10/24/06

Holodeck Strategies & the Lambda-Calculus

James Dolan

We're trying to understand the free CCC on one object X & the typed λ-calculus with one base type X, via games.

Objects in the free CCC on X correspond to derived types in the λ-calculus; morphisms correspond to "programs" or λ-terms.

We can understand both using games.

Objects correspond to games (described by "move trees"); morphisms correspond to strategies – but actually <u>holodeck</u> strategies.

For example, the object

$$X^{X^{X^{X}}}$$

corresponds to the type "functionalal", & the game with game tree

Note morphisms

$$1 \to B^A$$

correspond via "currying" to morphisms

$$A \to B$$

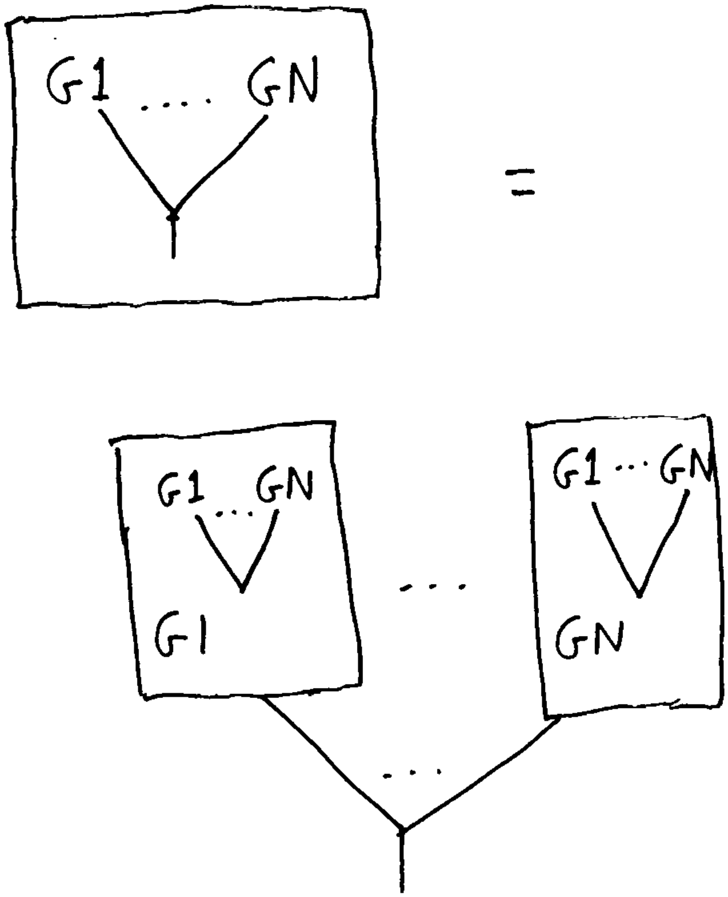
so we only need to understand "elements" or "points" $1 \to G$ to understand general morphisms. If, for example,

$$G = x^{x^{x^x}} \text{ “=” }$$

then

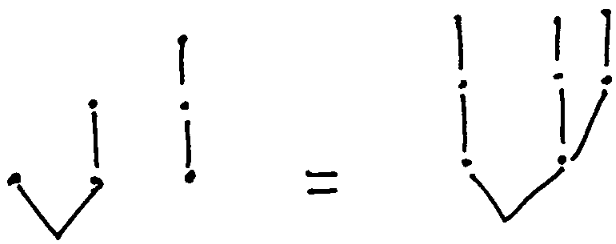
$$\text{Points}(G) = \text{Strat}(\boxed{G})$$

where $\boxed{G}$ is the holodeck version of G.

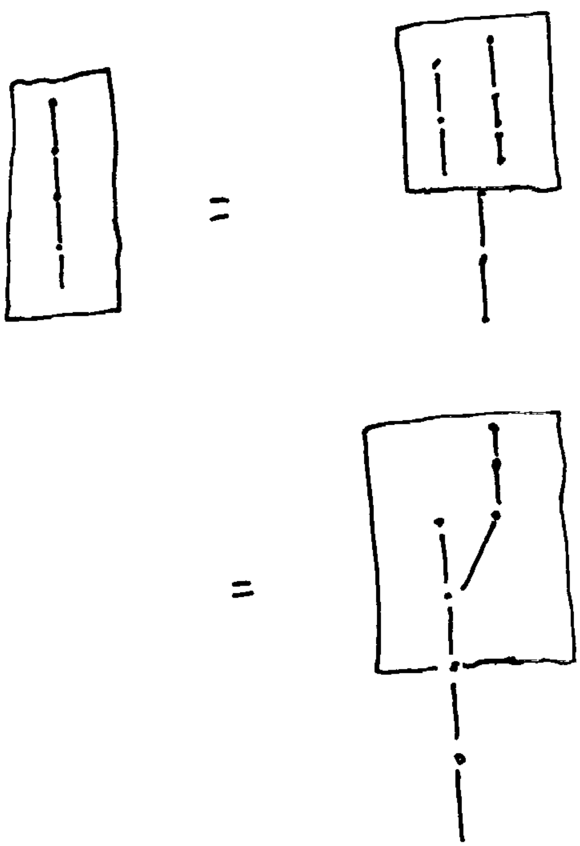


Here the exponential of games lets one take back moves: G^H is just like G except the 2nd player has the option, on their first move, of swithing to game H and "becoming the 1st player."

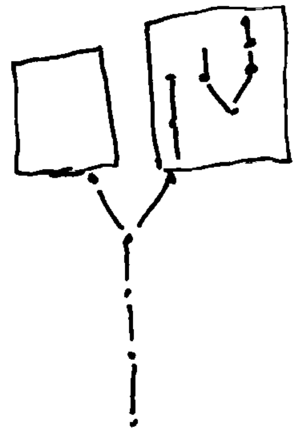
For example:

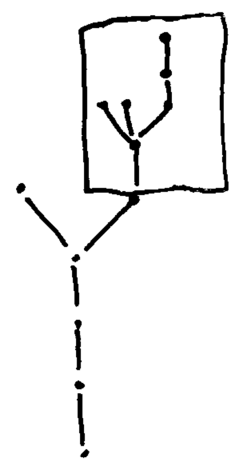


Let's do an example:

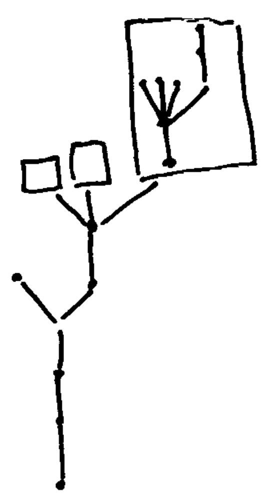


= ...

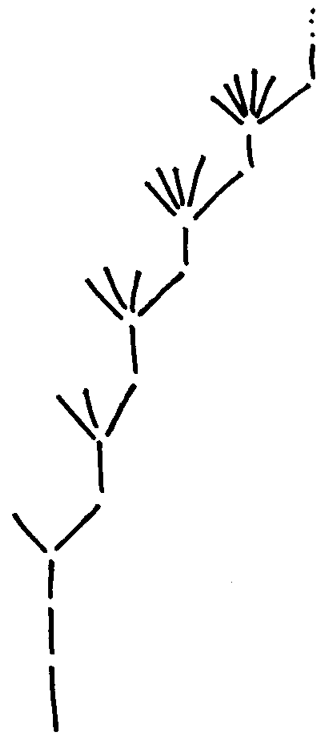
= 

= 

since we also decree that $\square$ is a homomorphism for multiplication

= 

= . . .

$=$ 

Note only the 2nd player gets to make choices in this game; he can win the game by choosing to head towards any *leaf*.

So, his winning strategies (in this example!) are just *leaves* of the tree $\boxed{G}$.

Lets show these correspond to λ-calculus expressions.

So we have:

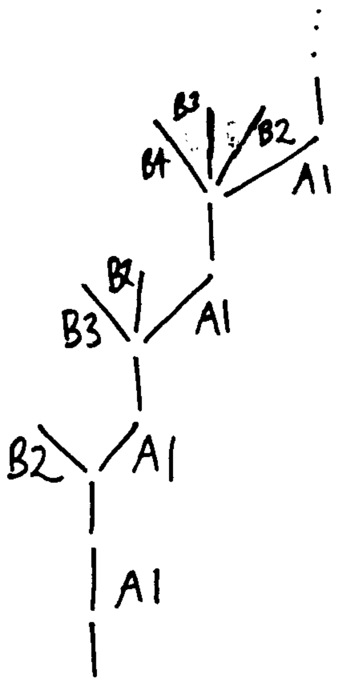
CCC's:	objects	morphisms
λ-calculus:	types	λ-terms
games:	move trees of games	2nd-player holodeck strategies

Composing strategies — say you have a (holodeck) strategy for checkers & your friend has a strategy for chess checkers. Then you can <u>compose</u> them & get a strategy for chess.

Name the 2nd player's moves :

& use these to name his moves in

$\boxed{G}$. We use this system :



In λ-calculus, A1 has type "functional", X^{X^X}, since that's where it shows up in G. A functionalal can be specified by a λ-calculus expression, like

$$A1 \mapsto A1(B2 \mapsto B2)$$

"evaluate the functional A1 on the identity function". This corresponds to the strategy for $\boxed{G}$ that corresponds to the leaf B2.

In general,

introductions of variables $\approx$ 2nd-player moves being offered for the first time

The A1 at the first level of the tree $\boxed{G}$ is being offered for the first time; the others aren't.

heads of expressions $\approx$ chosen 2nd-player moves

E.g. in $A1\,(B2 \mapsto B2)$, A1 is the head of an expression — a term being applied. In B2, B2 is a head of an expression, in a degenerate sort of way.

argument fields $\approx$ first-player moves

Here's a more interesting strategy, that leads to the leaf B3:

$$A1 \mapsto A1\,(B2 \mapsto A1(B3 \mapsto B3))$$

i.e. feed the functional A1 the <u>constant</u> function $B2 \mapsto A1\,(B3 \mapsto B3)$. Or:

$$A1 \mapsto A1\,(B2 \mapsto A1\,(B2 \mapsto B2))$$

This is a bit harder to describe in words!