Games, Logic and Interaction

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

Themes

- Towards Logical Dynamics
- The power of copying
- Logic – Computation – Physics

Computation in the Age of the Internet

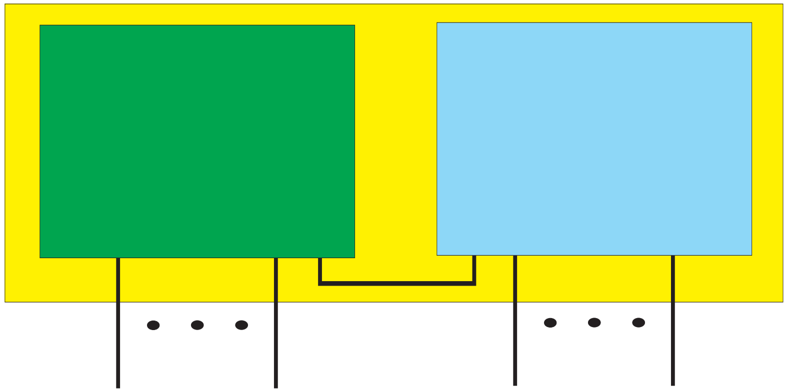
Instead of isolated systems, with rudimentary interactions with their environment, the standard unit of description or design becomes a *process* or *agent*, the essence of whose behaviour is *how it interacts* with its environment.

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



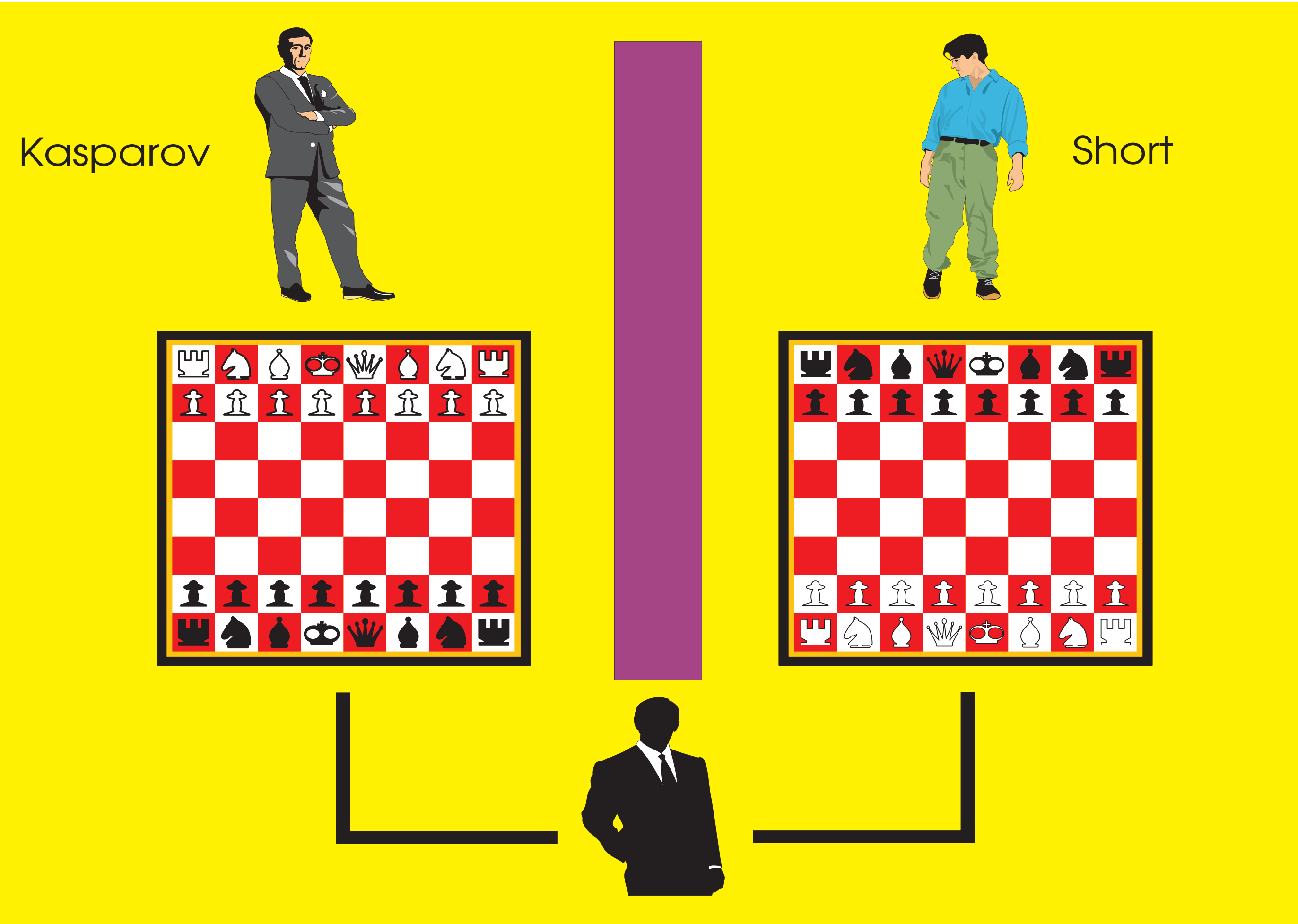
The static conception of logic

$$A \vee \neg A$$

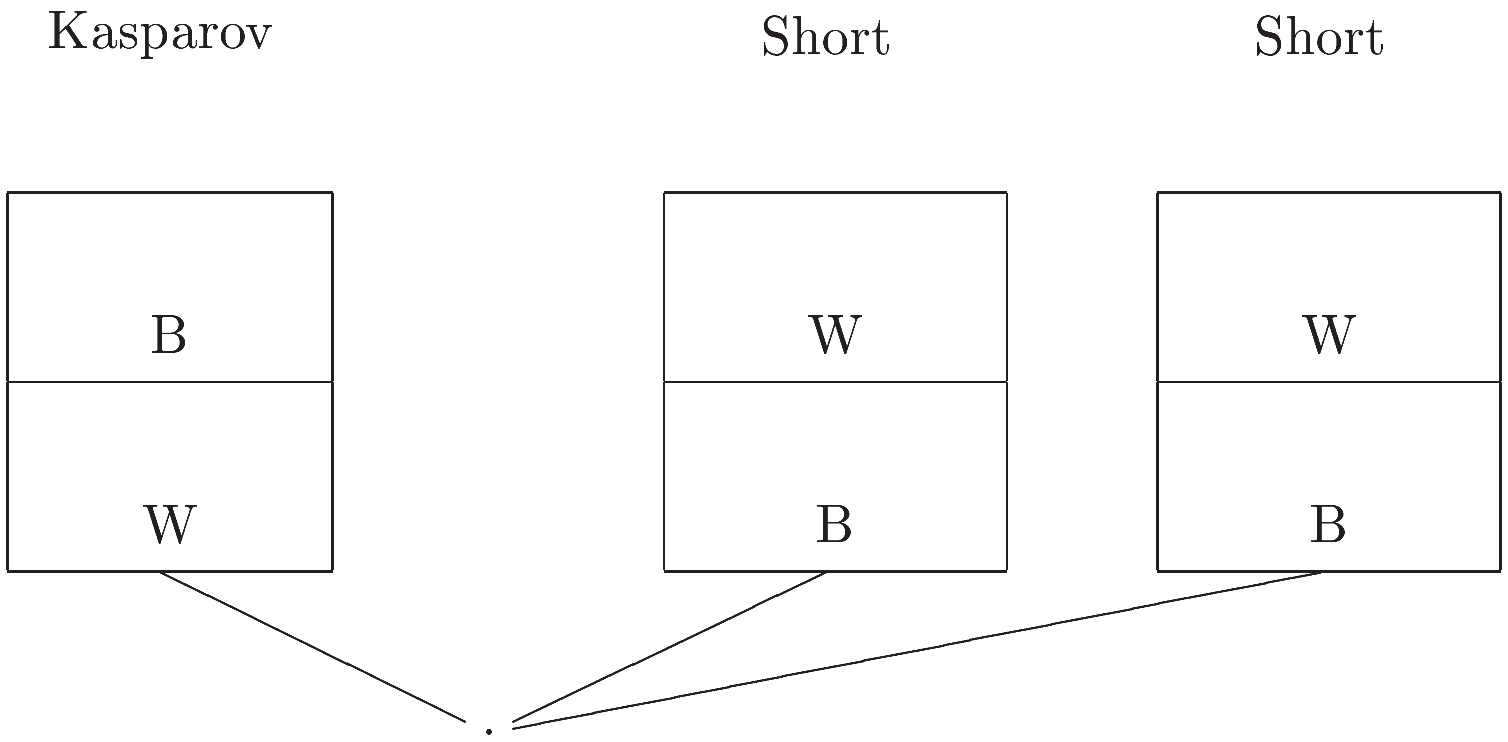
"It is raining *or* it is not raining"—truth-functional semantics.

(*Tertium non datur?*)

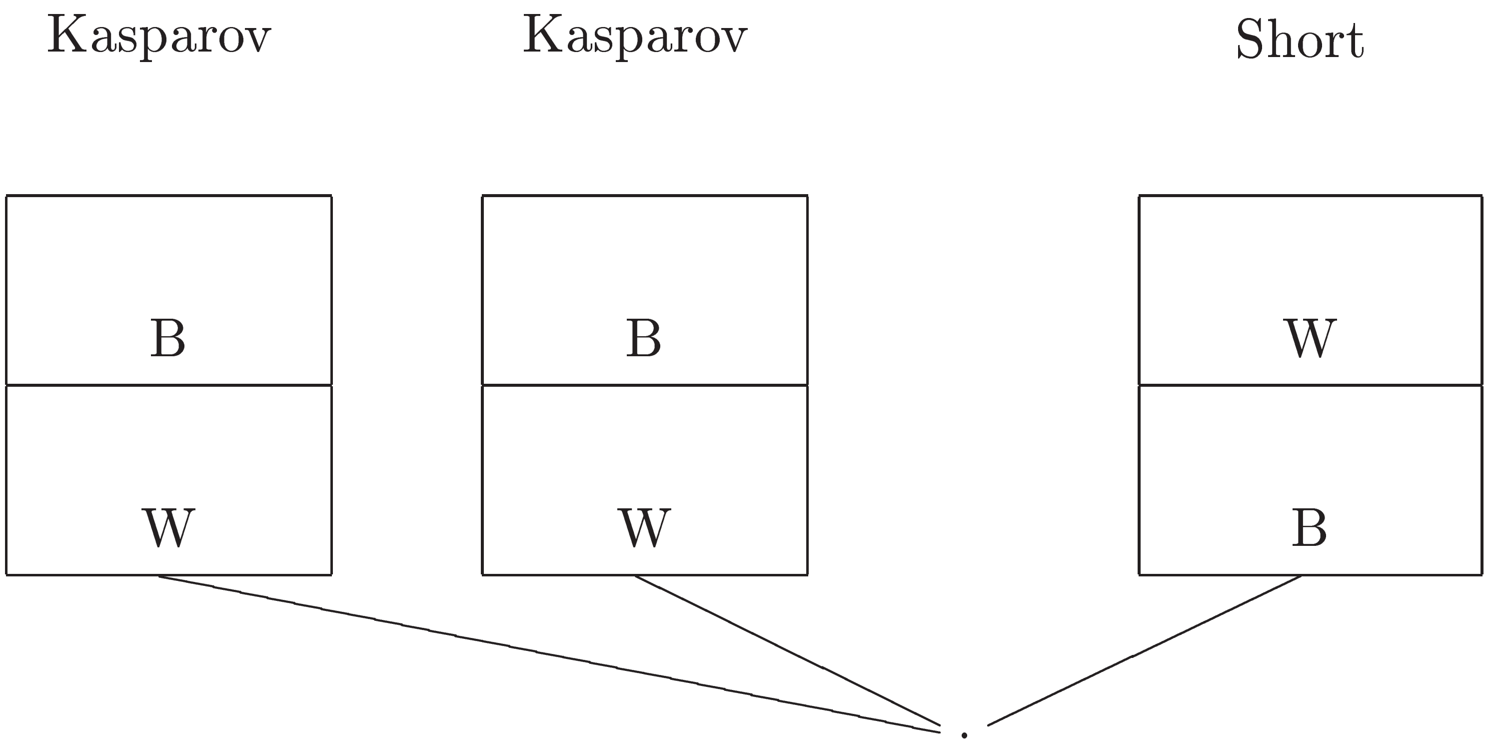
The Copy-Cat Strategy



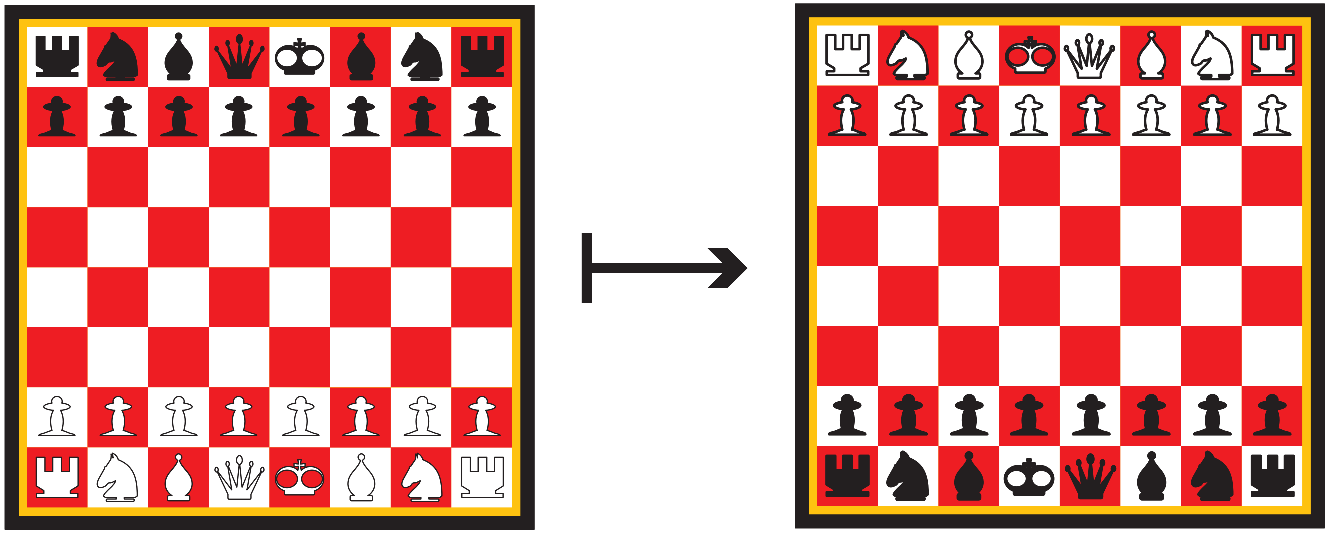
Does Copy-Cat still work here?



And here?

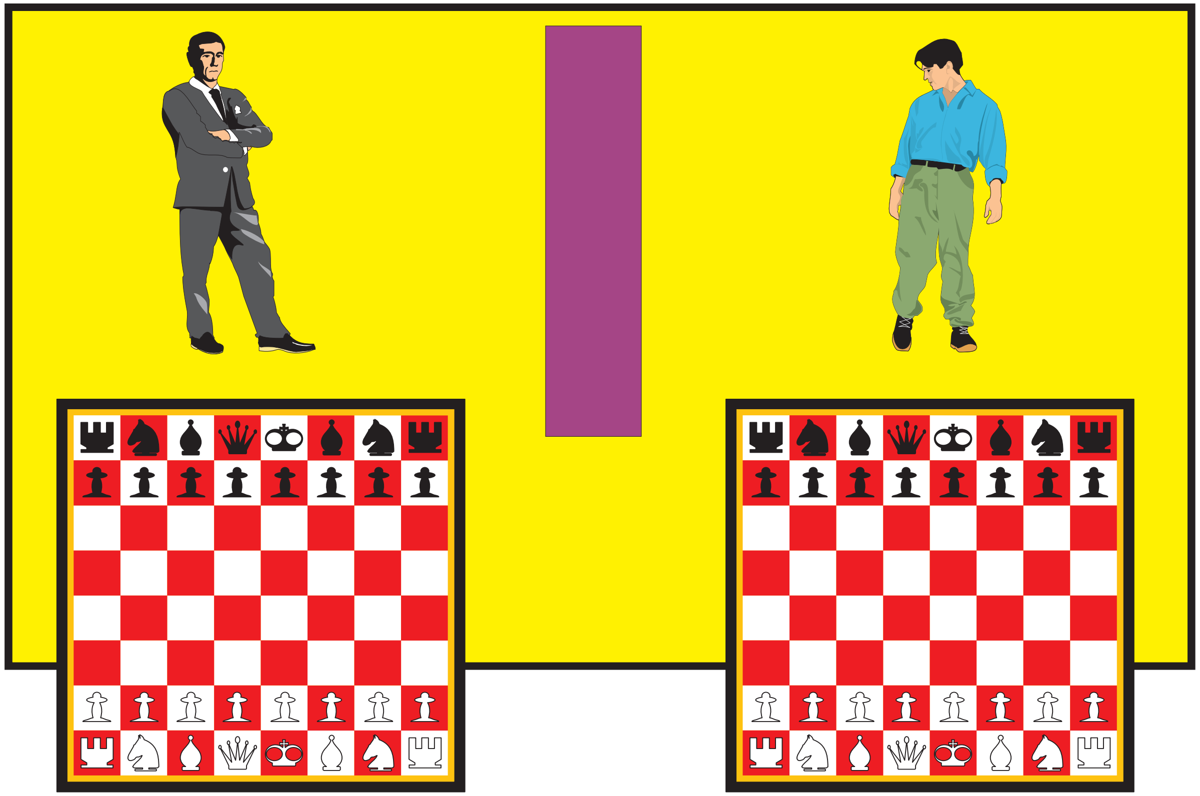


Duality—“Linear Negation”



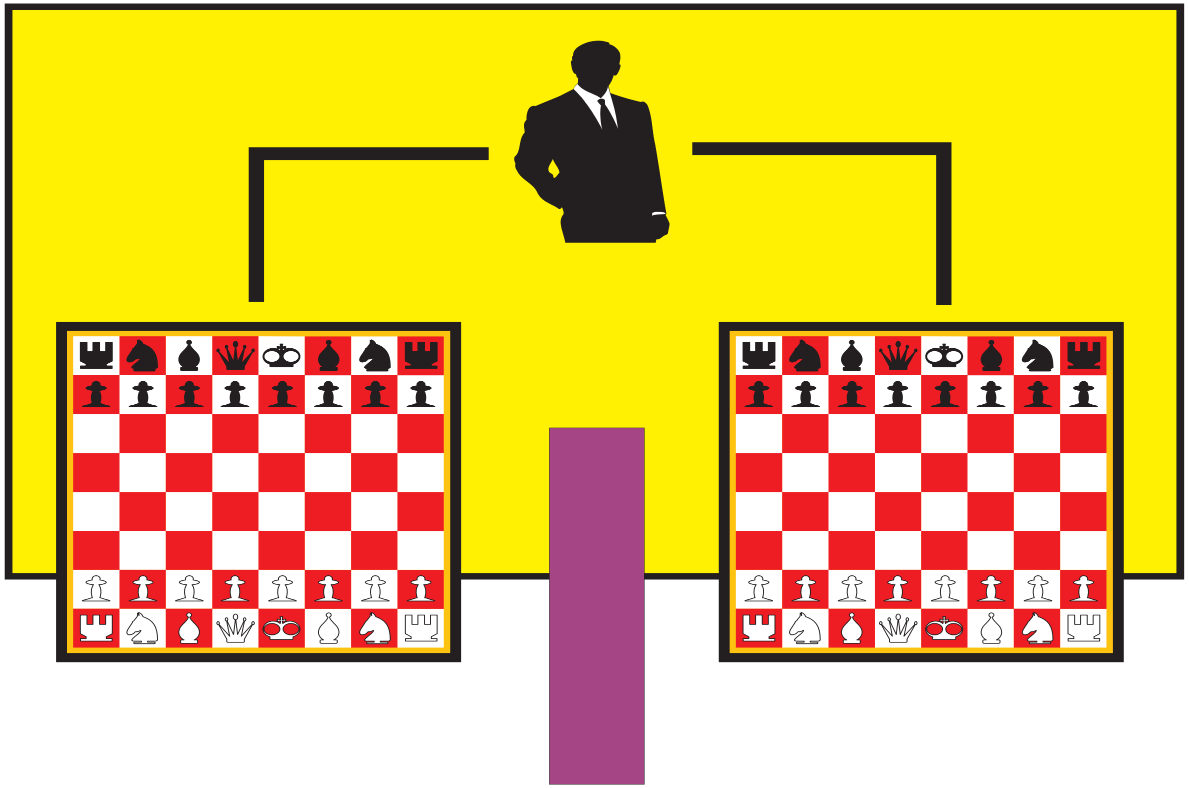
$$A^{\perp\perp} = A.$$

Tensor — "Linear conjunction"



$$A \otimes B$$

Par — "Linear disjunction"



$$A \parr B$$

Logical structure

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

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

$$(cf.\; A \supset B \;\equiv\; \neg A \vee B.)$$

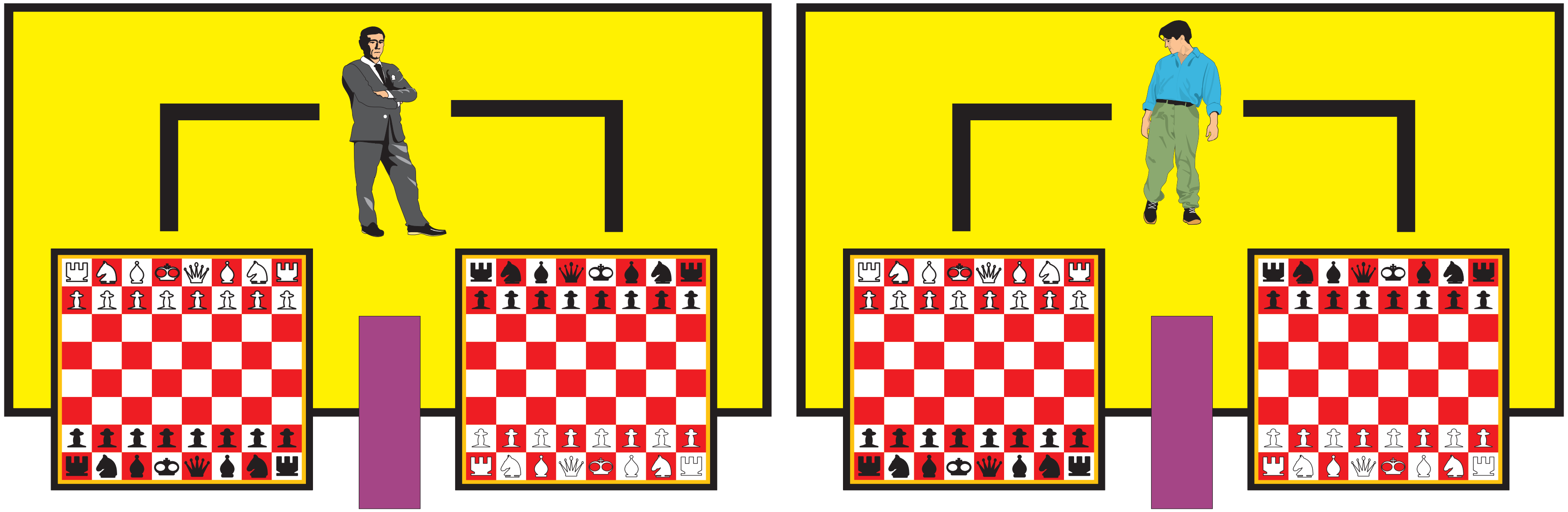
The Copy-cat strategy is a winning strategy for $A^{\perp} \parr A \;\equiv\; A \multimap A$.

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.

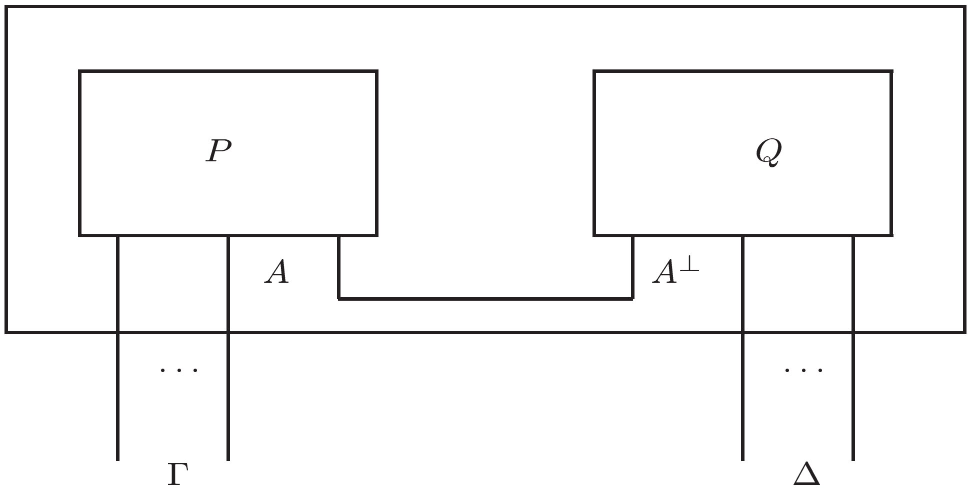


$A^{\perp} \parr B \;\equiv\; A \multimap B$ $\qquad$ $B^{\perp} \parr C \;\equiv\; B \multimap C$

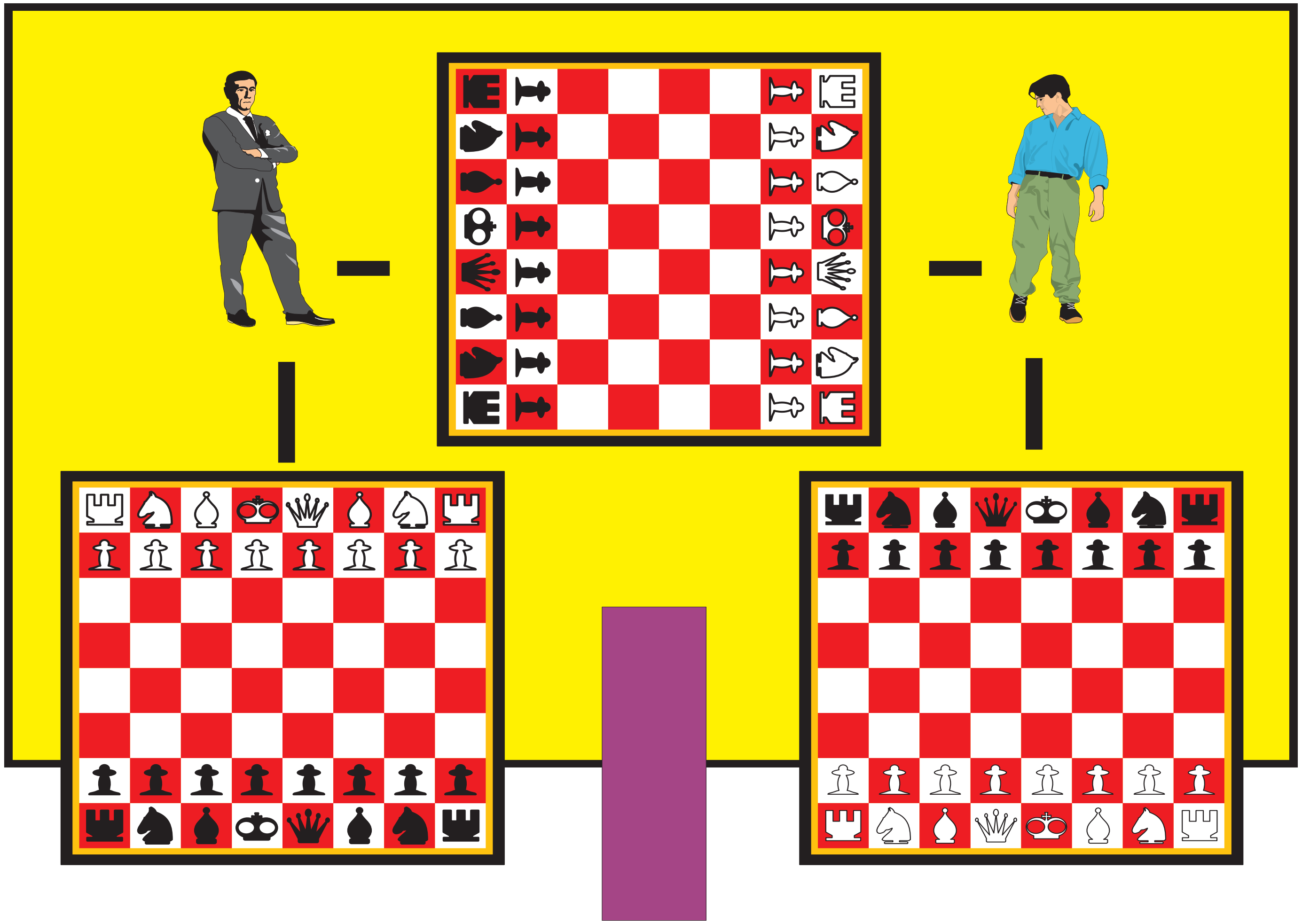
Composition

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

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



Composition as Interaction



Towards a Formalization

‘Particle-style Geometry of Interaction’

- Simple formalization, underlying geometry, ‘particle dynamics’.
- *Concrete models* of *higher-order phenomena.*
- Embodies Logic–Computation–Physics connection.

Combinatory Logic

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

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

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

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

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

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

The Curry Combinators: B, C, K, W

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

Principal types:

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

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

Sub-structural Logic

The **BCK** combinators support *affine bracket abstraction*: $[x]t$ where x *occurs at most once* in t.

We can define $\mathbf{I} \equiv \mathbf{CKK}$.

$$
\begin{array}{lcll}
[x]x & = & \mathbf{I} & \\
[x]t & = & \mathbf{K}t, & (x \notin FV(t)) \\
[x]tu & = & \mathbf{C}([x]t)u, & (x \in FV(t) \setminus FV(u)) \\
[x]tu & = & \mathbf{B}\,t\,([x]u), & (x \in FV(u) \setminus FV(t))
\end{array}
$$

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

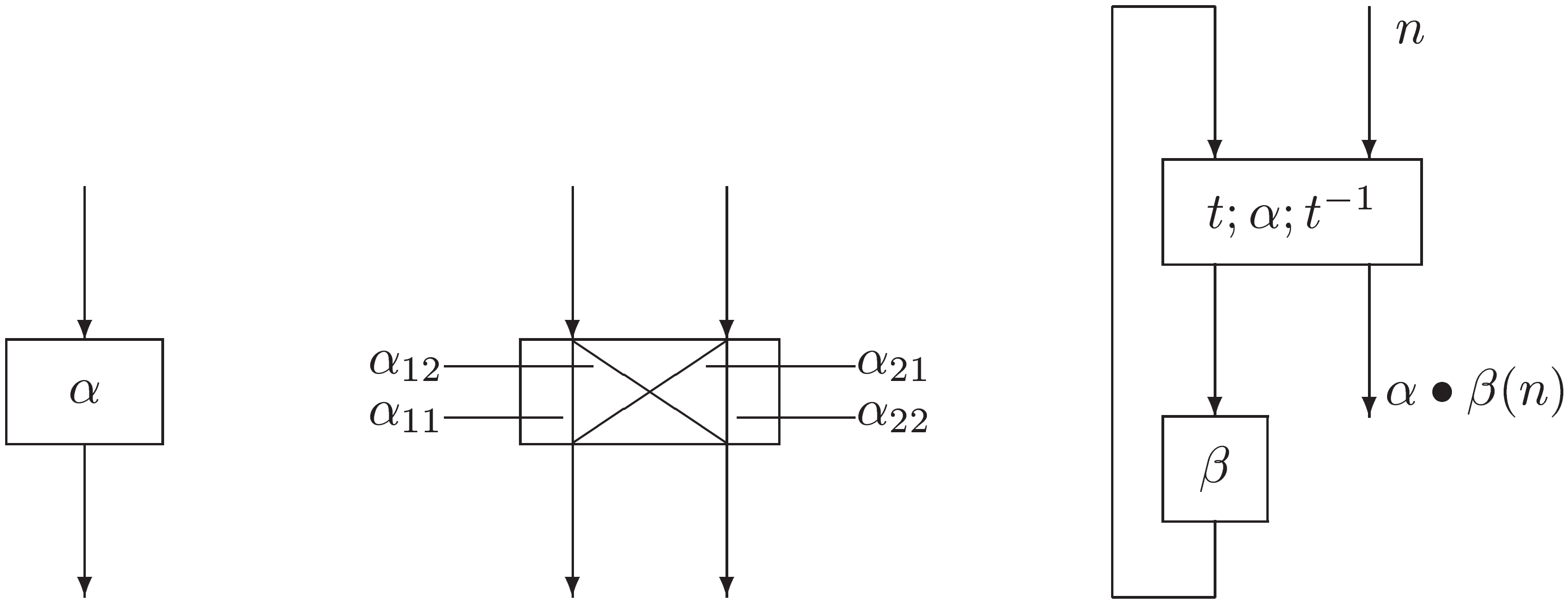
$$\mathbb{N} \rightharpoonup \mathbb{N}.$$

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

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

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:

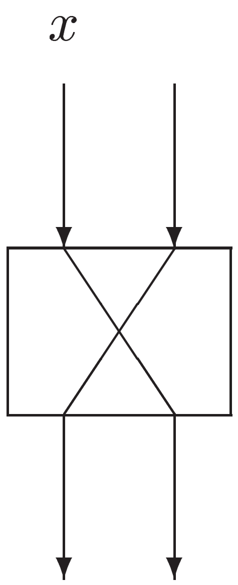
$$\alpha \bullet \beta = \alpha_{22} \cup \alpha_{21}; (\beta; \alpha_{11})^{\star}; \beta; \alpha_{12} \,,$$

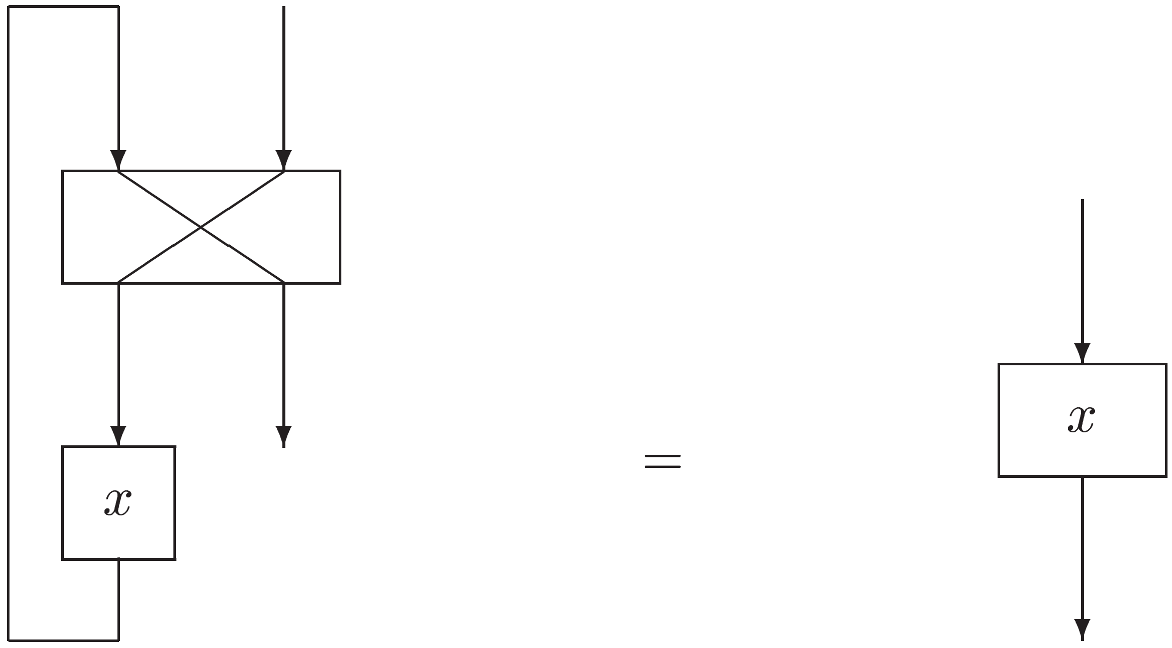
where

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

The Identity combinator

I

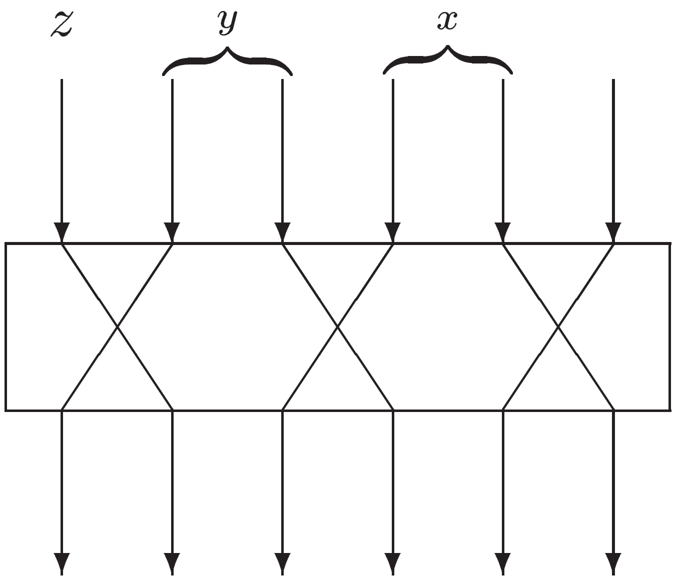


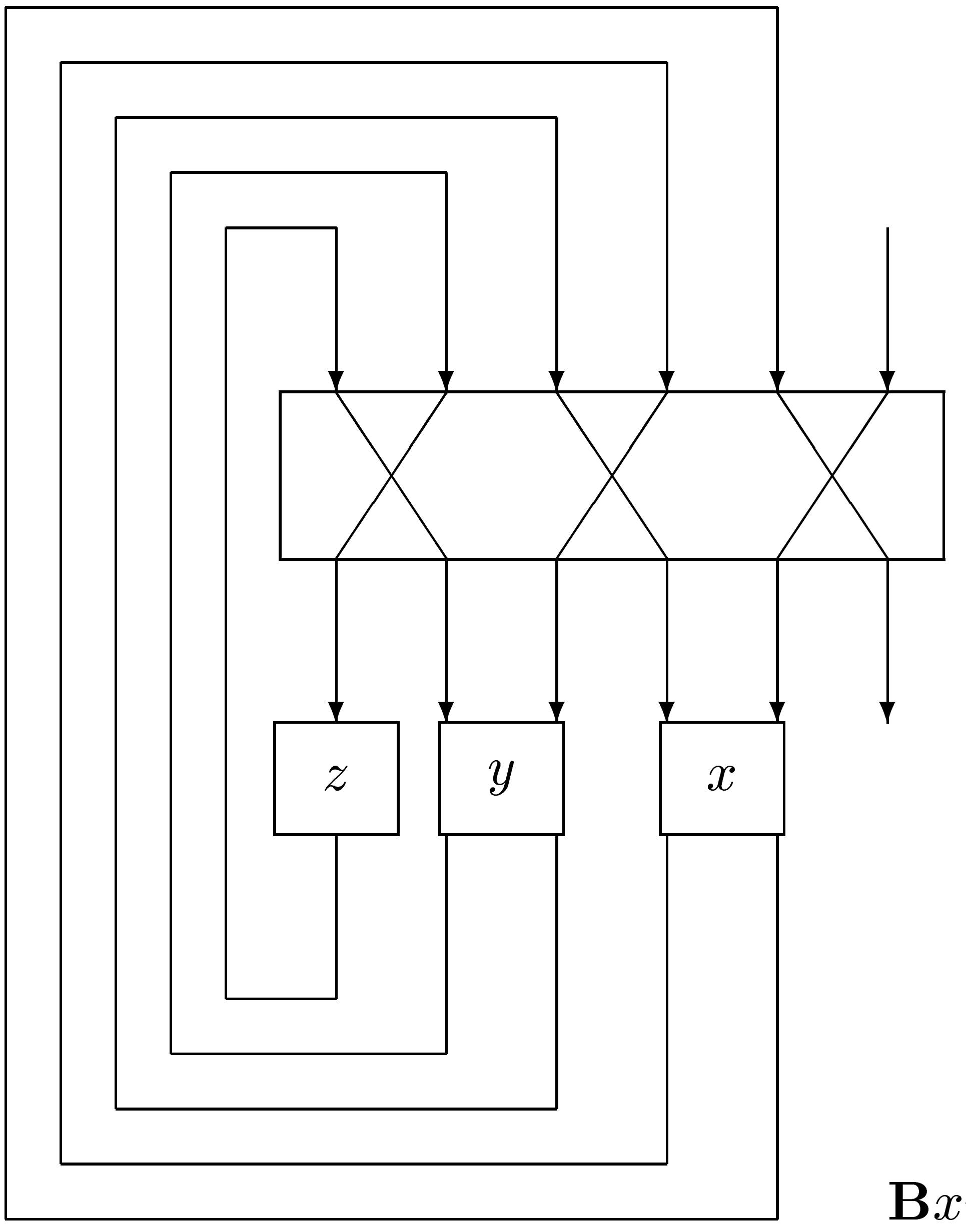


$$\mathbf{I}x = x$$

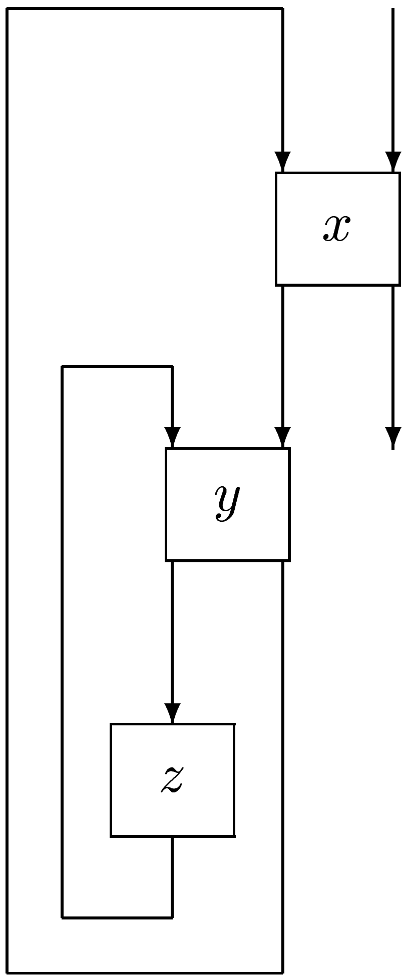
The Composition combinator

B

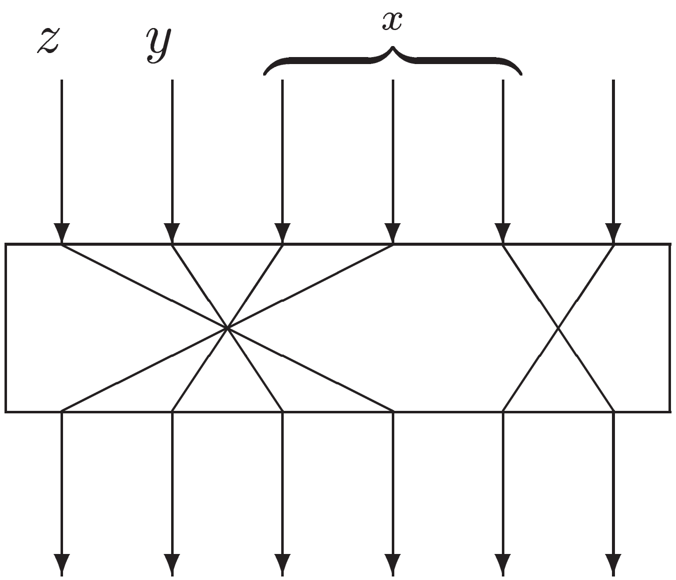




$$\mathbf{B}xyz = x(yz)$$

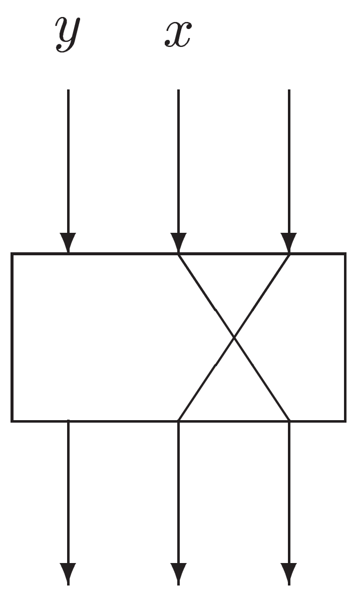


C



$$\mathbf{C}xyz = xzy.$$

K



$$\mathbf{K}xy = x.$$

Duplication

The algebra is *naturally affine*. Duplication of arguments must be catered for explicitly.

We define another coding function

$$p : \mathbf{N} \times \mathbf{N} \to \mathbf{N} \ .$$

and an operation $!\alpha$ which is intended to produce *infinitely many copies of* α. These are obtained by simply tagging each of these copies with a natural number, i.e. we define:

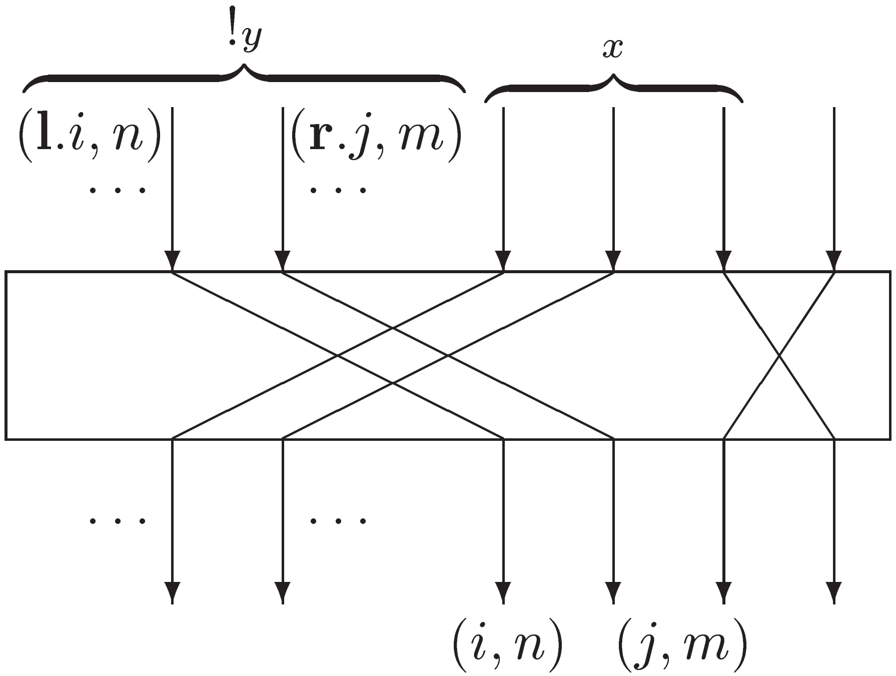
$$!\alpha = p^{-1}; (id_{\mathbf{N}} \times \alpha); p \ .$$

We can then define $\mathbf{W}'$ satisfying

$$\mathbf{W}'x\,!y = x\,!y\,!y.$$

The W′ combinator

W′



'Translation between dialects':

- x sees two arguments, each in many copies.
- $!y$ provides one argument, in as many copies as needed.

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.