String diagrams for regular logic

David I. Spivak (joint with Brendan Fong)

Presented on 2018/10/27

Octoberfest

Outline

1. **Introduction**
   - Application: playing with logic
   - Implications for string diagrams
   - String diagrams for regular logic

2. Regular categories and regular logic

3. Bringing it all together

Minority Report

The 2002 movie *Minority report* showed detective Tom Cruise playing seamlessly with logic.

- A computer database held relevant information.
- Cruise could pull it up, and manipulate it, to solve crimes.

Minority Report

The 2002 movie *Minority report* showed detective Tom Cruise playing seamlessly with logic.

- A computer database held relevant information.
- Cruise could pull it up, and manipulate it, to solve crimes.

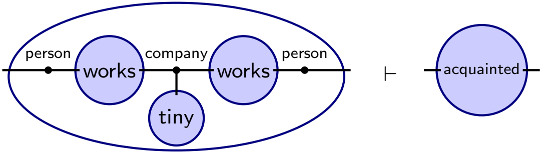
Let's imagine such a detective scenario. The knowledge base says:

- Any two people who work in the same tiny company are acquainted.
- *Categorical Informatics* is a tiny company.
- David works at *Categorical Informatics*.
- Ryan works at *Categorical Informatics*.

We of course want to conclude that David and Ryan are acquainted.

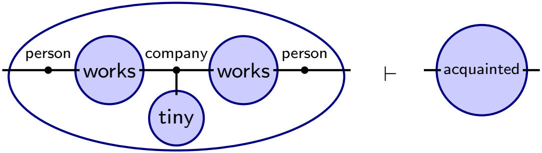
Sample scenario

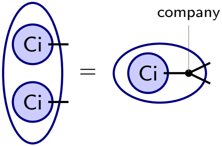
Assume:



Sample scenario

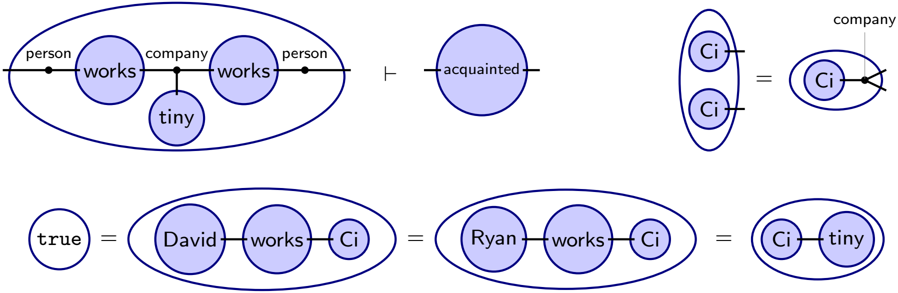
Assume:





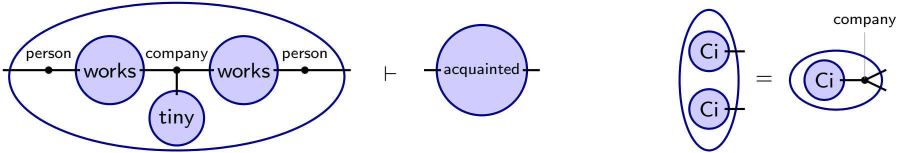
Sample scenario

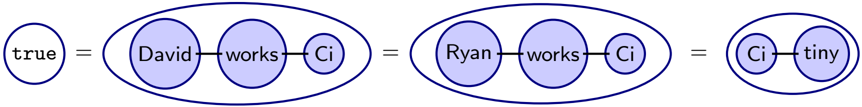
Assume:



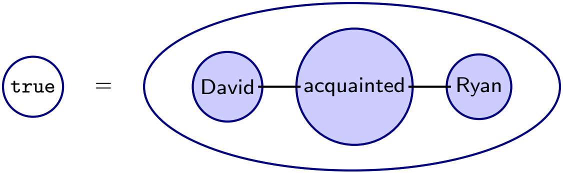
Sample scenario

Assume:

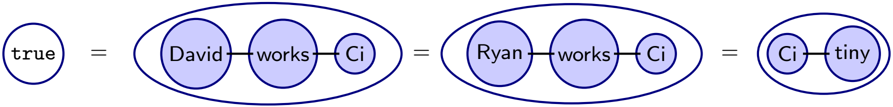




Show:



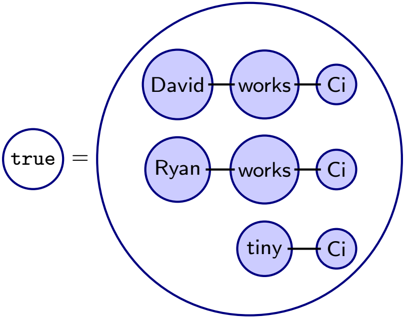
Picture proof



Combine!

Picture proof

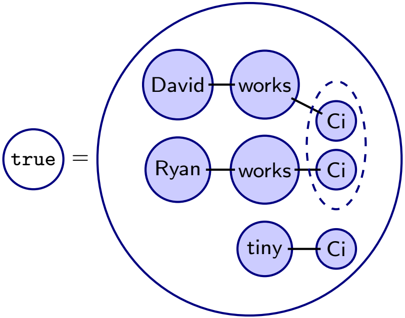
Combined:



Group two Ci's!

Picture proof

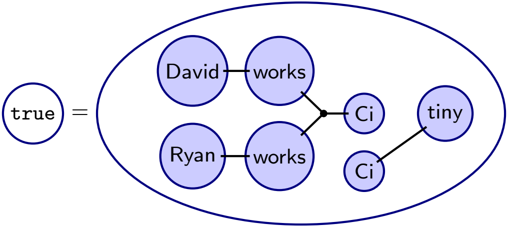
Ci's grouped:



Substitute!

Picture proof

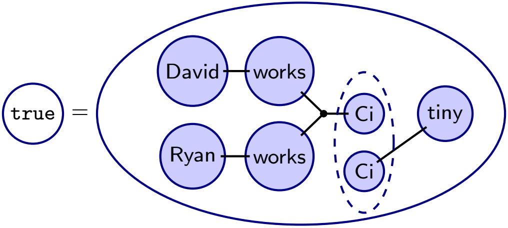
Substituted:



Group two Ci's!

Picture proof

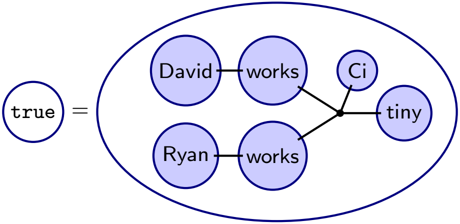
Two Ci's grouped:



Substitute!

Picture proof

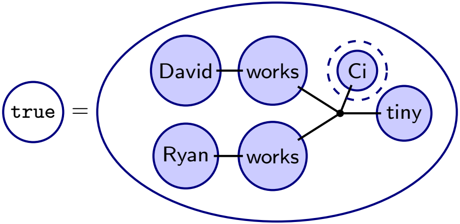
Substituted:



Group Ci!

Picture proof

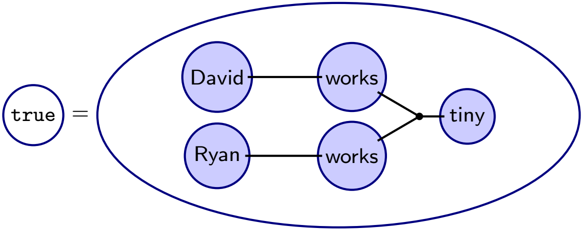
Ci grouped:



Discard group!

Picture proof

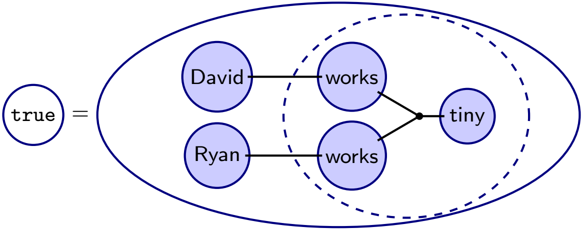
Group discarded:



Group!

Picture proof

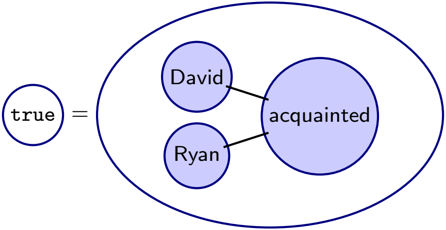
Grouped:



Substitute!

Picture proof

Substituted:

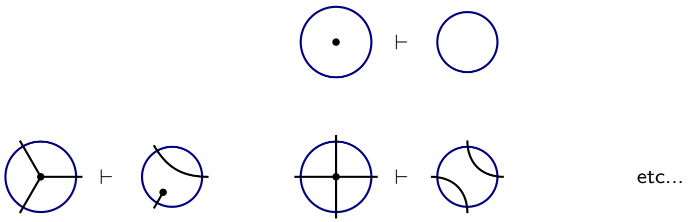


Done!

Two-dimensional manipulation of string diagrams

In this talk we discuss a 2-dimensional language for wiring diagrams.

- It includes all the sorts of operations shown above.
- Together with operations like discarding and breaking wires:



Comparing to other string diagram languages

Let's compare to string diagram calculus for traced SMCs and hypercats.

Comparing to other string diagram languages

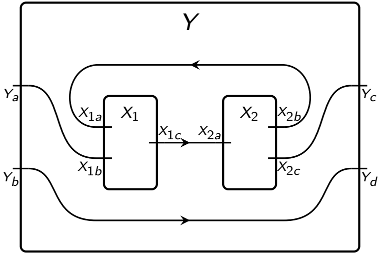
Let's compare to string diagram calculus for traced SMCs and hypercats.

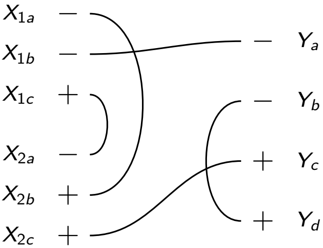
- In traced SMCs, you can compose, tensor, swap, and trace.
  - You can do these anywhere in the diagram, with axioms.

Comparing to other string diagram languages

Let's compare to string diagram calculus for traced SMCs and hypercats.

- In traced SMCs, you can compose, tensor, swap, and trace.
  - You can do these anywhere in the diagram, with axioms.
  - These can be considered generators and relations for an operad.
  - Traced categories are algebras on the operad 1-Cob.

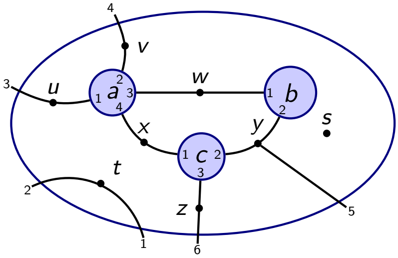


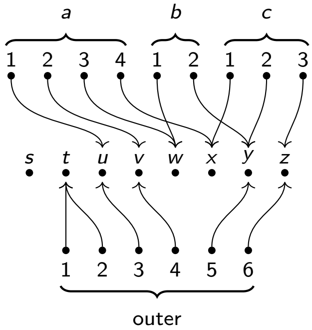


Comparing to other string diagram languages

Let's compare to string diagram calculus for traced SMCs and hypercats.

- In traced SMCs, you can compose, tensor, swap, and trace.
  - You can do these anywhere in the diagram, with axioms.
  - These can be considered generators and relations for an operad.
  - Traced categories are algebras on the operad 1-Cob.
- In hypergraph categories, add Frobenius maps, plus axioms.
  - Hypergraph categories are algebras on the operad Cospan.





Comparing to other string diagram languages

Let's compare to string diagram calculus for traced SMCs and hypercats.

- In traced SMCs, you can compose, tensor, swap, and trace.
  - You can do these anywhere in the diagram, with axioms.
  - These can be considered generators and relations for an operad.
  - Traced categories are algebras on the operad 1-Cob.
- In hypergraph categories, add Frobenius maps, plus axioms.
  - Hypergraph categories are algebras on the operad Cospan.

- In our picture proof, we had more operations and relations.
  - Order on elements of each arity, preserved by substitution.
  - Meet-semilattice structures on elements of each arity.
  - Top element (`true`) can be discarded; corresponding structure for $\wedge$.
  - Removing dots, breaking wires.

We will see that this is a 2-dimensional structure.

Formal presentation of the calculus I.

The graphical calculus shown above can be understood as follows.

- Fix a set Λ (elements will be string labels).
- Consider the monoidal bicategory $\mathcal{C}\mathsf{ospan}_\Lambda^{\mathsf{co}}$.

Formal presentation of the calculus I.

The graphical calculus shown above can be understood as follows.

- Fix a set Λ (elements will be string labels).
- Consider the monoidal bicategory $\mathcal{C}\mathrm{ospan}_{\Lambda}^{\mathrm{co}}$.
  - Objects: arities $\underline{n} \xrightarrow{v} \Lambda$, i.e. lists $(v(1), \ldots, v(n)) \in \Lambda^n$.
  - 1-morphisms: $\begin{tikzcd} n_1 \arrow[r] \arrow[dr, "v_1"'] & n_{12} \arrow[d] & n_2 \arrow[l] \arrow[dl, "v_2"] \\ & \Lambda & \end{tikzcd}$
  - 2-morphisms: opposite of usual direction (hence $-^{\mathrm{co}}$)
  - Monoidal structure: $(0, +)$.

Formal presentation of the calculus I.

The graphical calculus shown above can be understood as follows.

- Fix a set Λ (elements will be string labels).
- Consider the monoidal bicategory $\mathcal{C}\mathrm{ospan}_{\Lambda}^{\mathrm{co}}$.
  - Objects: arities $\underline{n} \xrightarrow{v} \Lambda$, i.e. lists $(v(1), \ldots, v(n)) \in \Lambda^n$.
  - 1-morphisms: $\begin{tikzcd} n_1 \arrow[r] \arrow[dr, "v_1"'] & n_{12} \arrow[d] & n_2 \arrow[l] \arrow[dl, "v_2"] \\ & \Lambda & \end{tikzcd}$
  - 2-morphisms: opposite of usual direction (hence $-^{\mathrm{co}}$)
  - Monoidal structure: $(0, +)$.
- Consider the (locally posetal) monoidal bicategory $\mathcal{P}\mathrm{oset}$.
  - Obj: posets; 1-morphisms: monotone maps; 2-morphisms: nat. trans.
  - Monoidal structure: $(1, \times)$.

Formal presentation of the calculus II.

We have monoidal bicategories $\mathcal{C}\mathsf{ospan}$ and $\mathcal{P}\mathsf{oset}$.

Definition

A *regular hypergraph category* is a lax monoidal 2-functor

$$T\colon \mathcal{C}\mathsf{ospan}_{\wedge}^{\mathsf{co}} \to \mathcal{P}\mathsf{oset}$$

such that the laxators are right adjoints.

Formal presentation of the calculus II.

We have monoidal bicategories $\mathcal{C}\mathsf{ospan}$ and $\mathcal{P}\mathsf{oset}$.

Definition

A *regular hypergraph category* is a lax monoidal 2-functor

$$T\colon \mathcal{C}\mathsf{ospan}_{\Lambda}^{\mathsf{co}} \to \mathcal{P}\mathsf{oset}$$

such that the laxators are right adjoints.

Silly terminology: *ajax* monoidal functors: the laxators

$$1 \xrightarrow{\rho_1} T(0) \qquad \text{and} \qquad T(v) \times T(v') \xrightarrow{\rho_{v,v'}} T(v+v')$$

Formal presentation of the calculus II.

We have monoidal bicategories $\mathcal{C}$ospan and $\mathcal{P}$oset.

Definition

A *regular hypergraph category* is a lax monoidal 2-functor

$$T\colon \mathcal{C}\mathrm{ospan}_{\wedge}^{\mathrm{co}} \to \mathcal{P}\mathrm{oset}$$

such that the laxators are right adjoints.

Silly terminology: *ajax* monoidal functors: the laxators are adjoints

$$1 \underset{\lambda_1}{\overset{\rho_1}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} T(0) \qquad \text{and} \qquad T(v) \times T(v') \underset{\lambda_{v,v'}}{\overset{\rho_{v,v'}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} T(v+v')$$

Aside: we're pushing this notation for adjunctions

Throughout this talk, I'll use a new notation for adjunctions.

- Usual notation: $C \underset{L}{\overset{R}{\underset{\leftarrow}{\overset{\rightarrow}{\top}}}} D \qquad C \underset{R}{\overset{L}{\underset{\rightarrow}{\overset{\leftarrow}{\bot}}}} D \qquad D \underset{R}{\overset{L}{\underset{\leftarrow}{\overset{\rightarrow}{\bot}}}} C.$
  - Note that $\top$ is sometimes used as the name of a monad, but...
  - ... it really doesn't indicate where the monad is (it's on D).

Aside: we're pushing this notation for adjunctions

Throughout this talk, I'll use a new notation for adjunctions.

- Usual notation: $C \underset{L}{\overset{R}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} D \qquad C \underset{R}{\overset{L}{\underset{\longrightarrow}{\overset{\longleftarrow}{\perp}}}} D \qquad D \underset{R}{\overset{L}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} C.$
  - Note that $\top$ is sometimes used as the name of a monad, but...
  - ... it really doesn't indicate where the monad is (it's on D).
- Our notation: $C \underset{L}{\overset{R}{\underset{\longleftarrow}{\overset{\longrightarrow}{\Leftarrow}}}} D \qquad C \underset{R}{\overset{L}{\underset{\longrightarrow}{\overset{\longleftarrow}{\Leftarrow}}}} D \qquad D \underset{R}{\overset{L}{\underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}}}} C$
  - The 2-arrow points in the direction of the left adjoint.
  - Reason: it tells you the direction of the unit and counit.

Aside: we're pushing this notation for adjunctions

Throughout this talk, I'll use a new notation for adjunctions.

- Usual notation: $$\begin{tikzcd} C \arrow[r, shift left, "R"] \arrow[r, phantom, "\top"] & D \arrow[l, shift left, "L"] \end{tikzcd}$$
  - Note that $\top$ is sometimes used as the name of a monad, but...
  - ... it really doesn't indicate where the monad is (it's on D).
- Our notation: $$\begin{tikzcd} C \arrow[r, shift left, "R"] \arrow[r, phantom, "\Leftarrow"] & D \arrow[l, shift left, "L"] \end{tikzcd}$$
  - The 2-arrow points in the direction of the left adjoint.
  - Reason: it tells you the direction of the unit and counit.

$$\begin{tikzcd} C \arrow[dr, "R"] \arrow[dd, equal] & \\ \arrow[r, phantom, "\Leftarrow"] & D \arrow[dl, "L"] \\ C & \end{tikzcd} \qquad \begin{tikzcd} & D \arrow[dd, equal] \\ C \arrow[ur, "R"] \arrow[r, phantom, "\Leftarrow"] & {} \\ & D \arrow[ul, "L"] \end{tikzcd}$$

Regular hypergraph categories and regular categories

Denote by $\mathcal{C}$ospan-Alg the category of regular hypergraph categories, i.e. sets Λ and ajax 2-functors

$$T\colon \mathcal{C}\mathsf{ospan}_{\Lambda}^{\mathsf{co}} \to \mathcal{P}\mathsf{oset}.$$

Regular hypergraph categories and regular categories

Denote by $\mathcal{C}$ospan-Alg the category of regular hypergraph categories, i.e. sets Λ and ajax 2-functors

$$T\colon \mathcal{C}\mathrm{ospan}_\Lambda^{\mathrm{co}} \to \mathcal{P}\mathrm{oset}.$$

Theorem

There is an adjunction

$$\mathcal{C}\mathrm{ospan}\text{-}\mathrm{Alg} \underset{\Psi}{\overset{\Phi}{\underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}}}} \mathrm{RegCat}\ ,$$

such that for any regular category $\mathcal{R}$, the counit $\Psi(\Phi(\mathcal{R})) \to \mathcal{R}$ is an equivalence of categories.

Plan

- We'll return to the theorem shortly.
- First we want to recall the definition of regular categories.
- We also want to make the connection to regular logic.
    - The $\mathcal{C}$ospan-algebra story is a graphical representation of the logic.
    - This will be evident, but one can take the theorem as justification.
- Then we'll unpack the theorem and conclude.

Outline

Regular categories

Definition

A *regular category* is a category for which

- all finite limits exist,
- the kernel pair of any morphism admits a coequalizer, and
- coequalizers are stable under pullback.

Regular categories

Definition

A *regular category* is a category for which
- all finite limits exist,
- the kernel pair of any morphism admits a coequalizer, and
- coequalizers are stable under pullback.

Examples of regular categories:
- Set, and more generally any topos;
- $\mathrm{Set}^{\mathrm{op}}$, opposite of any topos, $\mathrm{TopSp}^{\mathrm{op}}$;
- The category of models of any Lawvere theory (Groups, Rings, ...);
- The slice (also the coslice) of any regular category over any object;
- Exponential ideal: if $\mathcal{R}$ regular and $\mathcal{C}$ a category, then $\mathcal{R}^{\mathcal{C}}$ is regular.

How to think of regular categories

Regular categories are those with a good *bicategory of relations*.

- A relation in $\mathcal{R}$ is a subobject $S \subseteq A \times B$.
- When $\mathcal{R}$ is regular, pullbacks and images play nicely...

How to think of regular categories

Regular categories are those with a good *bicategory of relations*.

- A relation in $\mathcal{R}$ is a subobject $S \subseteq A \times B$.
- When $\mathcal{R}$ is regular, pullbacks and images play nicely...
- ... so that relations form a posetal bicategory $\mathcal{R}\mathrm{el}_{\mathcal{R}}$.
  - That is, relations can be composed and compared.
  - One can recover the morphisms in $\mathcal{R}$ as the adjunctions in $\mathcal{R}\mathrm{el}_{\mathcal{R}}$!

How to think of regular categories

Regular categories are those with a good *bicategory of relations*.

- A relation in $\mathcal{R}$ is a subobject $S \subseteq A \times B$.
- When $\mathcal{R}$ is regular, pullbacks and images play nicely...
- ... so that relations form a posetal bicategory $\mathcal{R}\mathrm{el}_{\mathcal{R}}$.
    - That is, relations can be composed and compared.
    - One can recover the morphisms in $\mathcal{R}$ as the adjunctions in $\mathcal{R}\mathrm{el}_{\mathcal{R}}$!

- Every young category theorist should prove to themselves that Set is the category of adjunctions in Rel.

How to think of regular categories

Regular categories are those with a good *bicategory of relations*.

- A relation in $\mathcal{R}$ is a subobject $S \subseteq A \times B$.
- When $\mathcal{R}$ is regular, pullbacks and images play nicely...
- ... so that relations form a posetal bicategory $\mathcal{R}\mathsf{el}_{\mathcal{R}}$.
  - That is, relations can be composed and compared.
  - One can recover the morphisms in $\mathcal{R}$ as the adjunctions in $\mathcal{R}\mathsf{el}_{\mathcal{R}}$!
- Every young category theorist should prove to themselves that Set is the category of adjunctions in Rel.

Regular categories have enough structure to do regular logic.

Regular logic and regular categories

In regular logic, one has

- A set of *types* Λ

Regular logic and regular categories

In regular logic, one has

- A set of *types* Λ
- A set of *relation symbols* $\vdash_{a_1:A_1,\ldots,a_k:A_k} R_1(a_1,\ldots,a_k) : Prop$

Regular logic and regular categories

In regular logic, one has

- A set of *types* Λ
- A set of *relation symbols* $\vdash_{a_1:A_1,\ldots,a_k:A_k} R_1(a_1,\ldots,a_k) : \mathit{Prop}$
- Operations $\wedge$, $\texttt{true}$, $=$, and $\exists$, from which to build up formulas φ, ψ.

Regular logic and regular categories

In regular logic, one has

- A set of *types* Λ
- A set of *relation symbols* $\vdash_{a_1:A_1,\ldots,a_k:A_k} R_1(a_1,\ldots,a_k) : \mathit{Prop}$
- Operations $\wedge$, `true` , $=$, and $\exists$, from which to build up formulas φ, ψ.
- A notion of entailment: $\varphi \vdash_{a:A,b:B} \psi$.
- A set of axioms involving entailment.

Regular logic and regular categories

In regular logic, one has

- A set of *types* Λ
- A set of *relation symbols* $\vdash_{a_1:A_1,\ldots,a_k:A_k} R_1(a_1,\ldots,a_k) : \mathit{Prop}$
- Operations $\wedge$, $\mathtt{true}$, $=$, and $\exists$, from which to build up formulas φ, ψ.
- A notion of entailment: $\varphi \vdash_{a:A,b:B} \psi$.
- A set of axioms involving entailment.

Example: the regular theory of "two sets and a function":

Regular logic and regular categories

In regular logic, one has

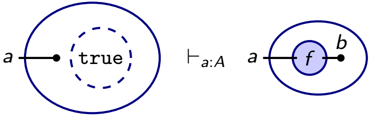
- A set of *types* Λ
- A set of *relation symbols* $\vdash_{a_1:A_1,\ldots,a_k:A_k} R_1(a_1,\ldots,a_k) : \mathit{Prop}$
- Operations $\wedge$, $\mathtt{true}$, $=$, and $\exists$, from which to build up formulas φ,ψ.
- A notion of entailment: $\varphi \vdash_{a:A,b:B} \psi$.
- A set of axioms involving entailment.

Example: the regular theory of "two sets and a function":

- $\Lambda = \{A, B\}$, one relation symbol: $\vdash_{a:A,b:B} f(a,b) : \mathit{Prop}$
- Axioms:

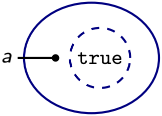
$$
\begin{array}{ll}
f \text{ is "total":} & \mathtt{true} \vdash_{a:A} \exists(b:B).\, f(a,b) \\
f \text{ is "deterministic":} & \exists(a:A).\, f(a,b) = f(a,b') \vdash_{b,b':B} b = b'
\end{array}
$$

Regular logic and cospan-algebras

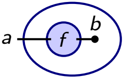


$$\texttt{true} \vdash_{a:A} \exists(b : B).\, f(a, b)$$

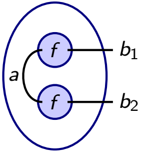
Regular logic and cospan-algebras



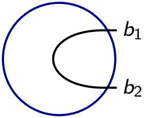
$\vdash_{a:A}$



$$\texttt{true} \vdash_{a:A} \exists(b : B).\, f(a, b)$$



$\vdash_{b_1, b_2 : B}$



$$\exists(a : A).\, f(a, b_1) = f(a, b_2) \vdash_{b_1, b_2 : B} b_1 = b_2$$

Outline

Where are we?

We have regular categories, regular logic, and cospan-algebras.

- They are three different perspectives on the same subject.
- Regular logic is an "internal language" for regular categories.

Where are we?

We have regular categories, regular logic, and cospan-algebras.

- They are three different perspectives on the same subject.
- Regular logic is an "internal language" for regular categories.
- The bicategory of cospans is a "string diagram language" for regcats.

Where are we?

We have regular categories, regular logic, and cospan-algebras.

- They are three different perspectives on the same subject.
- Regular logic is an “internal language” for regular categories.
- The bicategory of cospans is a “string diagram language” for regcats.

Next we’ll recall the theorem, give one slide of justification, and conclude.

Recalling the theorem

Recall that a $\mathcal{C}$ospan-algebra is an ajax 2-functor $T\colon \mathcal{C}\mathrm{ospan}_{\Lambda}^{\mathrm{co}} \to \mathcal{P}\mathrm{oset}$.

Theorem

There is an adjunction $\mathcal{C}\mathrm{ospan}\text{-}\mathrm{Alg} \underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}} \mathrm{RegCat}$ *, such that* $\Psi(\Phi(\mathcal{R})) \to \mathcal{R}$ *is an equivalence for any regular category* $\mathcal{R}$*.*

Recalling the theorem

Recall that a $\mathcal{C}$ospan-algebra is an ajax 2-functor $T\colon \mathcal{C}\mathrm{ospan}_\wedge^{\mathrm{co}} \to \mathcal{P}\mathrm{oset}$.

Theorem

There is an adjunction $\mathcal{C}\mathrm{ospan\text{-}Alg} \underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}} \mathrm{RegCat}$, *such that* $\Psi(\Phi(\mathcal{R})) \to \mathcal{R}$ *is an equivalence for any regular category* $\mathcal{R}$*.*

Comments:

- We can beef this up to a 2-reflection $\mathcal{R}\mathrm{egCat} \subseteq \mathcal{C}\mathrm{ospan\text{-}Alg}$.

Recalling the theorem

Recall that a $\mathcal{C}$ospan-algebra is an ajax 2-functor $T\colon \mathcal{C}\mathrm{ospan}_\wedge^{\mathrm{co}} \to \mathcal{P}\mathrm{oset}$.

Theorem

There is an adjunction $\mathcal{C}\mathrm{ospan}\text{-}\mathrm{Alg} \underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}} \mathrm{RegCat}$ *, such that* $\Psi(\Phi(\mathcal{R})) \to \mathcal{R}$ *is an equivalence for any regular category* $\mathcal{R}$*.*

Comments:

- We can beef this up to a 2-reflection $\mathcal{R}\mathrm{egCat} \subseteq \mathcal{C}\mathrm{ospan}\text{-}\mathrm{Alg}$.
- Cospan algebras and regular categories look different on the surface.
    - Remember how complicated the def. of regcats was?
    - Finite limits, coequalizers of kernel pairs, pullback stability.
    - $\mathcal{C}$ospan-Alg is certain functors $\mathcal{C}\mathrm{ospan} \to \mathcal{P}\mathrm{oset}$.

Recalling the theorem

Recall that a $\mathcal{C}$ospan-algebra is an ajax 2-functor $T \colon \mathcal{C}\text{ospan}_{\wedge}^{\text{co}} \to \mathcal{P}\text{oset}$.

Theorem

There is an adjunction $\mathcal{C}\text{ospan-Alg} \underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}} \text{RegCat}$ *, such that* $\Psi(\Phi(\mathcal{R})) \to \mathcal{R}$ *is an equivalence for any regular category* $\mathcal{R}$*.*

Comments:

- We can beef this up to a 2-reflection $\mathcal{R}\text{egCat} \subseteq \mathcal{C}\text{ospan-Alg}$.
- Cospan algebras and regular categories look different on the surface.
  - Remember how complicated the def. of regcats was?
  - Finite limits, coequalizers of kernel pairs, pullback stability.
  - $\mathcal{C}$ospan-Alg is certain functors $\mathcal{C}\text{ospan} \to \mathcal{P}\text{oset}$.
- Easier to see posets and adjunctions in RegCat: subobject lattices.

Why it works

One can form the *syntactic category* $\mathcal{R}_T$ of $T\colon \mathcal{C}\mathrm{ospan}_\Lambda \to \mathcal{P}\mathrm{oset}$.

- $\mathrm{Ob}(\mathcal{R}_T) := \{(v, \varphi) \mid \underline{n} \xrightarrow{v} \Lambda, \varphi \in T(v)\}$.

Why it works

One can form the *syntactic category* $\mathcal{R}_T$ of $T\colon \mathcal{C}\mathrm{ospan}_\Lambda \to \mathcal{P}\mathrm{oset}$.

- $\mathrm{Ob}(\mathcal{R}_T) \coloneqq \{(v,\varphi) \mid \underline{n} \xrightarrow{v} \Lambda, \varphi \in T(v)\}$. 

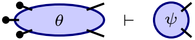
- $\mathcal{R}_T((v,\varphi),(v',\varphi')) \coloneqq \{\theta \in T(v+v') \mid \theta \vdash \varphi,\ \theta \vdash \varphi',\ \theta \text{ is functional}\}$

+ another logical condition

Why it works

One can form the *syntactic category* $\mathcal{R}_T$ of $T\colon \mathcal{C}\mathrm{ospan}_\Lambda \to \mathcal{P}\mathrm{oset}$.

- $\mathrm{Ob}(\mathcal{R}_T) := \{(v,\varphi) \mid \underline{n} \xrightarrow{v} \Lambda, \varphi \in T(v)\}$. 
- $\mathcal{R}_T((v,\varphi),(v',\varphi')) := \{\theta \in T(v{+}v') \mid \theta \vdash \varphi,\ \theta \vdash \varphi',\ \theta \text{ is functional}\}$



+ another logical condition

One shows that this syntactic category is regular.

- E.g. for each v, the poset $T(v)$ is automatically a meet-semilattice.

Why it works

One can form the *syntactic category* $\mathcal{R}_T$ of $T\colon \mathcal{C}\mathrm{ospan}_\Lambda \to \mathcal{P}\mathrm{oset}$.

- $\mathrm{Ob}(\mathcal{R}_T) \coloneqq \{(v,\varphi) \mid \underline{n} \xrightarrow{v} \Lambda, \varphi \in T(v)\}$.

- $\mathcal{R}_T((v,\varphi),(v',\varphi')) \coloneqq \{\theta \in T(v+v') \mid \theta \vdash \varphi,\ \theta \vdash \varphi',\ \theta \text{ is functional}\}$

$+$ another logical condition

One shows that this syntactic category is regular.

- E.g. for each v, the poset $T(v)$ is automatically a meet-semilattice.
  - Why? Any function $v \xrightarrow{f} w$ is an adjoint in $\mathcal{C}\mathrm{ospan}$...
  - ... so $T(f)$ will be an adjoint in $\mathcal{P}\mathrm{oset}$. Thus we get adjunctions:

Why it works

One can form the *syntactic category* $\mathcal{R}_T$ of $T\colon \mathcal{C}\text{ospan}_\Lambda \to \mathcal{P}\text{oset}$.

- $\text{Ob}(\mathcal{R}_T) := \{(v,\varphi) \mid \underline{n} \xrightarrow{v} \Lambda, \varphi \in T(v)\}$. 

- $\mathcal{R}_T((v,\varphi),(v',\varphi')) := \{\theta \in T(v+v') \mid \theta \vdash \varphi,\ \theta \vdash \varphi',\ \theta \text{ is functional}\}$

$+$ another logical condition

One shows that this syntactic category is regular.

- E.g. for each v, the poset $T(v)$ is automatically a meet-semilattice.
  - Why? Any function $v \xrightarrow{f} w$ is an adjoint in $\mathcal{C}\text{ospan}$...
  - ... so $T(f)$ will be an adjoint in $\mathcal{P}\text{oset}$. Thus we get adjunctions:

$$\begin{tikzcd} 1 \arrow[r, shift left, "\rho_1"] \arrow[r, phantom, "\Leftarrow"] & T(0) \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\Leftarrow"] & T(v) \arrow[l, shift left] \end{tikzcd}$$

Why it works

One can form the *syntactic category* $\mathcal{R}_T$ of $T\colon \mathcal{C}\mathrm{ospan}_\Lambda \to \mathcal{P}\mathrm{oset}$.

- $\mathrm{Ob}(\mathcal{R}_T) := \{(v,\varphi) \mid \underline{n} \xrightarrow{v} \Lambda, \varphi \in T(v)\}$. 

- $\mathcal{R}_T((v,\varphi),(v',\varphi')) := \{\theta \in T(v+v') \mid \theta \vdash \varphi,\ \theta \vdash \varphi',\ \theta \text{ is functional}\}$

+ another logical condition

One shows that this syntactic category is regular.

- E.g. for each v, the poset $T(v)$ is automatically a meet-semilattice.
  - Why? Any function $v \xrightarrow{f} w$ is an adjoint in $\mathcal{C}\mathrm{ospan}$...
  - ... so $T(f)$ will be an adjoint in $\mathcal{P}\mathrm{oset}$. Thus we get adjunctions:

$$\begin{tikzcd} 1 \arrow[r, shift left, "\rho_1"] \arrow[r, phantom, "\Leftarrow"] & T(0) \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\Leftarrow"] & T(v) \arrow[l, shift left] \end{tikzcd}$$

$$\begin{tikzcd} T(v)\times T(v) \arrow[r, shift left, "\rho_{v,v}"] \arrow[r, phantom, "\Leftarrow"] & T(v+v) \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\Leftarrow"] & T(v) \arrow[l, shift left] \end{tikzcd}$$

Conjecture and outlook

We conjecture that this story extends to coherent and geometric logic.

Conjecture and outlook

We conjecture that this story extends to coherent and geometric logic.

Conjecture

The 2-category of coherent categories is reflective in that of:

Conjecture and outlook

We conjecture that this story extends to coherent and geometric logic.

Conjecture

The 2-category of coherent categories is reflective in that of: lax monoidal 2-functors $\mathcal{C}\mathsf{ospan}^{\mathsf{co}} \to \mathcal{J}\mathsf{Lat}$

Conjecture and outlook

We conjecture that this story extends to coherent and geometric logic.

Conjecture

The 2-category of coherent categories is reflective in that of: lax monoidal 2-functors $\mathcal{C}\mathrm{ospan}^{\mathrm{co}} \to \mathcal{J}\mathrm{Lat}$ whose composite with $\mathcal{J}\mathrm{Lat} \to \mathcal{P}\mathrm{oset}$ is ajax.

Conjecture and outlook

We conjecture that this story extends to coherent and geometric logic.

Conjecture

The 2-category of geometric categories is reflective in that of: lax monoidal 2-functors $\mathcal{C}\mathsf{ospan}^{\mathsf{co}} \to \mathcal{S}\mathsf{upLat}$ whose composite with $\mathcal{S}\mathsf{upLat} \to \mathcal{P}\mathsf{oset}$ is ajax.

Conjecture and outlook

We conjecture that this story extends to coherent and geometric logic.

Conjecture

The 2-category of geometric categories is reflective in that of: lax monoidal 2-functors $\mathcal{C}\mathrm{ospan}^{\mathrm{co}} \to \mathcal{S}\mathrm{upLat}$ whose composite with $\mathcal{S}\mathrm{upLat} \to \mathcal{P}\mathrm{oset}$ is ajax.

- Dropping the ajax condition may give something like quantaloids.

Conjecture and outlook

We conjecture that this story extends to coherent and geometric logic.

Conjecture

The 2-category of geometric categories is reflective in that of: lax monoidal 2-functors $\mathcal{C}\mathrm{ospan}^{\mathrm{co}} \to \mathcal{S}\mathrm{upLat}$ whose composite with $\mathcal{S}\mathrm{upLat} \to \mathcal{P}\mathrm{oset}$ is ajax.

- Dropping the ajax condition may give something like quantaloids.
- Landing in categories other than $\mathcal{P}\mathrm{oset}$ gives "fuzzy regcats."
  - E.g. $\mathcal{C}\mathrm{ospan} \to \mathcal{L}\mathrm{awvMetSp}$: "distance to entailment" $\varphi \vdash^{17} \psi$.
  - Other quantales (e.g. powerset of a monoid) give other fuzz.

Summary

- Formulas in regular logic looks like this:

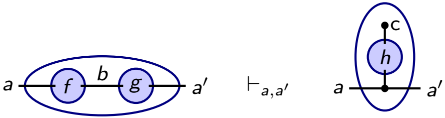
$$\exists b.\, f(a,b) \wedge g(b,a') \vdash_{a,a'} \exists c.\, h(c,a) \wedge a = a'.$$

Summary

- Formulas in regular logic looks like this:

$$\exists b.\, f(a,b) \wedge g(b,a') \vdash_{a,a'} \exists c.\, h(c,a) \wedge a = a'.$$

- Such things can be represented pictorially in a regular hypercat:



i.e. as an inequality of elements in an ajax monoidal 2-functor

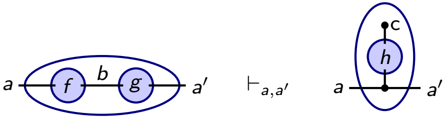
$$T\colon \mathcal{C}\mathrm{ospan}^{\mathrm{co}} \to \mathcal{P}\mathrm{oset}.$$

Summary

- Formulas in regular logic looks like this:

$$\exists b.\, f(a,b) \wedge g(b,a') \vdash_{a,a'} \exists c.\, h(c,a) \wedge a = a'.$$

- Such things can be represented pictorially in a regular hypercat:



i.e. as an inequality of elements in an ajax monoidal 2-functor

$$T\colon \mathcal{C}\mathrm{ospan}^{\mathrm{co}} \to \mathcal{P}\mathrm{oset}.$$

- We have 2-reflectivity, suggesting that the diagram language is robust.

Thanks! Comments and questions welcome.