Logic and Geometry of Agents

Samson Abramsky
Oxford University Computing Laboratory
http://web.comlab.ox.ac.uk/oucl/work/samson.abramsky/

Agent-based Modelling and Simulation

> ‘Complex Systems in which agents interact with each other and their environment using simple local rules.
>
> Building a sound and widely applicable theory for such systems will require an inter-disciplinary approach and the development of new mathematical and computational concepts.’

My perspective from Theoretical Computer Science: connections with logic, ‘geometry of interaction’, and physics (reversible and quantum computing).

Compositionality

A methodological principle from Computer Science (and Logic) of **major** potential importance for mathematical modelling throughout the sciences.

Traditional aproach: whole-system (monolithic) analysis of given systems. Key structuring templates, e.g. 'Find the Hamiltonian'.

Compositional approach: start with a fixed set of basic (simple) building blocks, and **constructions** for building new (in general more complex) systems out of given sub-systems, and build up the required complex system with these.

The algebraic view:

$$S \; = \; \omega(S_1, \ldots, S_n)$$

The logical view:

$$\frac{S_1 \models \phi_1, \ldots, S_n \models \phi_n}{\omega(S_1, \ldots, S_n) \models \phi}$$

Key Points

Not just one-level composition of basic agents:

$$S = \bigparallel_{i \in I} a_i$$

but **hierarchical**:

$$S = \omega_1(\omega_2(a_1, a_2, a_3), \omega_1(a_4, a_5)).$$

We track the properties of the sub-compound systems all the way up (or down).

The repertoire of system constructors is also an important aspect of the modelling.

This paradigm is already starting to be applied in

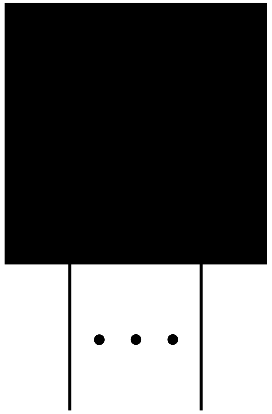
- quantum computing (quantum programming languages)
- biological modelling (process calculi)
- business modelling (idem)

and will surely be applied more widely and deeply, in economics as well as physical and biological sciences.

This kind of modelling carries in its train a range of **analytical techniques** : types, semantics, verification, model-checking, etc.

Computation in the Age of the Internet

Instead of isolated systems, with rudimentary interactions with their environment, the standard unit of description or design becomes a **process** or **agent**, the essence of whose behaviour is **how it interacts** with its environment.

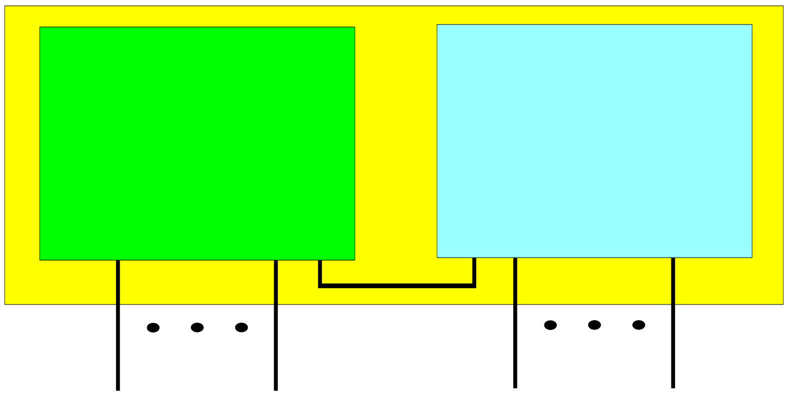


Who is the System? Who is the Environment?
This **symmetry** will lead us to a key **duality**, and a deep connection to logic.

Interaction

Complex behaviour arises as the global effect of a **system** of **interacting agents** (or processes).

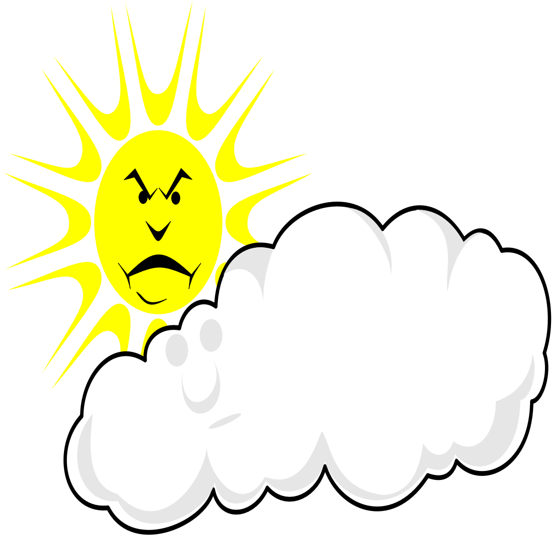
The key building block is the agent. The key operation is **interaction** – plugging agents together so that they interact with each other

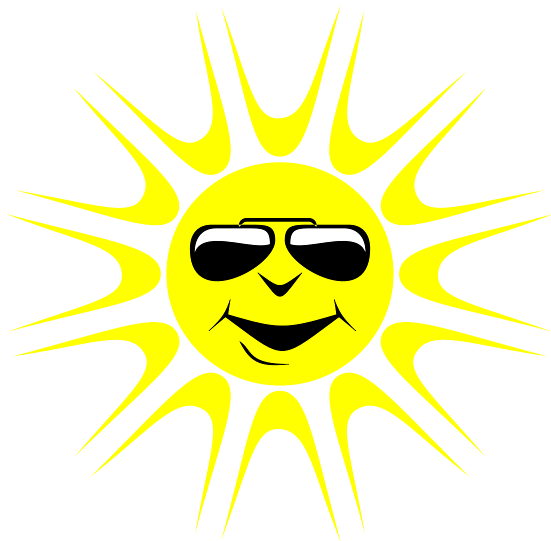


The static conception of logic

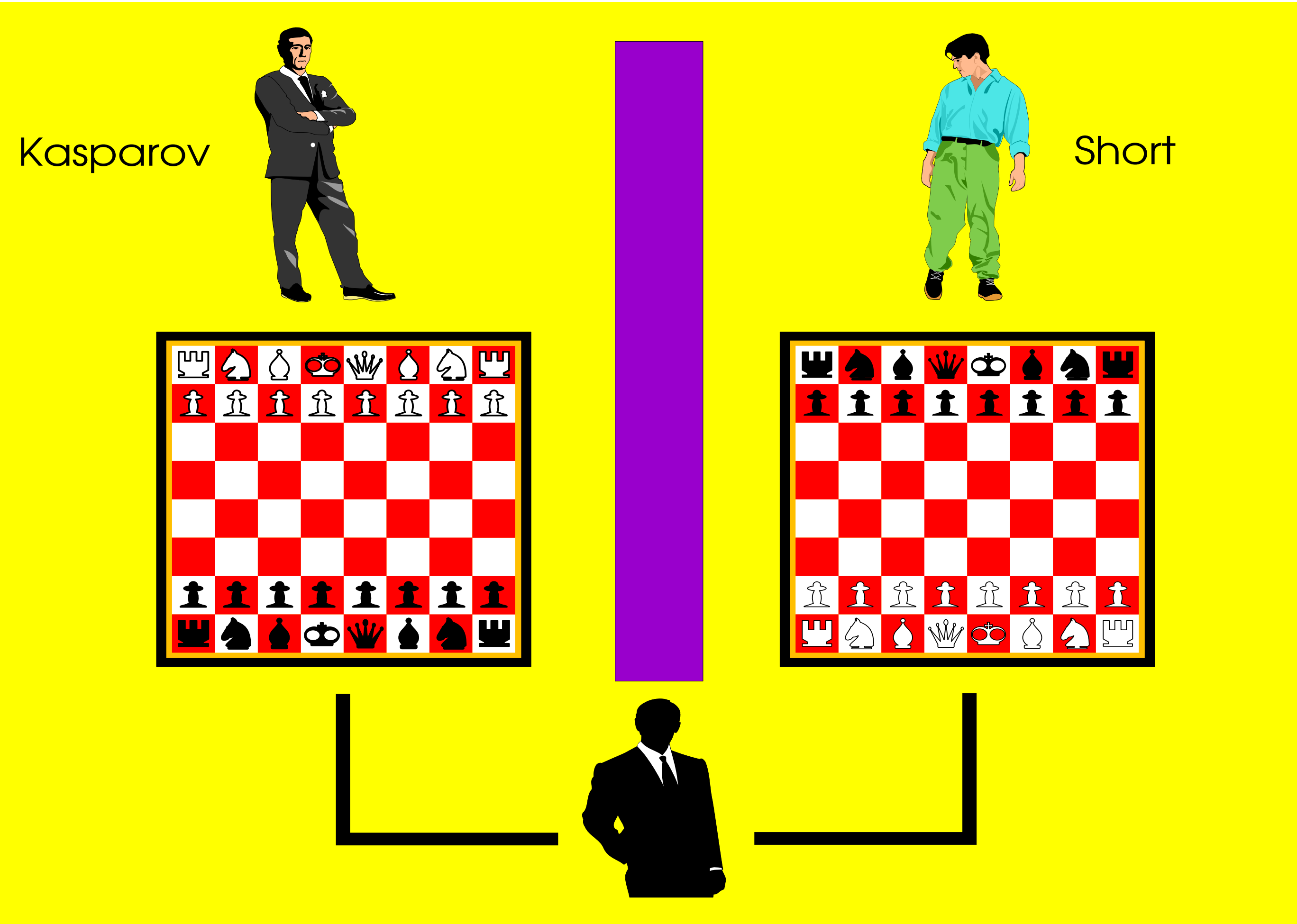
$$A \vee \neg A$$

"It is raining **or** it is not raining"—truth-functional semantics.
(**Tertium non datur?**)

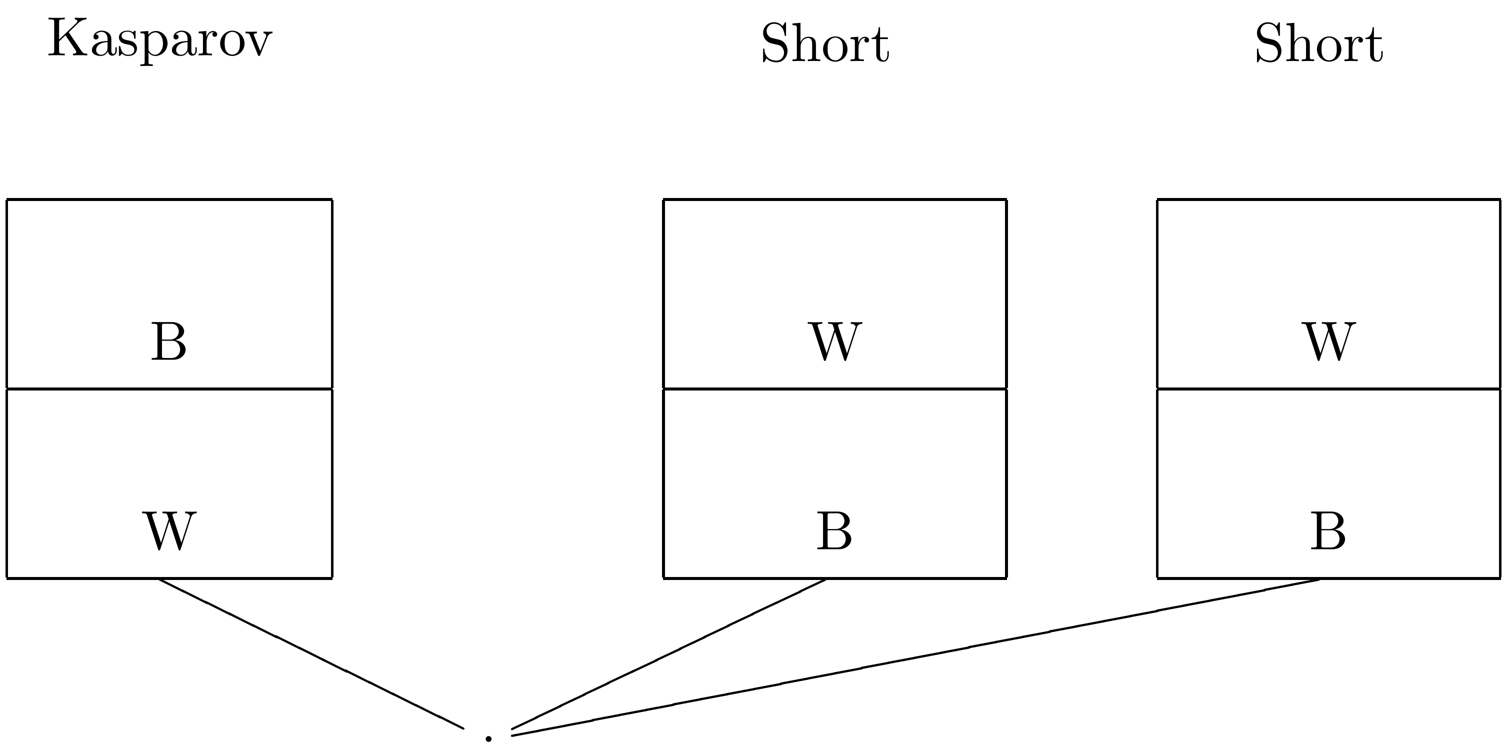




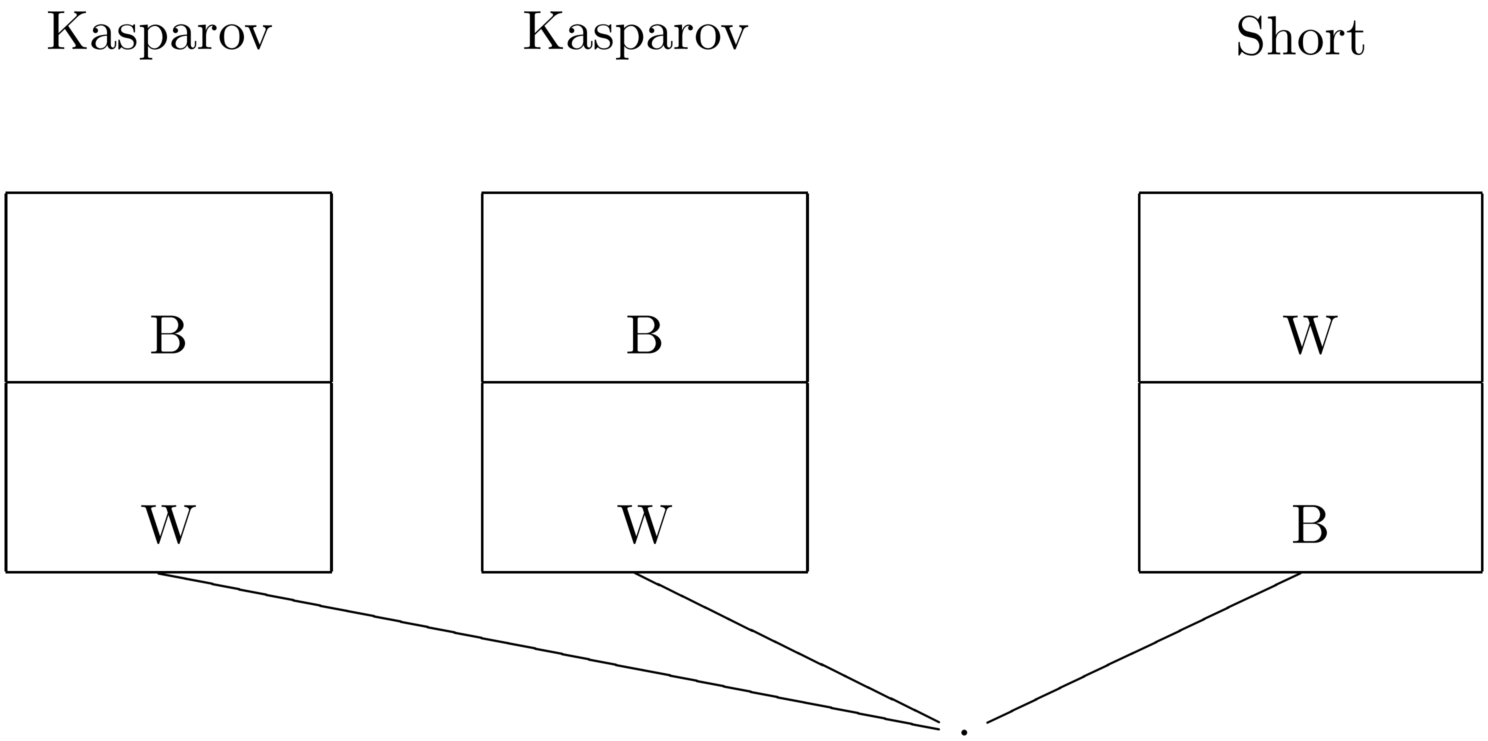
The Copy-Cat Strategy



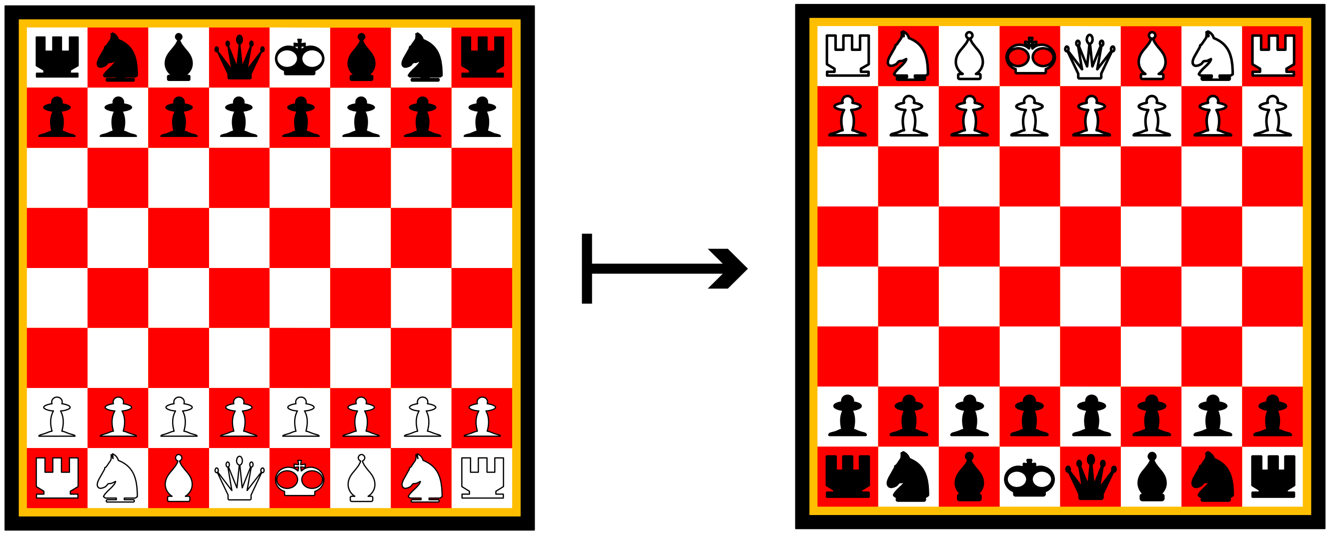
Does Copy-Cat still work here?



And here?

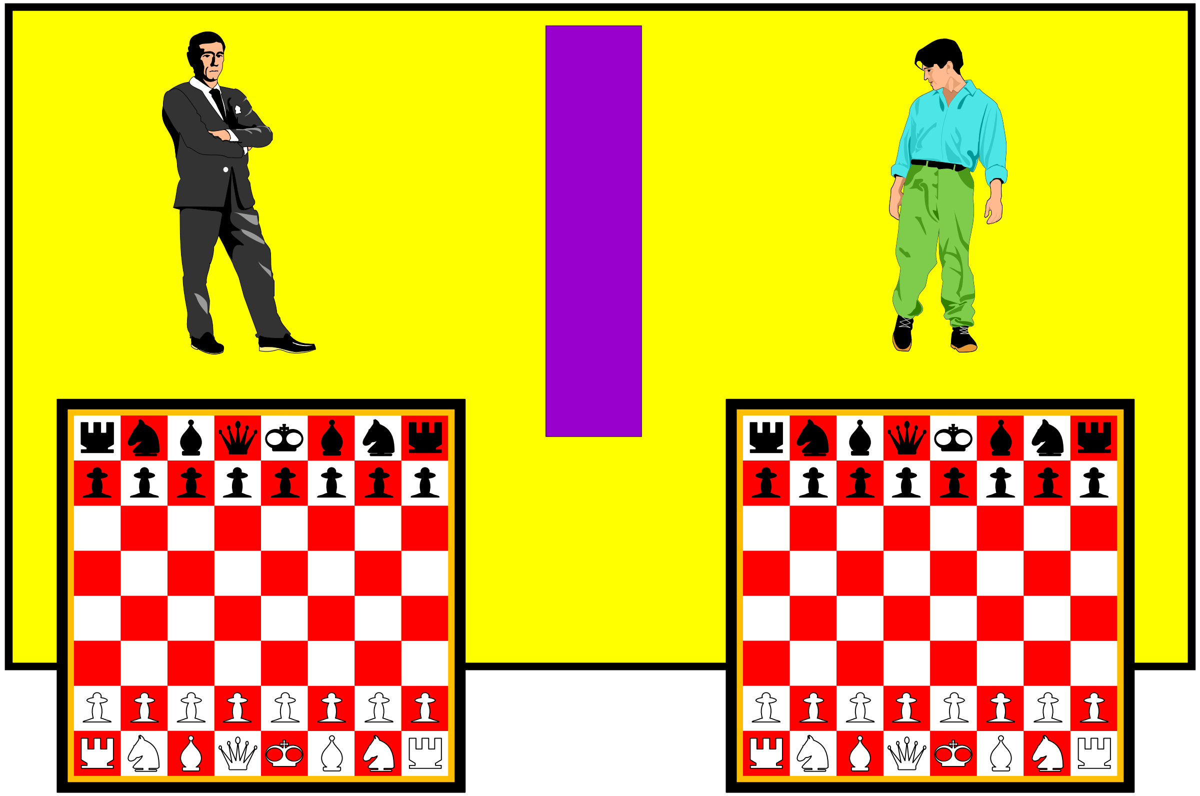


Duality—“Linear Negation”



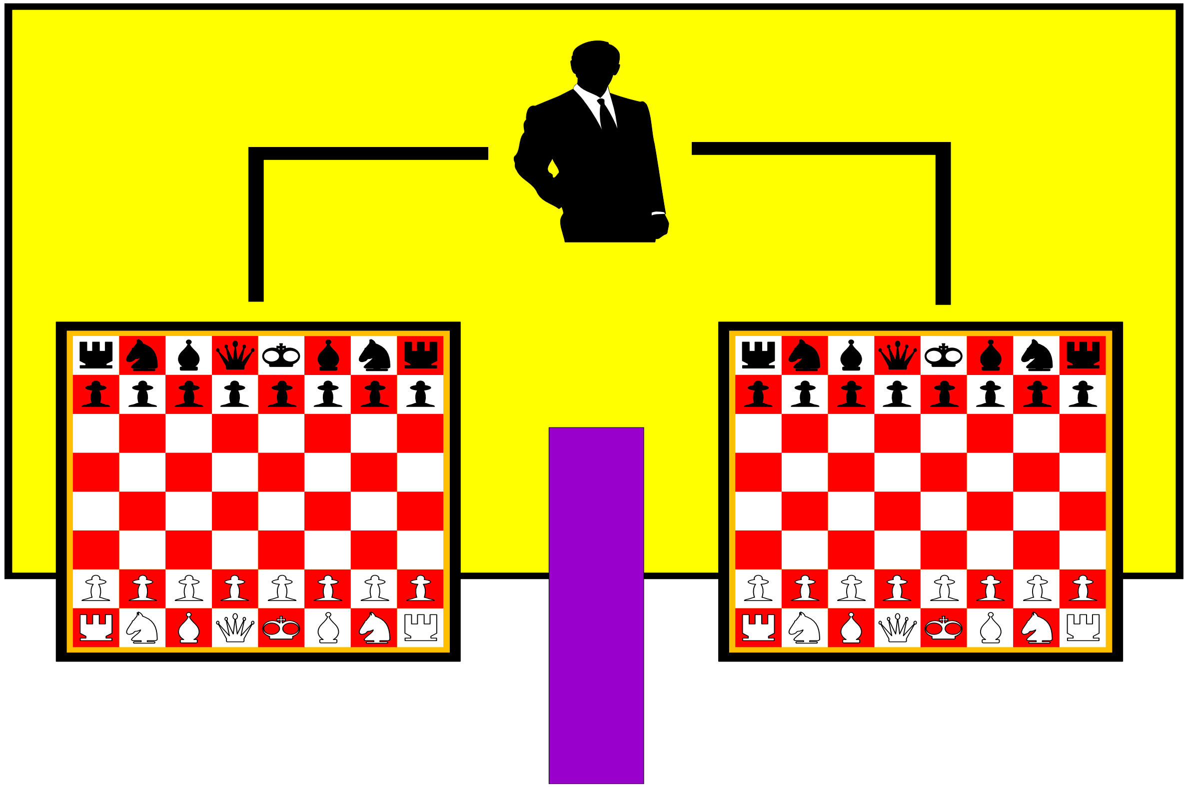
$$A^{\perp\perp} = A.$$

Tensor — "Linear conjunction" (pure tensors)



$$A \otimes B$$

Par — “Linear disjunction” (entangled states)



$A ⅋ B$

Logical structure

We can define $A \multimap B$ ("Linear implication") by

$$A \multimap B \equiv A^{\perp} \parr B$$

$$(\textit{cf.}\ A \supset B \equiv \neg A \vee B.)$$

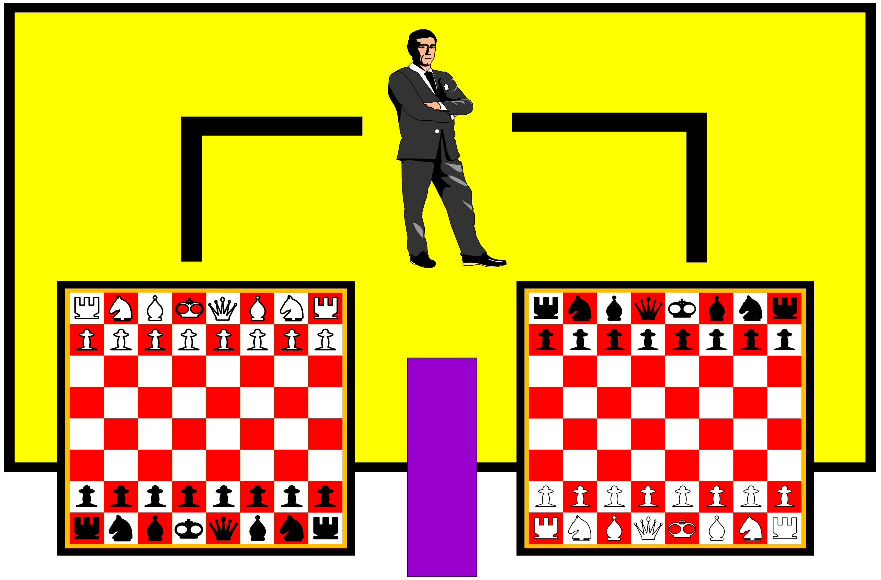
The Copy-cat strategy is a winning strategy for $A^{\perp} \parr A \equiv A \multimap A$.

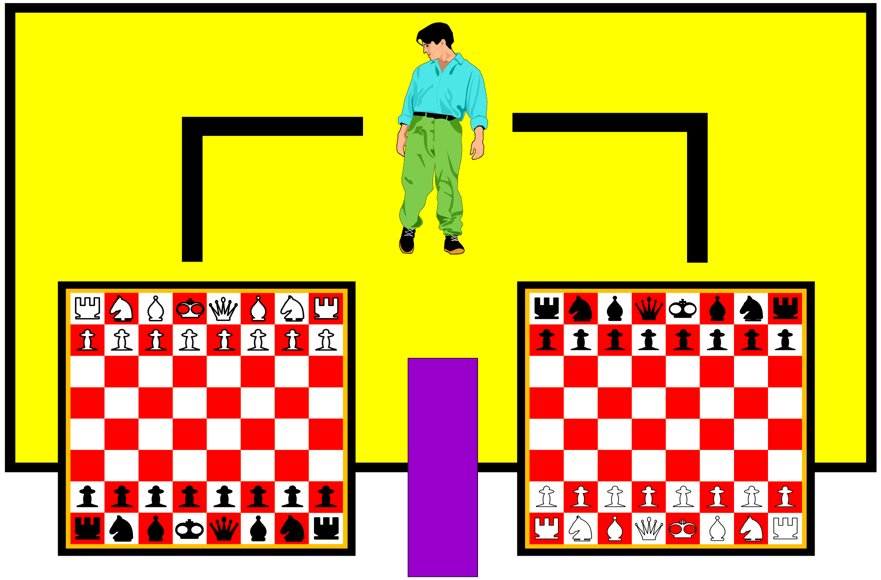
De Morgan duality:

$$\begin{aligned} (A \otimes B)^{\perp} &= A^{\perp} \parr B^{\perp} \\ (A \parr B)^{\perp} &= A^{\perp} \otimes B^{\perp} \end{aligned}$$

Interaction

Constructors create "potentials" for interaction; the operation of plugging modules together so that they can communicate releases this potential into actual computation.



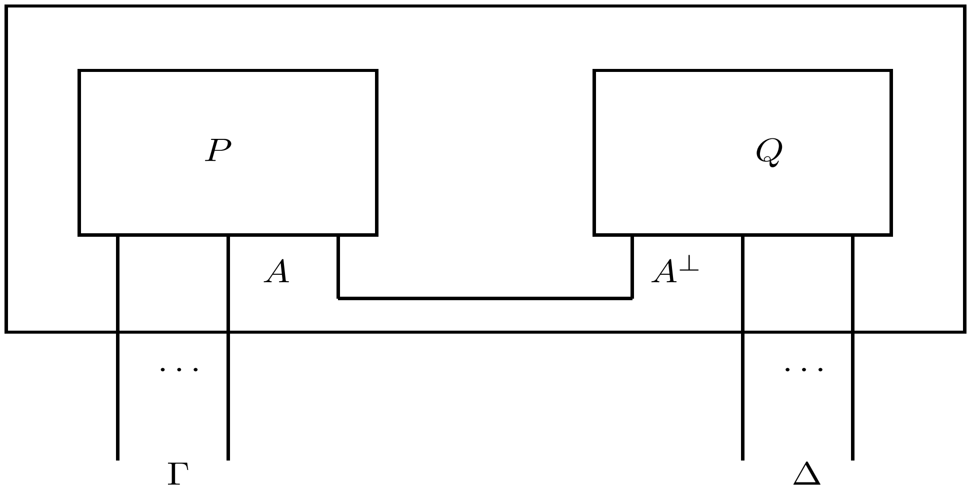


$A^{\perp} \parr B \;\equiv\; A \multimap B$ $\qquad$ $B^{\perp} \parr C \;\equiv\; B \multimap C$

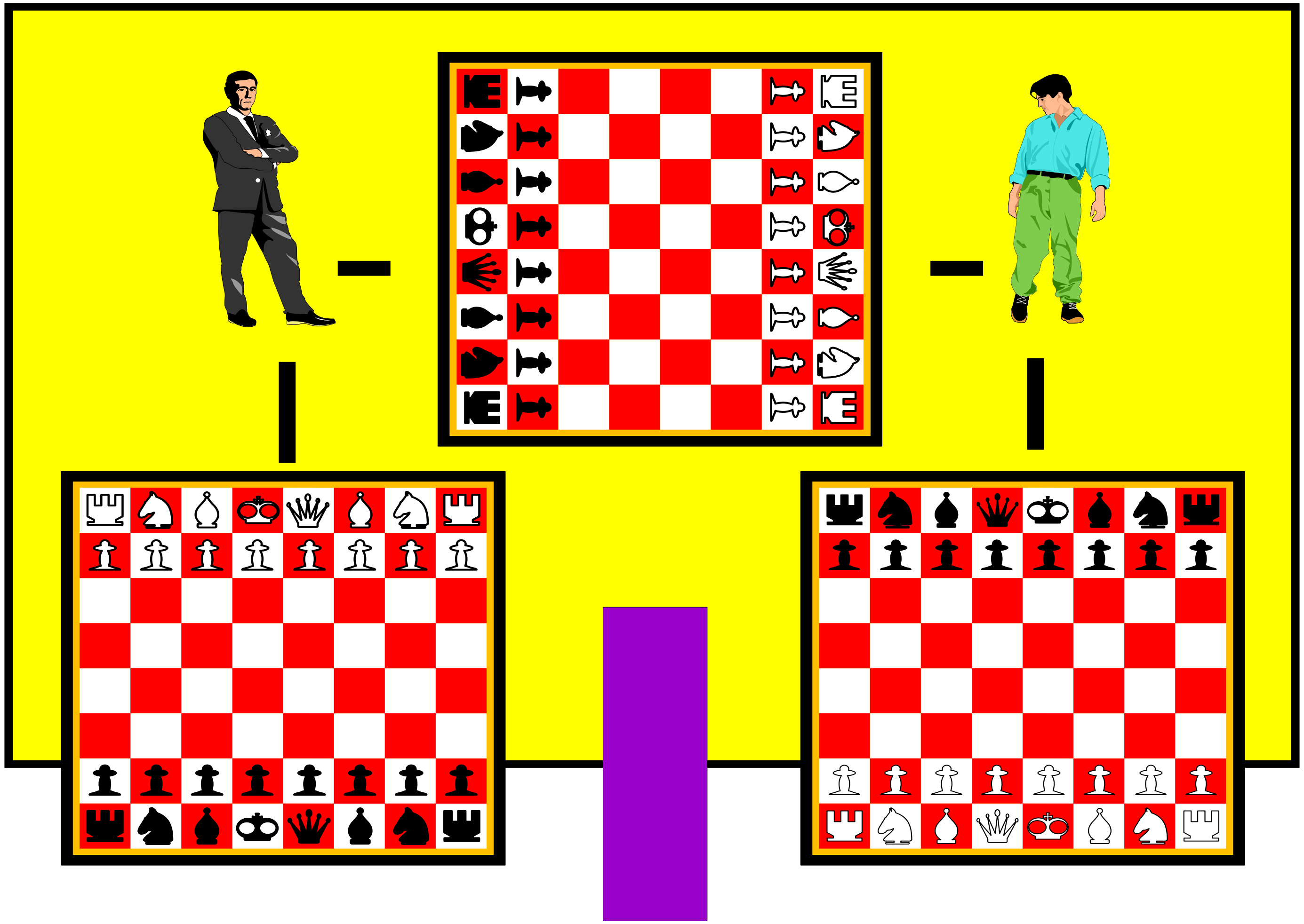
Composition

$$\frac{A \xrightarrow{\text{Gary}} B \xrightarrow{\text{Nigel}} C}{A \xrightarrow{\text{Gary;Nigel}} C}$$

Cut: $$\frac{\vdash \Gamma, A \quad \vdash \Delta, A^{\perp}}{\Gamma, \Delta}$$



Composition as Interaction



Categorical Structure

Games and strategies organize themselves naturally into categories with considerable structure (symmetric monoidal closed, products, certain monoidal comonads etc.)

Note that **games** are **objects** (types); **strategies** (*i.e.* agents) are **morphisms**; **composition** is **interaction**!

This paradigm for finding mathematical structure for interaction has proved quite fruitful over the past decade. In particular, there has been extensive work in Game Semantics, and now Algorithmic Game Semantics.

Emergent Logic

'Particle-style Geometry of Interaction'

- Simple formalization, underlying geometry, 'particle dynamics'.
- **Concrete models** of **higher-order phenomena**.
- Embodies Logic–Computation–Physics connection.
- Axiomatic setting: **traced monoidal categories**. Covers deterministic, non-deterministic, probabilistic and quantum forms of interaction.

Combinatory Logic

Application: $x \cdot y$.
(We write $xy_1 \cdots y_n$ for $(\cdots(x \cdot y_1) \cdots) \cdot y_n))$.

Combinators $\mathbf{S}$, $\mathbf{K}$:

$$
\begin{aligned}
\mathbf{S}xyz &= xz(yz) \\
\mathbf{K}xy &= x
\end{aligned}
$$

(We can define $\mathbf{I} \equiv \mathbf{SKK}$, satisfying $\mathbf{I}x = x$).

Functional or 'bracket' abstraction:
$[x]t$ such that $([x]t)u = t[u/x]$.

$$
\begin{aligned}
[x]x &= \mathbf{I} \\
[x]y &= \mathbf{K}y \qquad (x \neq y) \\
[x]tu &= \mathbf{S}([x]t)([x]u)
\end{aligned}
$$

The Curry Combinators: B, C, K, W

$$\begin{aligned} \mathbf{B}xyz &= x(yz) \\ \mathbf{C}xyz &= xzy \\ \mathbf{W}xy &= xyy \end{aligned}$$

Principal types:

$\mathbf{I}$	:	$\alpha \to \alpha$	Axiom
$\mathbf{B}$	:	$(\beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$	Cut
$\mathbf{C}$	:	$(\alpha \to \beta \to \gamma) \to \beta \to \alpha \to \gamma$	Exchange
$\mathbf{K}$	:	$\alpha \to \beta \to \alpha$	Weakening
$\mathbf{W}$	:	$(\alpha \to \alpha \to \beta) \to \alpha \to \beta$	Contraction

Curry's analysis of substitution is close to Gentzen's analysis of proofs.

Computational Power

In Combinatory Logic, all partial recursive functions are **numeralwise representable**.

This means that we can define numeral systems representing each number n as a term $\bar{n}$ such that, for every recursive function

$$f : \mathbb{N} \longrightarrow \mathbb{N}$$

there is a term t satisfying:

$$\forall n \in \mathbb{N}.\ t\bar{n} = \bar{m} \iff f(n) = m.$$

A Combinatory Algebra of Partial Involutions

We model simple 'history-free' interactive processes by partial functions

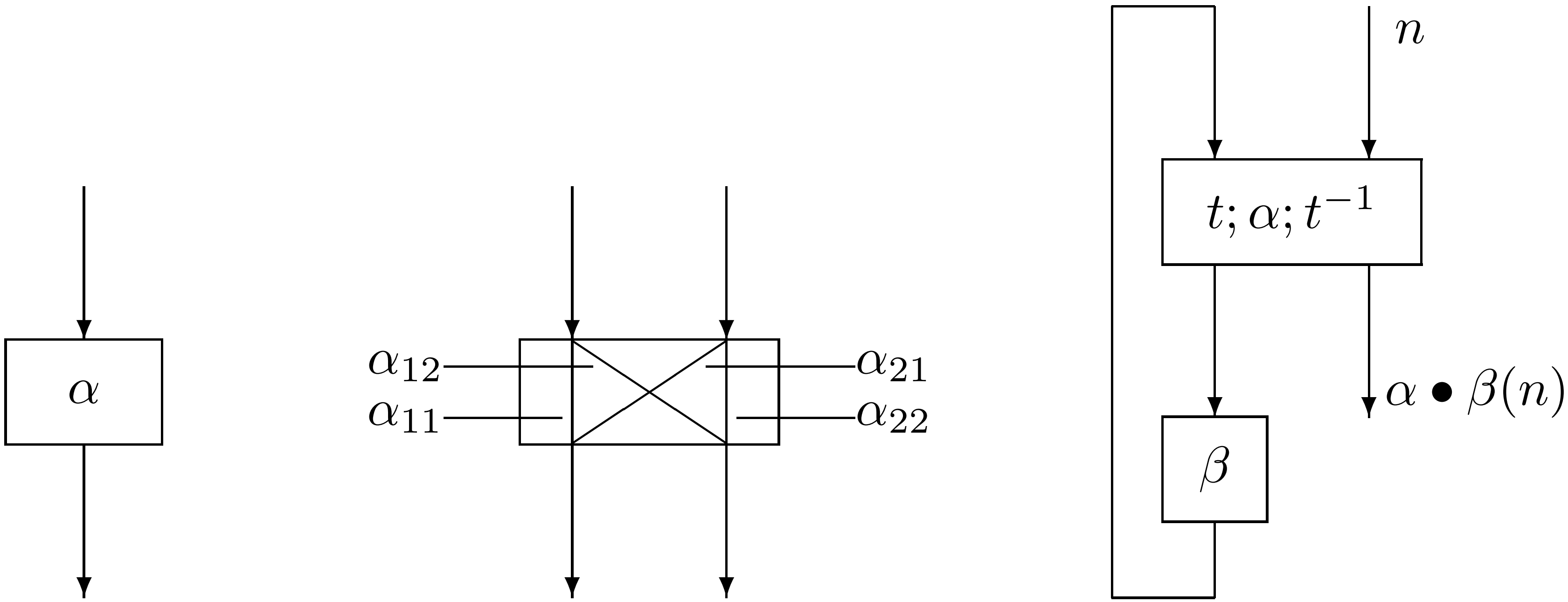
$$\mathbb{N} \rightharpoonup \mathbb{N}.$$

We start by fixing an injective *coding* function t:

$$t : \mathbb{N} + \mathbb{N} \longrightarrow \mathbb{N}.$$

This is used in order to define application, it allows us to transform an one-input/one-output function into a two-input/two-output function.

Geometrical representation of application



Formally:

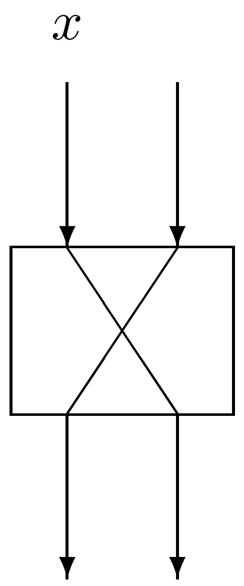
$$\alpha \bullet \beta = \alpha_{22} \cup \alpha_{21} ; (\beta ; \alpha_{11})^{\star} ; \beta ; \alpha_{12} \ ,$$

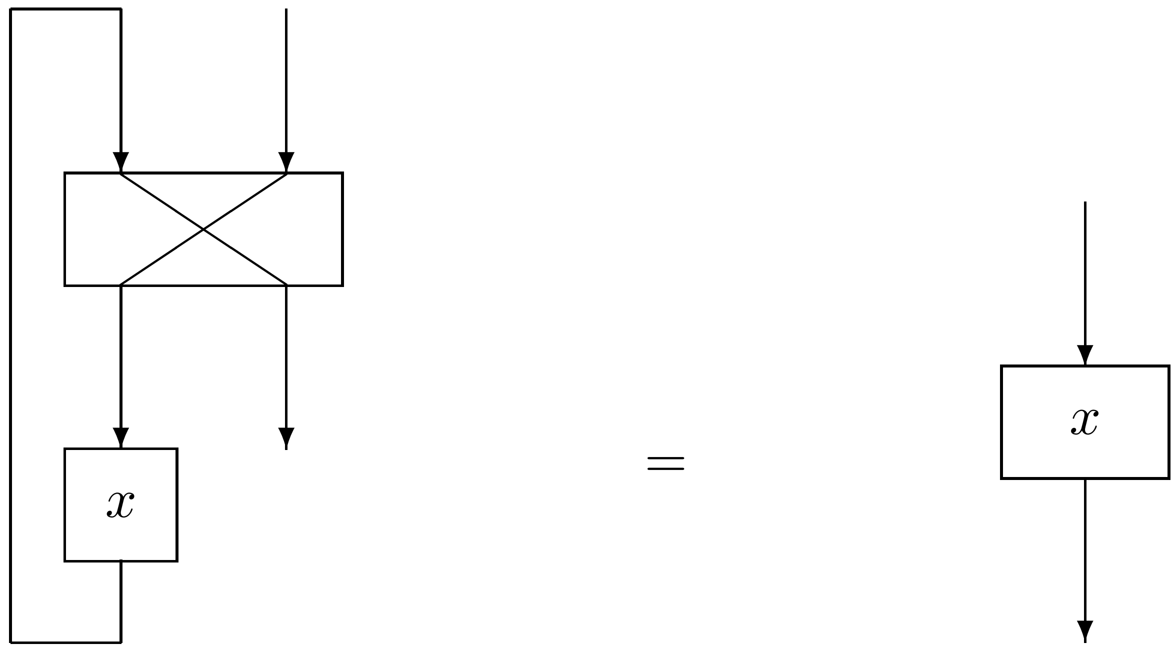
where

$$(\beta ; \alpha_{11})^{\star} = \bigcup_{n \geq 0} (\beta ; \alpha_{11})^{n}.$$

The Identity combinator

I

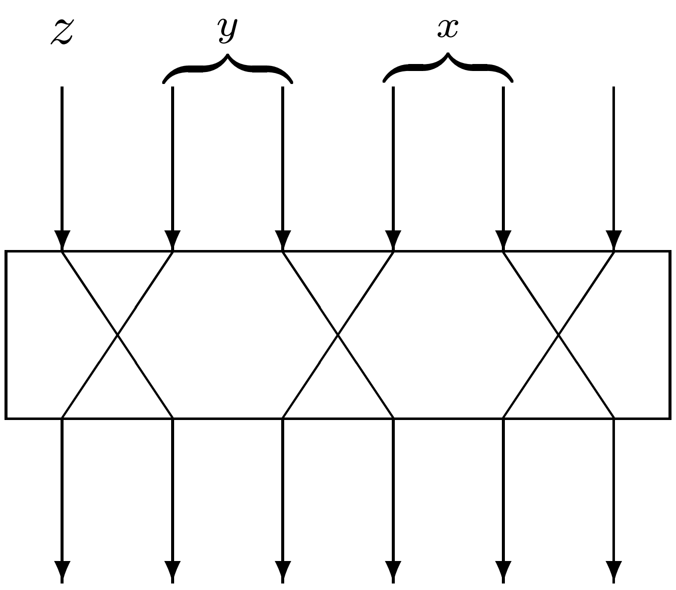


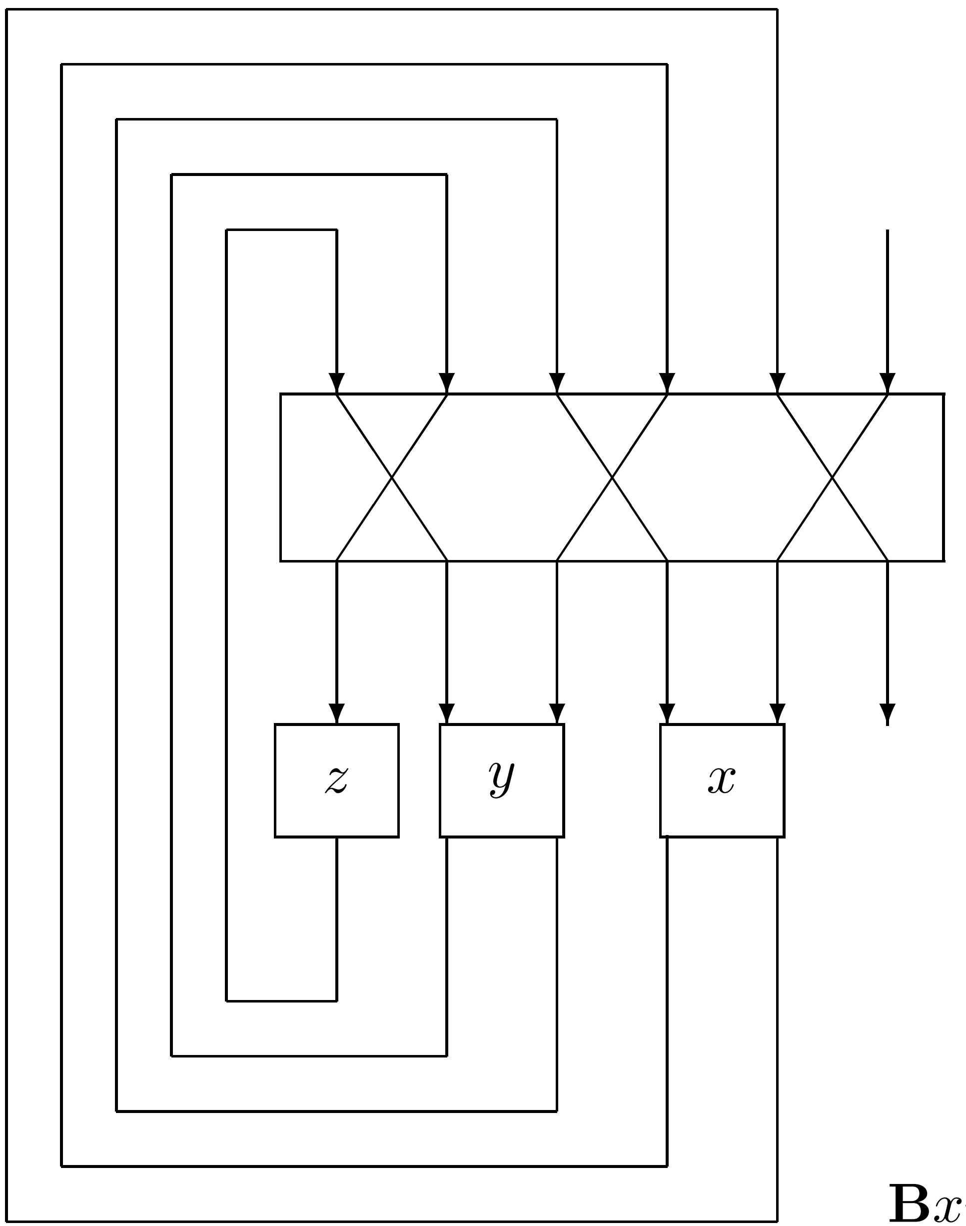


$$\mathbf{I}x = x$$

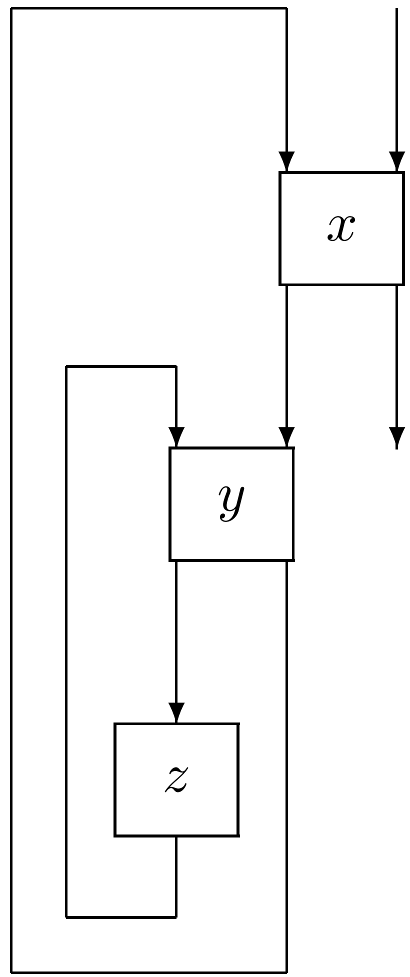
The Composition combinator

B





$\mathbf{B}xyz = x(yz)$



Further Developments

- The full story of the structure we have described is that it is a **Linear Combinatory Algebra**, which gives rise to a model of Linear Logic.
- Moreover, a standard combinatory algebra can be built from it, hence it is computationally universal.
- All of this holds even if we restrict to extremely simply functions as elements of the algebra - **partial involutions** - i.e. partial functions satisfying

$$f(x) = y \iff f(y) = x.$$

- As this shows, the computations in this algebra are **reversible** in a very strong sense. This already indicates the connection to physical computation (since reversible computations are exactly those which can be performed with perfect thermodynamic efficiency).
- There is a further link to **Quantum computation**. There is a 'wave style' as well as a particle style Geometry of Interaction, and an interpretation in Hilbert spaces. Using these ideas, the paper 'Physical Traces' by Abramsky and Coecke begins to develop a high-level, logic-based approach to quantum computation.
- There is a general axiomatic version in terms of **traced monoidal categories**, with instances for deterministic, non-deterministic, probabilistic and quantum interaction.

Papers available at:

`http://web.comlab.ox.ac.uk/oucl/work/samson.abramsky/`

For introductory accounts, see in particular:

- 'Semantics of Interaction'
- 'Algorithmic Game Semantics: a tutorial introduction'
- 'Algorithmic Game semantics and Component-Based Verification'
- 'A Structural Approach to Reversible Computation'