Retracing GoI with Phil

Samson Abramsky

Department of Computer Science, UCL

MFPS 2024

Memories of Phil

Enthusiasm, Laughter, Kindness

Paper with Phil

Abramsky, Samson, Esfandiar Haghverdi, and Philip Scott. "Geometry of interaction and linear combinatory algebras." Mathematical Structures in Computer Science 12, no. 5 (2002): 625-665.

Geometry of Interaction

Series of papers by Girard:

- Multiplicatives (1988, unpublished preprint)
- Towards a Geometry of Interaction (1989)
- Geometry of Interaction I: Interpretation of System F (1989)
- Geometry of Interaction II-V (1988,1995,2011)

Early work by Danos, Regnier and Malacaria

- Some results on the interpretation of λ-calculis in operator algebras (M & R, 1991)
- Local and asynchronous beta-reduction (D & R, 1993)

SA and Radha Jagadeesan (1992)

- New Foundations for the Geometry of Interaction
- Games and Full Completeness for Multiplicative Linear Logic

GoI: what was that all about?

Views of the elephant:

GoI: what was that all about?

Views of the elephant:

- D & R:
  - leads to a novel kind of abstract machine
  - connections with optimal reduction

GoI: what was that all about?

Views of the elephant:

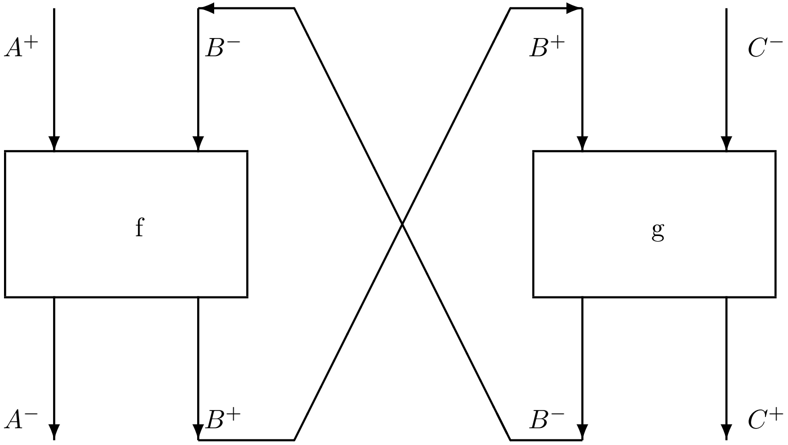
- D & R:
  - leads to a novel kind of abstract machine
  - connections with optimal reduction
- Girard
  - between proof theory and static semantics (e.g. coherence spaces)
  - dynamics of cut-elimination
  - operator algebras: a red herring

GoI: what was that all about?

Views of the elephant:

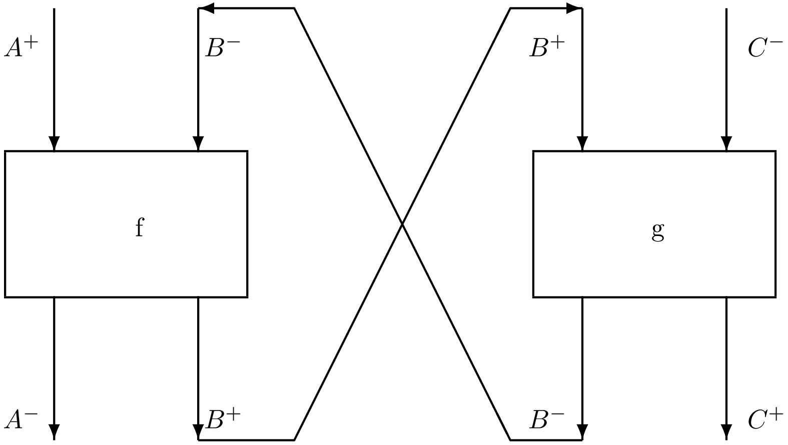
- D & R:
  - leads to a novel kind of abstract machine
  - connections with optimal reduction
- Girard
  - between proof theory and static semantics (e.g. coherence spaces)
  - dynamics of cut-elimination
  - operator algebras: a red herring
- My perspective
  - Adapting Girard, novel way of interpolating between operational and denotational semantics
  - close connection with game semantics (being developed at the same time)
  - GoI = "game semantics without games"

GoI in one picture



Composition as symmetric feedback.

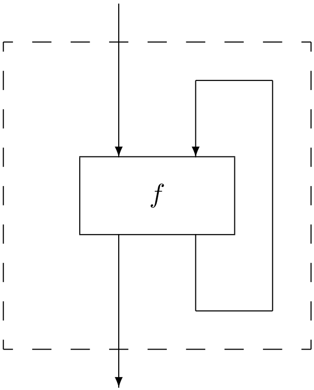
GoI in one picture



Composition as symmetric feedback.
Almost commutative! (up to symmetry).

Traced monoidal categories

$$\frac{f : A \otimes B \longrightarrow C \otimes B}{\mathsf{Tr}(f) : A \longrightarrow C}$$



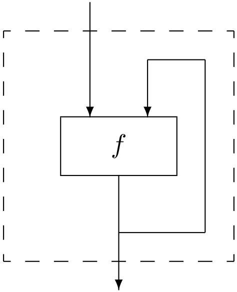
Linearity of Trace

Note that trace is linear, unlike the **iteration** concept of iterative theories (Elgot, Bloom, Esik) used to model flowcharts.

$$\frac{f : A \times B \longrightarrow B}{\mathsf{It}(f) : A \longrightarrow B}$$

$$f \circ \langle 1, \mathsf{It}(f) \rangle \quad = \quad \mathsf{It}(f)$$

There is no implicit **sharing** of wires as in iteration.



Mathematical essence of multiplicative GoI

- Feedback in a **linear** (monoidal) setting
- Building a free compact closed category from a traced category – "higher-order" vs. "first-order".
- The composition in this free compact closed category (i.e. symmetric feedback) subsumes the "Execution formula", hence the "dynamics of Cut-elimination".

Mathematical essence of multiplicative GoI

- Feedback in a **linear** (monoidal) setting
- Building a free compact closed category from a traced category – "higher-order" vs. "first-order".
- The composition in this free compact closed category (i.e. symmetric feedback) subsumes the "Execution formula", hence the "dynamics of Cut-elimination".

Traced monoidal categories were introduced in: Joyal, André, Ross Street, and Dominic Verity. "Traced monoidal categories" In Mathematical proceedings of the Cambridge Philosophical Society, vol. 119, no. 3, pp. 447-468. Cambridge University Press, 1996.

Mathematical essence of multiplicative GoI

- Feedback in a **linear** (monoidal) setting
- Building a free compact closed category from a traced category – "higher-order" vs. "first-order".
- The composition in this free compact closed category (i.e. symmetric feedback) subsumes the "Execution formula", hence the "dynamics of Cut-elimination".

Traced monoidal categories were introduced in: Joyal, André, Ross Street, and Dominic Verity. "Traced monoidal categories" In Mathematical proceedings of the Cambridge Philosophical Society, vol. 119, no. 3, pp. 447-468. Cambridge University Press, 1996.

Avant la lettre:

- NFGoI built a traced monoidal category from domains and fixpoints (product based)
- In "Games and Full Completeness for MLL" it was observed that composition of history-free strategies was given by a suitable abstraction of the Execution formula (coproduct based)
- I showed these examples in detail to André Joyal during LiCS 1993 in Montreal.

Examples of Traced Monoidal Categories

Three families:

Examples of Traced Monoidal Categories

Three families:

- “Particle style”: the tensor product is coproduct (disjoint union). **Rel**, **Pfn**, **PInj**, etc. Covers the original version of GoI introduced by Girard. Matrices over a Kleene algebra are another example. (The ‘magic formula’ for star is exactly the trace!)

Examples of Traced Monoidal Categories

Three families:

- "Particle style": the tensor product is coproduct (disjoint union). **Rel**, **Pfn**, **PInj**, etc. Covers the original version of GoI introduced by Girard. Matrices over a Kleene algebra are another example. (The 'magic formula' for star is exactly the trace!)

- "Wave style": the tensor product is (categorical) product. This forces the trace to be a fixpoint (Hasegawa, Hyland), hence we are in the realm of **Domain theory**. S.A. and Radha Jagadeesan, 'New Foundations for the Geometry of Interaction' .

Examples of Traced Monoidal Categories

Three families:

- "Particle style": the tensor product is coproduct (disjoint union). **Rel**, **Pfn**, **PInj**, etc. Covers the original version of GoI introduced by Girard. Matrices over a Kleene algebra are another example. (The 'magic formula' for star is exactly the trace!)
- "Wave style": the tensor product is (categorical) product. This forces the trace to be a fixpoint (Hasegawa, Hyland), hence we are in the realm of **Domain theory**. S.A. and Radha Jagadeesan, 'New Foundations for the Geometry of Interaction' .
- "Acausal": Compact closed categories. $(\mathbf{Rel}, \times)$, $(\mathbf{FDVect}, \otimes)$. Physical interpretation: 'Physical Traces', S.A. and Bob Coecke.

Closing the gap

In

Abramsky, Samson. "Retracing some paths in process algebra." In International Conference on Concurrency Theory, pp. 1-17. Springer 1996.

I showed how the basic (multiplicative) part of GoI could be understood in terms of traced monoidal categories and the construction of the free compact closed category (essentially the Int construction of JSV).

Closing the gap

In

Abramsky, Samson. "Retracing some paths in process algebra." In International Conference on Concurrency Theory, pp. 1-17. Springer 1996.

I showed how the basic (multiplicative) part of GoI could be understood in terms of traced monoidal categories and the construction of the free compact closed category (essentially the **Int** construction of JSV).

However, to achieve a full understanding of GoI, more was needed:

- Extending the axiomatic framework to cover the whole of GoI, and computational universality
- GoI does not exactly give a model of Linear Logic (or λ-calculus) in the usual sense – many extensional equalities fail. What **is** it doing?

Closing the gap

In

Abramsky, Samson. "Retracing some paths in process algebra." In International Conference on Concurrency Theory, pp. 1-17. Springer 1996.

I showed how the basic (multiplicative) part of GoI could be understood in terms of traced monoidal categories and the construction of the free compact closed category (essentially the Int construction of JSV).

However, to achieve a full understanding of GoI, more was needed:

- Extending the axiomatic framework to cover the whole of GoI, and computational universality
- GoI does not exactly give a model of Linear Logic (or λ-calculus) in the usual sense – many extensional equalities fail. What **is** it doing?

I outlined an approach to answering these questions in lectures given in Siena and Edinburgh in 1997. This developed into a joint project with Phil and Esfan (who was doing his Ph.D. with Phil), and led to our paper.

Our answers

- We introduce **GoI situations** as an axiomatic framework for the full GoI construction, including exponentials.
- We propose **combinatory algebra** as the appropriate intensional setting to capture what GoI does.
- We introduce **linear combinatory algebras**, show that these give rise to standard, computationally universal combinatory algebras, and prove our **main result**:

Theorem

Every GoI situation gives rise to a linear combinatory algebra.

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

- Conceptually, they are algebras of **closed terms**.

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

- Conceptually, they are algebras of **closed terms**.
- Thus the basic axioms are weaker than those of $\lambda\beta$ - they do not form λ-algebras.

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

- Conceptually, they are algebras of **closed terms**.
- Thus the basic axioms are weaker than those of $\lambda\beta$ - they do not form λ-algebras.
- The equations of basic combinatory algebra are **just good enough** to show that all partial recursive functions can be represented (acting on numerals encoded as combinator terms).

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

- Conceptually, they are algebras of **closed terms**.
- Thus the basic axioms are weaker than those of $\lambda\beta$ - they do not form λ-algebras.
- The equations of basic combinatory algebra are **just good enough** to show that all partial recursive functions can be represented (acting on numerals encoded as combinator terms).
- A beautifully succinct, purely equational presentation of computability.

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

- Conceptually, they are algebras of **closed terms**.
- Thus the basic axioms are weaker than those of $\lambda\beta$ - they do not form λ-algebras.
- The equations of basic combinatory algebra are **just good enough** to show that all partial recursive functions can be represented (acting on numerals encoded as combinator terms).
- A beautifully succinct, purely equational presentation of computability.
- The Curry correspondence - they correspond to Hilbert-style axiomatizations of logics.

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

- Conceptually, they are algebras of **closed terms**.
- Thus the basic axioms are weaker than those of $\lambda\beta$ - they do not form λ-algebras.
- The equations of basic combinatory algebra are **just good enough** to show that all partial recursive functions can be represented (acting on numerals encoded as combinator terms).
- A beautifully succinct, purely equational presentation of computability.
- The Curry correspondence - they correspond to Hilbert-style axiomatizations of logics.
- They also form the basis of realizability, where extensional theories can be built over these intensional algebras.

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

- Conceptually, they are algebras of **closed terms**.
- Thus the basic axioms are weaker than those of $\lambda\beta$ - they do not form λ-algebras.
- The equations of basic combinatory algebra are **just good enough** to show that all partial recursive functions can be represented (acting on numerals encoded as combinator terms).
- A beautifully succinct, purely equational presentation of computability.
- The Curry correspondence - they correspond to Hilbert-style axiomatizations of logics.
- They also form the basis of realizability, where extensional theories can be built over these intensional algebras.
- They are handy for substructural logics, e.g. BCI and BCK.

Combinatory algebra: Ugh?

Combinatory algebras are not so bad:

- Conceptually, they are algebras of **closed terms**.
- Thus the basic axioms are weaker than those of $\lambda\beta$ - they do not form λ-algebras.
- The equations of basic combinatory algebra are **just good enough** to show that all partial recursive functions can be represented (acting on numerals encoded as combinator terms).
- A beautifully succinct, purely equational presentation of computability.
- The Curry correspondence - they correspond to Hilbert-style axiomatizations of logics.
- They also form the basis of realizability, where extensional theories can be built over these intensional algebras.
- They are handy for substructural logics, e.g. BCI and BCK.

Crucially, they capture the right level of intensionality for GoI, explaining why it suffices for computation.

The Curry combinators

Curry's original set of combinators was **B**, **C**, **K**, and **W**:

$$
\begin{aligned}
\mathbf{B} \cdot x \cdot y \cdot z &= x \cdot (y \cdot z) \\
\mathbf{C} \cdot x \cdot y \cdot z &= x \cdot z \cdot y \\
\mathbf{W} \cdot x \cdot y &= x \cdot y \cdot y
\end{aligned}
$$

These are equivalent to the usual **SK**-combinators, thus functionally complete, computationally universal, etc.

The Curry combinators

Curry's original set of combinators was **B**, **C**, **K**, and **W**:

$$\begin{array}{rcl} \mathbf{B} \cdot x \cdot y \cdot z & = & x \cdot (y \cdot z) \\ \mathbf{C} \cdot x \cdot y \cdot z & = & x \cdot z \cdot y \\ \mathbf{W} \cdot x \cdot y & = & x \cdot y \cdot y \end{array}$$

These are equivalent to the usual **SK**-combinators, thus functionally complete, computationally universal, etc.

They have the following principal types:

$$\begin{array}{lcll} \mathbf{I} & : & \alpha \to \alpha & \text{Axiom} \\ \mathbf{B} & : & (\beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma & \text{Cut} \\ \mathbf{C} & : & (\alpha \to \beta \to \gamma) \to \beta \to \alpha \to \gamma & \text{Exchange} \\ \mathbf{K} & : & \alpha \to \beta \to \alpha & \text{Weakening} \\ \mathbf{W} & : & (\alpha \to \alpha \to \beta) \to \alpha \to \beta & \text{Contraction} \end{array}$$

Linear Combinatory Algebras

These are structures $(A, \cdot, \mathbf{!})$, where in addition to the applicative structure $(A, \cdot)$ we have a unary operator $!$.

Linear Combinatory Algebras

These are structures $(A, \cdot, \mathbf{!})$, where in addition to the applicative structure $(A, \cdot)$ we have a unary operator !.

Logically, this corresponds to the Necessitation rule for modal logic:

$$\frac{A}{\Box A}$$

Linear Combinatory Algebras

These are structures $(A, \cdot, !)$, where in addition to the applicative structure $(A, \cdot)$ we have a unary operator $!$.

Logically, this corresponds to the Necessitation rule for modal logic:

$$\frac{A}{\Box A}$$

There are constants $B, C, I, K, W, D, \delta, F$ satisfying:

1. $Bxyz = x(yz)$ Composition, Cut
2. $Cxyz = (xz)y$ Exchange
3. $Ix = x$ Identity
4. $Kx!y = x$ Weakening
5. $Wx!y = x!y!y$ Contraction
6. $D!x = x$ Dereliction
7. $\delta!x = !!x$ Comultiplication
8. $F!x!y = !(xy)$ Monoidal Functoriality

LCA and Linear Logic

The principal types:

1. $B : (\beta \multimap \gamma) \multimap (\alpha \multimap \beta) \multimap \alpha \multimap \gamma$
2. $C : (\alpha \multimap \beta \multimap \gamma) \multimap (\beta \multimap \alpha \multimap \gamma)$
3. $I : \alpha \multimap \alpha$
4. $K : \alpha \multimap \,!\beta \multimap \alpha$
5. $W : (!\alpha \multimap \,!\alpha \multimap \beta) \multimap \,!\alpha \multimap \beta$
6. $D : \,!\alpha \multimap \alpha$
7. $\delta : \,!\alpha \multimap \,!!\alpha$
8. $F : \,!(\alpha \multimap \beta) \multimap \,!\alpha \multimap \,!\beta.$

LCA and Linear Logic

The principal types:

1. $B : (\beta \multimap \gamma) \multimap (\alpha \multimap \beta) \multimap \alpha \multimap \gamma$
2. $C : (\alpha \multimap \beta \multimap \gamma) \multimap (\beta \multimap \alpha \multimap \gamma)$
3. $I : \alpha \multimap \alpha$
4. $K : \alpha \multimap \mathbf{!}\beta \multimap \alpha$
5. $W : (\mathbf{!}\alpha \multimap \mathbf{!}\alpha \multimap \beta) \multimap \mathbf{!}\alpha \multimap \beta$
6. $D : \mathbf{!}\alpha \multimap \alpha$
7. $\delta : \mathbf{!}\alpha \multimap \mathbf{!!}\alpha$
8. $F : \mathbf{!}(\alpha \multimap \beta) \multimap \mathbf{!}\alpha \multimap \mathbf{!}\beta.$

This corresponds to a Hilbert-style axiomatization of $\multimap, \mathbf{!}$ fragment of Linear Logic.

LCA and Linear Logic

The principal types:

1. $B : (\beta \multimap \gamma) \multimap (\alpha \multimap \beta) \multimap \alpha \multimap \gamma$
2. $C : (\alpha \multimap \beta \multimap \gamma) \multimap (\beta \multimap \alpha \multimap \gamma)$
3. $I : \alpha \multimap \alpha$
4. $K : \alpha \multimap \mathbf{!}\beta \multimap \alpha$
5. $W : (\mathbf{!}\alpha \multimap \mathbf{!}\alpha \multimap \beta) \multimap \mathbf{!}\alpha \multimap \beta$
6. $D : \mathbf{!}\alpha \multimap \alpha$
7. $\delta : \mathbf{!}\alpha \multimap \mathbf{!!}\alpha$
8. $F : \mathbf{!}(\alpha \multimap \beta) \multimap \mathbf{!}\alpha \multimap \mathbf{!}\beta.$

This corresponds to a Hilbert-style axiomatization of $\multimap, \mathbf{!}$ fragment of Linear Logic.

A combinatory version of the Girard translation results in:

Theorem

Given an LCA, we can construct a standard CA, where $x \bullet_s y := x \bullet \mathbf{!} y$.

GoI situations

A *GoI Situation* is a triple $(\mathbb{C}, T, U)$ where:

- $\mathbb{C}$ is a traced symmetric monoidal category
- $T : \mathbb{C} \longrightarrow \mathbb{C}$ is a traced symmetric monoidal functor with the following retractions (which are monoidal natural transformations):
  1. $e : TT \lhd T : e'$ (Comultiplication)
  2. $d : Id \lhd T : d'$ (Dereliction)
  3. $c : T \otimes T \lhd T : c'$ (Contraction)
  4. $w : \mathcal{K}_I \lhd T : w'$ (Weakening), where $\mathcal{K}_I$ is the constant I functor.

GoI situations

A *GoI Situation* is a triple $(\mathbb{C}, T, U)$ where:

- $\mathbb{C}$ is a traced symmetric monoidal category
- $T : \mathbb{C} \longrightarrow \mathbb{C}$ is a traced symmetric monoidal functor with the following retractions (which are monoidal natural transformations):
  1. $e : TT \lhd T : e'$ (Comultiplication)
  2. $d : Id \lhd T : d'$ (Dereliction)
  3. $c : T \otimes T \lhd T : c'$ (Contraction)
  4. $w : \mathcal{K}_I \lhd T : w'$ (Weakening), where $\mathcal{K}_I$ is the constant I functor.

We apply the GoI (or Int) construction to $\mathbf{C}$ to get a compact closed category $\mathcal{G}(\mathbf{C})$. We can define $\mathbf{!}$ on $\mathcal{G}(\mathbf{C})$ using the functor T.

GoI situations

A *GoI Situation* is a triple $(\mathbb{C}, T, U)$ where:

- $\mathbb{C}$ is a traced symmetric monoidal category
- $T : \mathbb{C} \longrightarrow \mathbb{C}$ is a traced symmetric monoidal functor with the following retractions (which are monoidal natural transformations):
    1. $e : TT \lhd T : e'$ (Comultiplication)
    2. $d : Id \lhd T : d'$ (Dereliction)
    3. $c : T \otimes T \lhd T : c'$ (Contraction)
    4. $w : \mathcal{K}_I \lhd T : w'$ (Weakening), where $\mathcal{K}_I$ is the constant I functor.

We apply the GoI (or Int) construction to $\mathbf{C}$ to get a compact closed category $\mathcal{G}(\mathbf{C})$. We can define $\mathbf{!}$ on $\mathcal{G}(\mathbf{C})$ using the functor T.

Because the monoidal comonadic structure of T holds only "up to retractions", we get only a "weak linear category", in which the naturality requirements are weakened to **pointwise** naturality, i.e. with respect to arrows $I \longrightarrow A$.

GoI situations

A *GoI Situation* is a triple $(\mathbb{C}, T, U)$ where:

- $\mathbb{C}$ is a traced symmetric monoidal category
- $T : \mathbb{C} \longrightarrow \mathbb{C}$ is a traced symmetric monoidal functor with the following retractions (which are monoidal natural transformations):
  1. $e : TT \lhd T : e'$ (Comultiplication)
  2. $d : Id \lhd T : d'$ (Dereliction)
  3. $c : T \otimes T \lhd T : c'$ (Contraction)
  4. $w : \mathcal{K}_I \lhd T : w'$ (Weakening), where $\mathcal{K}_I$ is the constant I functor.

We apply the GoI (or Int) construction to $\mathbf{C}$ to get a compact closed category $\mathcal{G}(\mathbf{C})$. We can define $\mathbf{!}$ on $\mathcal{G}(\mathbf{C})$ using the functor T.

Because the monoidal comonadic structure of T holds only "up to retractions", we get only a "weak linear category", in which the naturality requirements are weakened to **pointwise** naturality, i.e. with respect to arrows $I \longrightarrow A$.

This matches the idea that in combinatory logic, in general λ-equations hold only for **closed terms**.

Completing the construction

We then show that weak linear categories give rise to linear combinatory algebras.
To get a type-free model, we assume a "reflexive object" U with retractions $U \otimes U \lhd U$, $I \lhd U$, $TU \lhd U$.

GoI Situation

$\Downarrow$ construct $\mathcal{G}(\mathbb{C})$

Weak Linear Category

$\Downarrow$ take $\mathcal{G}(\mathbb{C})(I, U)$

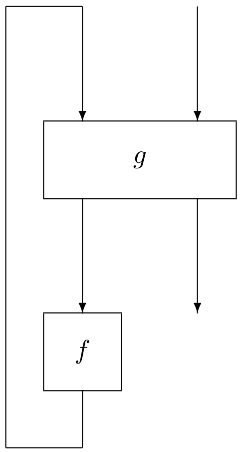
Linear Combinatory Algebra

$\Downarrow$ by standardization

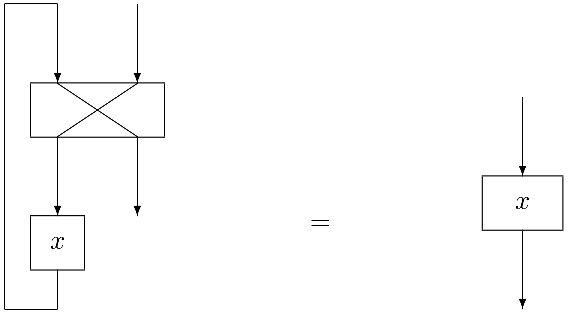
Combinatory Algebra

Application pictorially

This is just a special case of composition:

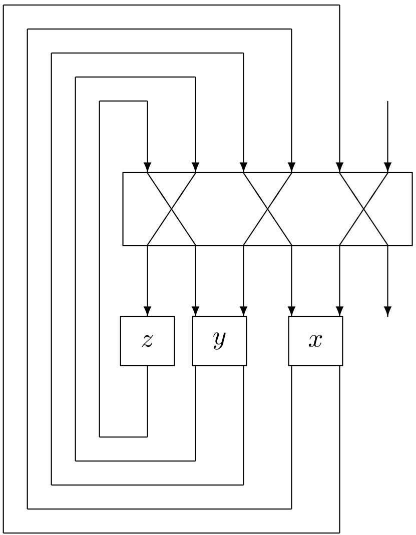


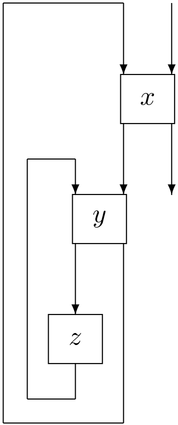
The I combinator



$$\mathbf{I} \bullet x = x$$

The B combinator





The W combinator

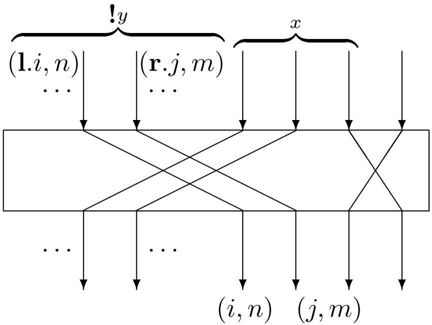
$$Wx\mathbf{!}y = x\mathbf{!}y\mathbf{!}y$$

Hilbert Hotelling:

$$\mathsf{p} : \mathbb{N} \times \mathsf{Pos} \xrightarrow{\cong} \mathsf{Pos}$$

Translation between dialects

W



Further Developments

- Phil and Esfan wrote a further series of papers on GoI. Topics include: typed GoI, unique decomposition categories, partial traces. Tutorial in *New Structures in Physics* (2010).
- Peter Hines has found many fascinating connections of the Hilbert hotel aspects, e.g. to the Thompson group, strictification of coherence, the Collatz conjecture, etc.
- Mark Lawson has studied algebraic aspects.
- Naohiko Hoshino, Ichiro Hasuo and Koko Muroya have studied extensions such as higher-order quantum GoI and memoryful GoI.
- Kuko Muroya and Dan Ghica developed a dynamic GoI.
- Carsten Furhmann and David Pym developed a categorical model of Classical logic using GoI ideas.
- I gave a very concrete account of GoI as the basis for a **structural approach to reversible computation**. This shows that one can build a combinatory algebra of **partial involutions** (graph matchings) in a very simple fashion.
- An interesting question arising from this work has been answered in recent work by Alberto Ciaffaglione, Pietro Di Gianantonio, Furio Honsell, Marina Lenisa, and Ivan Scagnetto.


Paintings