$
denotes the comma square.

The embeddings of $\boldsymbol{set}$ into $\boldsymbol{cat}^{\mathrm{co}}$ into $\boldsymbol{prf}$, and the characterization of $\boldsymbol{prf}$ as the bicategory of monads on $\boldsymbol{spn}$ yield

$$
\begin{tikzcd}
{[\boldsymbol{X}, \boldsymbol{set}]} \arrow[r, shift left] \arrow[r, phantom, "\top"] & {[\boldsymbol{X}, \boldsymbol{cat}^{\mathrm{co}}]} \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\top"] \arrow[d, hook, "\cong"] & \lfloor \boldsymbol{X}, \boldsymbol{prf} \rfloor^{\mathrm{n}}_{\mathrm{mp}} \arrow[l, shift left] \arrow[r, phantom, "\cong"] \arrow[d, hook] & \lfloor \boldsymbol{X}, \boldsymbol{spn} \rfloor_{\mathrm{mp}} \arrow[r, phantom, "\cong"] \arrow[d, hook] & \boldsymbol{Cat}^{\mathrm{mp}}/\boldsymbol{X} \arrow[d, hook] \\
& \llbracket \boldsymbol{X}, \boldsymbol{cat}^{\mathrm{co}} \rrbracket^{\mathrm{n}}_{\mathrm{mp}} \arrow[r, shift left] \arrow[r, phantom, "\top"] \arrow[d, hook] & \llbracket \boldsymbol{X}, \boldsymbol{prf} \rrbracket^{\mathrm{n}}_{\mathrm{mp}} \arrow[l, shift left] \arrow[r, phantom, "\cong"] \arrow[d, hook] & \llbracket \boldsymbol{X}, \boldsymbol{spn} \rrbracket_{\mathrm{mp}} \arrow[r, phantom, "\cong"] \arrow[d, hook] & \boldsymbol{Cat}^{\mathrm{mp}}/\!/\boldsymbol{X} \arrow[d, hook] \\
& \llbracket \boldsymbol{X}, \boldsymbol{cat}^{\mathrm{co}} \rrbracket^{\mathrm{n}}_{\mathrm{md}} \arrow[r, shift left] \arrow[r, phantom, "\top"] & \llbracket \boldsymbol{X}, \boldsymbol{prf} \rrbracket^{\mathrm{n}}_{\mathrm{md}} \arrow[l, shift left] \arrow[r, phantom, "\cong"] & \llbracket \boldsymbol{X}, \boldsymbol{spn} \rrbracket_{\mathrm{md}} \arrow[r, phantom, "\cong"] & \boldsymbol{Cat}^{\mathrm{md}}/\!/\boldsymbol{X} \\
& \llbracket \boldsymbol{X}, \boldsymbol{cat}^{\mathrm{co}} \rrbracket^{\mathrm{n}}_{\mathrm{lx}} \arrow[r, shift left] \arrow[r, phantom, "\top"] \arrow[u, hook] & \llbracket \boldsymbol{X}, \boldsymbol{prf} \rrbracket^{\mathrm{n}}_{\mathrm{lx}} \arrow[l, shift left] \arrow[r, phantom, "\cong"] \arrow[u, hook] & \llbracket \boldsymbol{X}, \boldsymbol{spn} \rrbracket_{\mathrm{lx}} \arrow[r, phantom, "\cong"] \arrow[u, hook] & \boldsymbol{Cat}^{\mathrm{lx}}/\!/\boldsymbol{X} \arrow[u, hook] \\
& \left(\llbracket \boldsymbol{X}, \boldsymbol{cat}^{\mathrm{co}} \rrbracket^{\mathrm{n}}_{\mathrm{mp}}\right)^{\mathrm{coop}} \arrow[u, hook] & \left(\llbracket \boldsymbol{X}, \boldsymbol{prf} \rrbracket^{\mathrm{n}}_{\mathrm{mp}}\right)^{\mathrm{coop}} \arrow[r, phantom, "\cong"] \arrow[u, hook] & \left(\llbracket \boldsymbol{X}, \boldsymbol{spn} \rrbracket_{\mathrm{mp}}\right)^{\mathrm{coop}} \arrow[r, phantom, "\cong"] \arrow[u, hook] & (\boldsymbol{Cat}^{\mathrm{mp}}/\!/\boldsymbol{X})^{\mathrm{coop}} \arrow[u, hook]
\end{tikzcd}
$$

11. Specialization in various directions

- Posettal collapse: restrict to faithful processes over $\mathcal{X}$ and to systems into $\boldsymbol{rel}$; substitute $\boldsymbol{ord}$ for $\boldsymbol{cat}$ and $\boldsymbol{idl}$ for $\boldsymbol{prf}$.
- Size constraints: restrict to λ-small graphs/categories for some inaccessible cardinal λ.
- Symmetrization: restrict to symmetric graphs and spans, replace categories by groupoids, this allows modelling reversible computations, as, *e.g.*, in a quantum computer):

$$\begin{tikzcd}[E, \boldsymbol{set}] \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\mathrm{mp}}/_{\mathrm{ff}}\boldsymbol{E} \arrow[l, bend left]\end{tikzcd}$$

$$\begin{tikzcd}[\boldsymbol{E}, \boldsymbol{gpd}] \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\mathrm{mp}}/\boldsymbol{E} \arrow[l, bend left]\end{tikzcd}$$

$$\begin{tikzcd}\llbracket \boldsymbol{E}, \boldsymbol{gpd}^{\mathrm{co}} \rrbracket_{©} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{©}/\!/\boldsymbol{E} \arrow[l, bend left]\end{tikzcd}$$

11. Specialization in various directions

- Posettal collapse: restrict to faithful processes over $\mathcal{X}$ and to systems into $\boldsymbol{rel}$; substitute $\boldsymbol{ord}$ for $\boldsymbol{cat}$ and $\boldsymbol{idl}$ for $\boldsymbol{prf}$.
- Size constraints: restrict to λ-small graphs/categories for some inaccessible cardinal λ.
- Symmetrization: restrict to symmetric graphs and spans, replace categories by groupoids, this allows modelling reversible computations, as, *e.g.*, in a quantum computer):

$$\begin{tikzcd}
[\boldsymbol{E}, \boldsymbol{set}] \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\mathrm{mp}}/_{\mathrm{ff}}\boldsymbol{E} \arrow[l, bend left]
\end{tikzcd}$$

$$\begin{tikzcd}
[\boldsymbol{E}, \boldsymbol{gpd}] \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\mathrm{mp}}/\boldsymbol{E} \arrow[l, bend left]
\end{tikzcd}$$

$$\begin{tikzcd}
[\![\boldsymbol{E}, \boldsymbol{gpd}^{\mathrm{co}}]\!]_{©} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{©}/\!/\boldsymbol{E} \arrow[l, bend left]
\end{tikzcd}$$

- Posettal collapse: restrict to faithful processes over $\mathfrak{X}$ and to systems into $\boldsymbol{rel}$; substitute $\boldsymbol{ord}$ for $\boldsymbol{cat}$ and $\boldsymbol{idl}$ for $\boldsymbol{prf}$.
- Size constraints: restrict to λ-small graphs/categories for some inaccessible cardinal λ.
- Symmetrization: restrict to symmetric graphs and spans, replace categories by groupoids, this allows modelling reversible computations, as, *e.g.*, in a quantum computer):

$$\begin{tikzcd}[\boldsymbol{E}, \boldsymbol{set}] \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\mathrm{mp}}/_{\mathrm{ff}}\boldsymbol{E} \arrow[l, bend left]\end{tikzcd}$$

$$\begin{tikzcd}[\boldsymbol{E}, \boldsymbol{gpd}] \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\mathrm{mp}}/\boldsymbol{E} \arrow[l, bend left]\end{tikzcd}$$

$$\begin{tikzcd}[\![\boldsymbol{E}, \boldsymbol{gpd}^{\mathrm{co}}]\!]_{©} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{©}/\!\!/\boldsymbol{E} \arrow[l, bend left]\end{tikzcd}$$

11. Specialization in various directions

- Posettal collapse: restrict to faithful processes over $\mathcal{X}$ and to systems into $\boldsymbol{rel}$; substitute $\boldsymbol{ord}$ for $\boldsymbol{cat}$ and $\boldsymbol{idl}$ for $\boldsymbol{prf}$.
- Size constraints: restrict to λ-small graphs/categories for some inaccessible cardinal λ.
- Symmetrization: restrict to symmetric graphs and spans, replace categories by groupoids, this allows modelling reversible computations, as, *e.g.*, in a quantum computer):

$$
\begin{tikzcd}
{[\boldsymbol{E}, \boldsymbol{set}]} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\mathrm{mp}}/_{\mathrm{ff}}\, \boldsymbol{E} \arrow[l, bend left]
\end{tikzcd}
$$

$$
\begin{tikzcd}
{[\boldsymbol{E}, \boldsymbol{gpd}]} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\mathrm{mp}}/\, \boldsymbol{E} \arrow[l, bend left]
\end{tikzcd}
$$

$$
\begin{tikzcd}
{[\![\boldsymbol{E}, \boldsymbol{gpd}^{\mathrm{co}}]\!]_{@}} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{@}/\!/\, \boldsymbol{E} \arrow[l, bend left]
\end{tikzcd}
$$

- Posettal collapse: restrict to faithful processes over $\mathcal{X}$ and to systems into $\boldsymbol{rel}$; substitute $\boldsymbol{ord}$ for $\boldsymbol{cat}$ and $\boldsymbol{idl}$ for $\boldsymbol{prf}$.
- Size constraints: restrict to λ-small graphs/categories for some inaccessible cardinal λ .
- Symmetrization: restrict to symmetric graphs and spans, replace categories by groupoids, this allows modelling reversible computations, as, *e.g.*, in a quantum computer):

$$\begin{tikzcd} {[\boldsymbol{E}, \boldsymbol{set}]} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\,\mathrm{mp}}\!/_{\mathrm{ff}}\,\boldsymbol{E} \arrow[l, bend left] \end{tikzcd}$$

$$\begin{tikzcd} {[\boldsymbol{E}, \boldsymbol{gpd}]} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\,\mathrm{mp}}\!/\,\boldsymbol{E} \arrow[l, bend left] \end{tikzcd}$$

$$\begin{tikzcd} {\llbracket \boldsymbol{E}, \boldsymbol{gpd}^{\mathrm{co}} \rrbracket_{@}} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \boldsymbol{Gpd}^{\,@}\!/\!/\,\boldsymbol{E} \arrow[l, bend left] \end{tikzcd}$$