COMBINATORICS OF PROOFS

LICS 2007

+

LOGIC COLLOQUIUM

2007

Martin Hyland

PLAN

Reflections on (extreme) abstraction

Game semantics

Combinatorics of innocent strategies

(Marginalia to LICS paper)

Work with Harmer + Melliès

CATEGORY THEORY

An example of Abstract Mathematics (also Model Theory, Graph Theory, ...)

NOT Abstract nonsense

NOT Just a language

MAYBE Useful level of generalization
(eg: Abstract Homotopy Theory)

SURELY Code of combinatorial complexity

TWO ASPECTS

Categories as worlds of structures

Categories as structures in themselves

CATEGORIES OF STRUCTURES

- Sets, Boolean-Valued Sets, Toposes
- Models of theories
- $\underbrace{\text{Topological spaces, schemes, étale toposes}}_{\text{categories of space}}$
- Domains

  Games

CATEGORY - STRUCTURES

(Groups, Posets)

Free structures

The free

- symmetric monoidal category on an object : $\mathbb{P}$ (permutations)
- symmetric monoidal category on a Frobenius object (commutative) topological category of Riemann surfaces
- free tortile category on an object the category of ribbon tangles

THE ABSTRACT IS CONCRETE

TQFT AND LOGIC

(an aside)

To give a 1–2 dimensional TQFT is to give a commutative Frobenius algebra.
(normal forms and 1-D algebraic topology)

But also interpreting proofs in propositional calculus in the same free category gives a notion of genus of proof
(modern version of idea of Kreisel)

GAME SEMANTICS

Developed in 1990s

- to model theories of proofs
- to give semantics of programming languages

(Berry-Curien Sequential Algorithms.)

Full Completeness for affine MLL (Abramsky-Jagadeesan)

Full Abstraction for PCF (AJM ; HON)

FRAGMENT OF HISTORY

(many other contributors but...)

Laird	LICS 1997
Abramsky Honda McCusker	LICS 1998
Hamer McCusker	LICS 1999
Hyland Schalk	LICS 2002
Abramsky Ghica Murawski Ong Stark	LICS 2004
Ong	LICS 2006
Tzevelekos	LICS 2007
Melliès-Tabareau	LICS 2007

Simple Games

Given by trees (with a start node) so

$1 = A(0) \leftarrow A(1) \leftarrow A(2) \leftarrow A(3) \ldots$

or just

$A(1) \leftarrow A(2) \leftarrow A(3) \ldots$

Opponent starts and after that Player/Proponent and Opponent alternate.

So moves in $A(2k+1)$ are for O

moves in $A(2k+2)$ are for P.

STRATEGIES

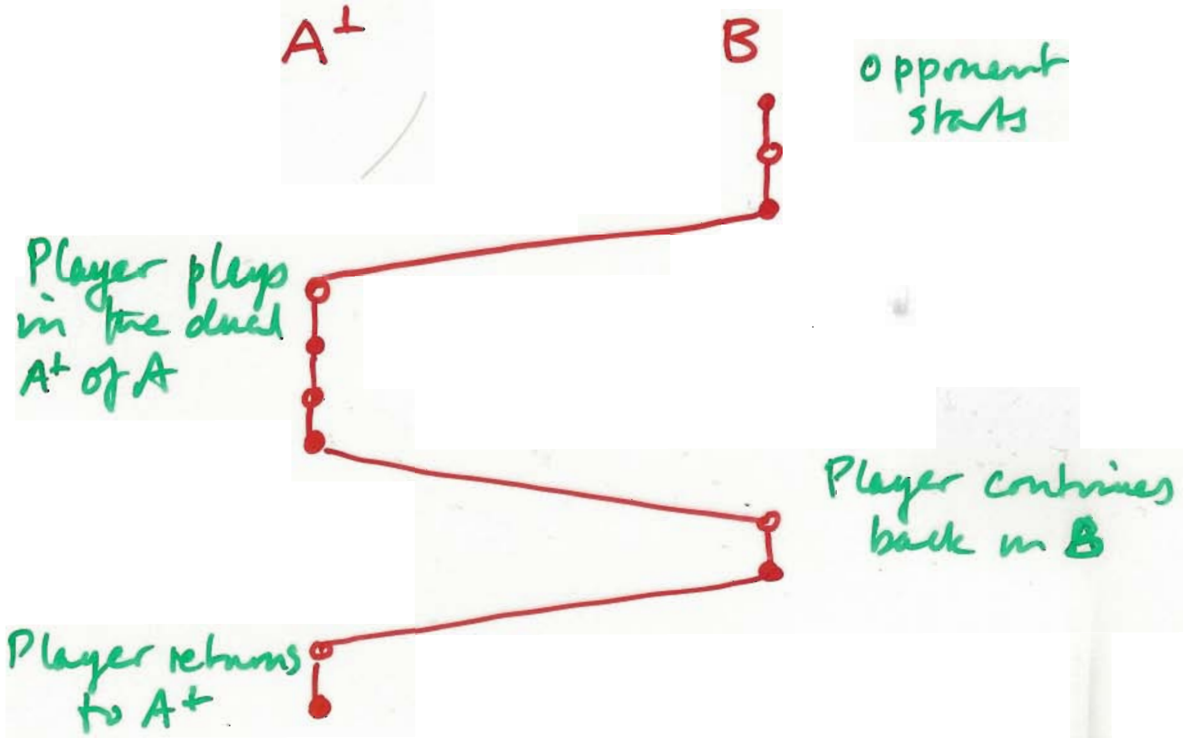
Obvious notion of strategy σ for P (Player, Proponent) in a game A.

Logic $\sigma : A$ deterministic and winning. σ proof of A.

Computer Science $\sigma : A$ partial, deterministic. σ a program of type A.

FUNCTION SPACE

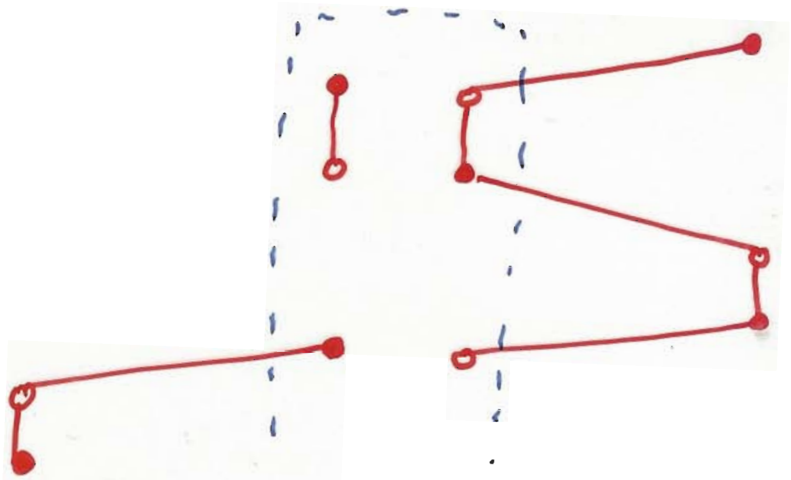
For games A, B we have a game $A \multimap B$ (Linear Logic) with plays



Player can decide to change games. Opponent plays where he can: so after opening move he follows Player.

Composition of Strategies

$$A \overset{\sigma}{\multimap} B \qquad B \overset{\tau}{\multimap} C$$



(Parallel composition plus hiding : Abramsky)

In Logic
$$\frac{A \vdash B \quad B \vdash C}{A \vdash C}$$

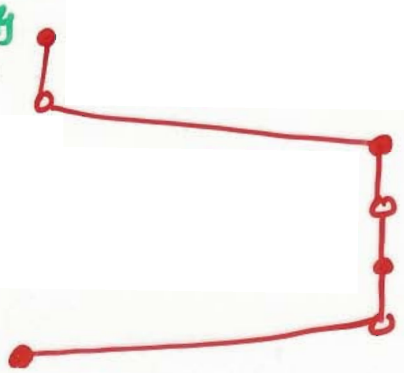
In Programming
$$\frac{b : B \quad g : B \to C}{g.b : C}$$

CATEGORICAL STRUCTURE

Tensor product

$A \otimes B$

Opponent starts in one or other



Oppenent changes

Oppenent changes again

(Player has to follow in whichever game opponent plays us.)

Categorical product

$A \times B$

Opponent starts in one of A, B and play is then restricted to that game. (No switching.)

FACT

Games and strategies form a category which is symmetric monoidal closed with products.

This is a model for multiplicative – additive intuitionistic Linear Logic.

Write $\mathbb{L}$ for this category.

NON-LINEAR GAMES

Allow repetition or backtracking.

Linear exponential comonad

$$(!, \varepsilon, \delta)$$

$$\varepsilon_A : !A \longrightarrow A$$

$$\delta_A : !A \longrightarrow !!A$$

natural in A

with coherent isomorphism

$$!A \otimes !B \cong !(A \times B)$$

Kleisli Category

$$\mathbb{K}(A, B) = \mathbb{L}(!A, B)$$

with composition

$$\frac{!A \xrightarrow{f} B \qquad !B \xrightarrow{g} C}{!A \xrightarrow[\delta_A]{} !!A \xrightarrow[!f]{} !B \xrightarrow[g]{} C}$$

is cartesian closed

$$\begin{aligned}
\mathbb{K}(A \times B, C) &\cong \mathbb{L}(!(A \times B), C) \\
&\cong \mathbb{L}(!A \otimes !B, C) \\
&\cong \mathbb{L}(!A, !B \multimap C) = \mathbb{K}(A, !B \multimap C)
\end{aligned}$$

Thus the function space is

$$B \Rightarrow C = !B \multimap C$$

(Girard translation)

TAKING STOCK

Simple game models give sound interpretations of

- proofs
- programs

but far from

fully complete
fully abstract

(Too much wild behaviour.)

Innocent strategies is one successful approach to control behaviour.

ARENA GAMES

An arena A is itself a (little) game. It generates a game tree of plays from itself with arbitrary backtracking.

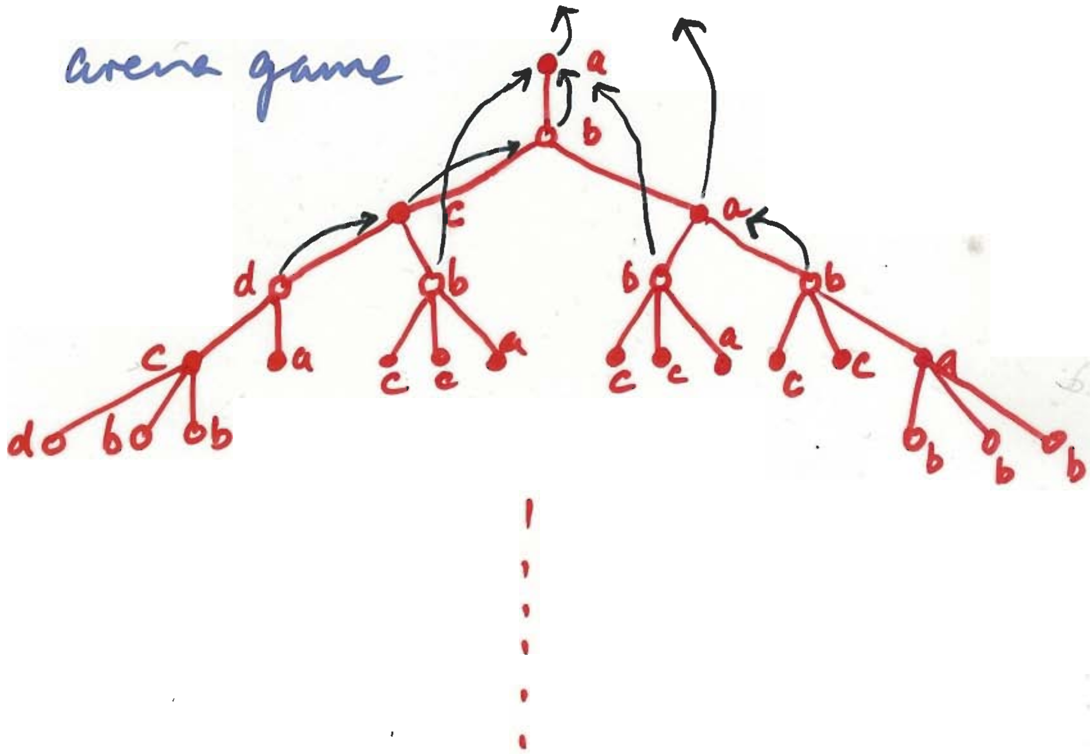
(This is the most general situation which will not be analyzed here.)

EXAMPLE: ARENA GAME

Arena

generates the

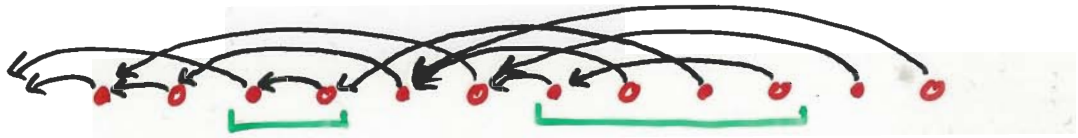
arena game



The backtracking is recorded by pointers

INNOCENT STRATEGIES

Given a play with backtracking the P-view is obtained by starting at the end and discarding moves between O-pointers as one works back

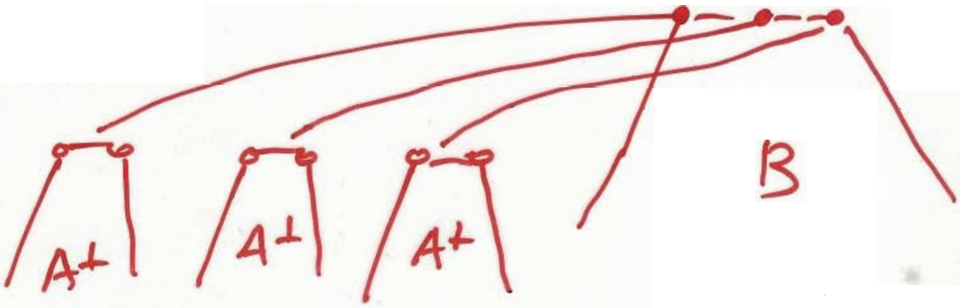


Now only P-pointers give information

An innocent strategy is one in which the next move for Player depends only on the view

Function Space Arena

$A \mathbin{-\!\!\boxplus} B$



As a game, start in B;
Player then either continues in B
or switches to $A^{\perp}$
(irrevocable choice)

Additive
construction

COMPOSITION ISSUE

Innocent strategies in the arena games for

$A \mathbin{-\!\boxplus} B$ and $B \mathbin{-\!\boxplus} C$

give

innocent strategies in the arena game for

$A \mathbin{-\!\boxplus} C$

FACT

Innocent strategies are the basis for a range of

full completeness
full abstraction

results.

BUT why do such strategies compose. It is not trivial and looks like a hack.

OBSERVATION

Innocent strategies in the arena game for A are just strategies in $?A$.

But $?(A \mathbin{-\!\!\boxplus} B)$

$\cong \; !A \multimap ?B$

How to compose

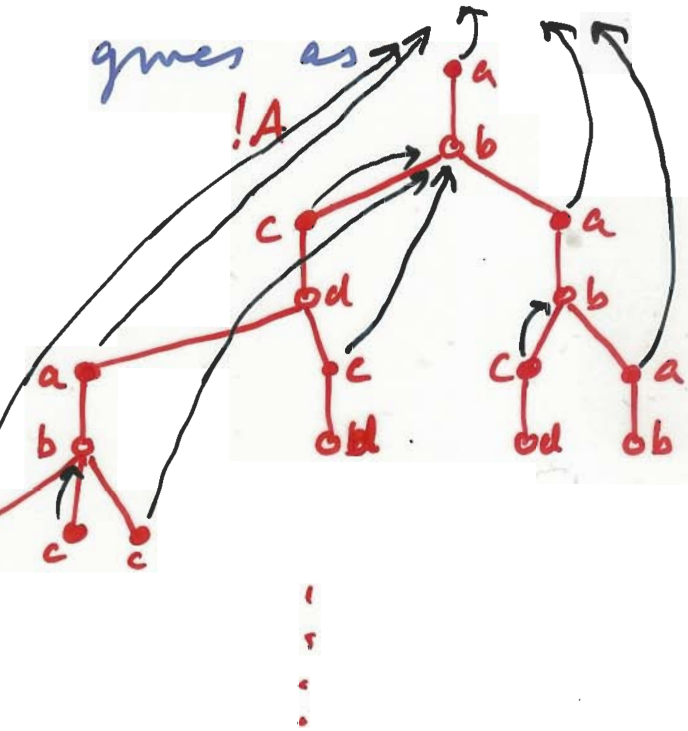
$!A \overset{\sigma}{\multimap} ?B$ and $!B \overset{\tau}{\multimap} ?C$?

(Type mismatch)

EXPONENTIAL COMONAD

!A is the game in which only Opponent can backtrack.

Arena A  gives as !A

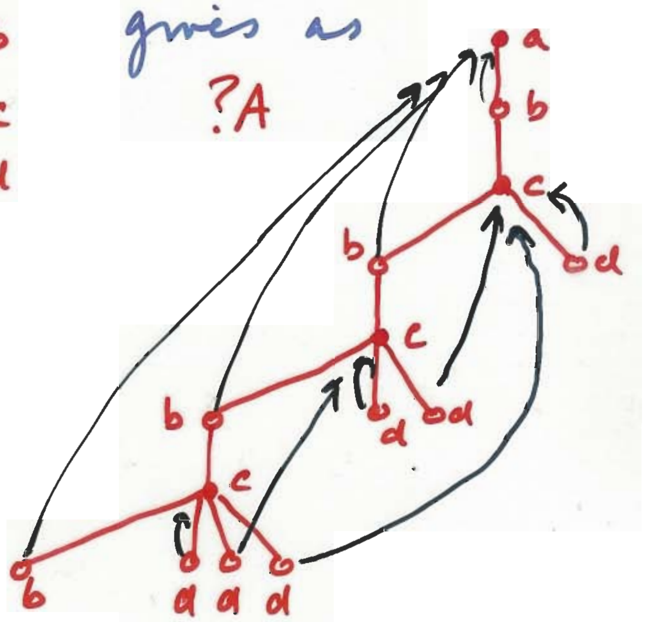


EXPONENTIAL MONAD

?A is the game in which only Player can backtrack

Arena A

gives as ?A



(Strategies for $\lambda F.\ F(\lambda x.x)$

$\lambda F.\ F(\lambda x.\ F(\lambda y.x))$

$\lambda F.\ F(\lambda x.\ F(\lambda y.y))$ etc

in $((A \Rightarrow A) \Rightarrow A) \Rightarrow A$)

DISTRIBUTIVE LAW

(comonad - monad case)

We have a natural transformation

$$\lambda_A : \,!?A \longrightarrow ?!A$$

with good properties.

Then from

$$\frac{!A \xrightarrow{f} ?B \qquad !B \xrightarrow{g} ?C}{}$$

get $!A \xrightarrow[\delta_A]{} !!A \xrightarrow[!f]{} !?B \xrightarrow[\lambda_B]{} ?!B \xrightarrow[?g]{} ??C \xrightarrow[\mu_C]{} ?C$

RESULT

A cartesian closed category of games and innocent strategies (plus other structure).

BUT where do the

comonad $(!, \varepsilon, \delta)$

monad $(?, \eta, \mu)$

distributive law. λ

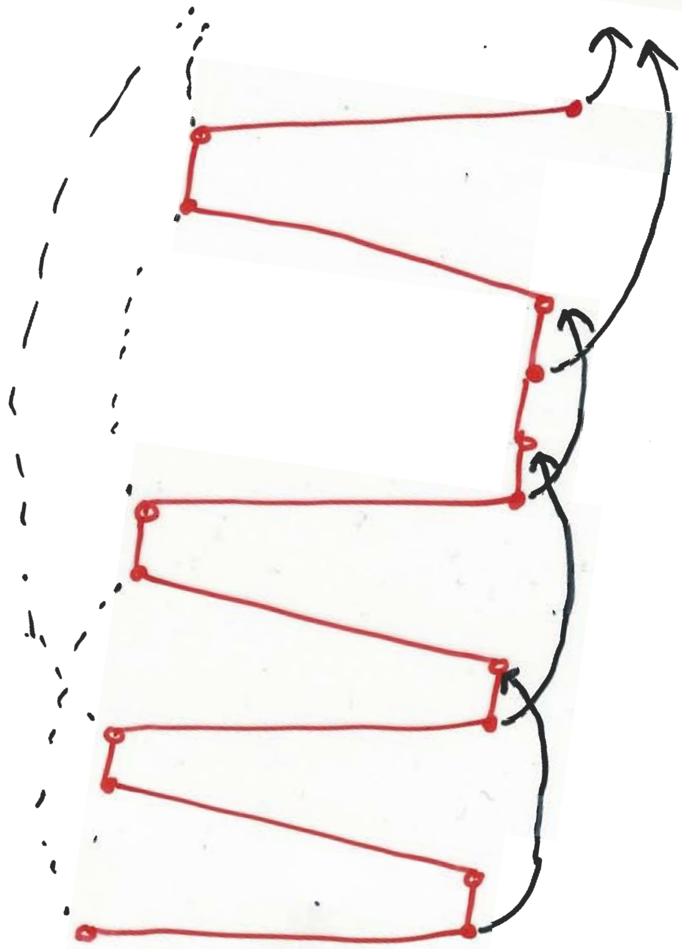
come from?

ANSWER From the combinatorics of pointers

What's the mathematics of that?

BASIC TOOL

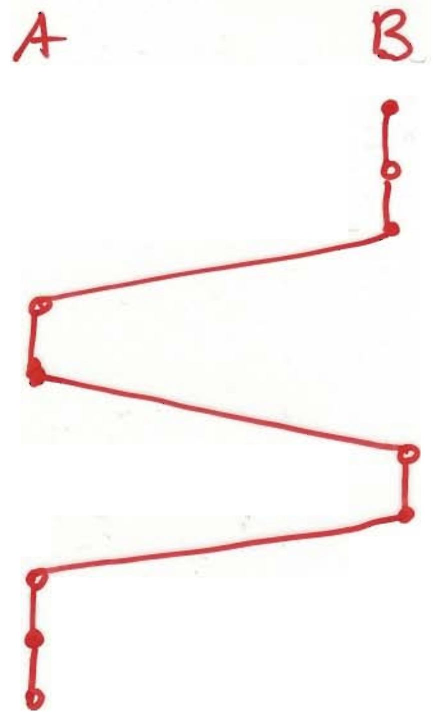
Pulling back O-pointers



Dually pushing forward P-pointers

SCHEDULES

Patterns of play in $A \to B$



What is this combinatorial information?

(And how does it induce composition via parallel composition and hiding.)

PARITY CONDITION

If the length of the play in A is odd, then so is that in B

That is we can have

A	B
even	even
even	odd
odd	odd

BUT NOT

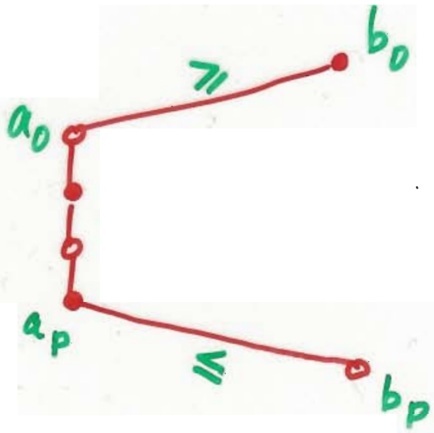
odd	even

MERGES

(H-S) Schedules can be given as merges by the information

$$a_0 \geqslant b_0$$

$$b_p \geqslant a_p$$



Composition of strategies comes from composition of relations.

THE CATEGORY Δ_+

(Drop the + and) write Δ for the natural (to category theorists) category of

- finite ordinals and
- order preserving maps

Δ is the free monoidal category generated by a monoid

$[\Delta^{op}, Set]$ is the category of (augmented) simplicial sets

$\Delta^{\top}$ category of non-0 ordinals and top preserving

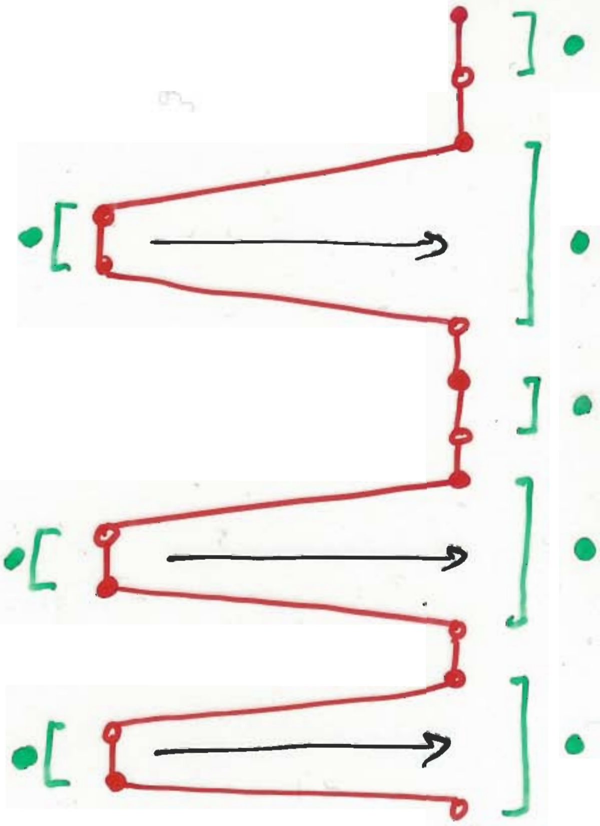
$\Delta_{\perp}$ ———— and bottom preserving

$\Delta^{\top}_{\perp}$ ———— and endpoint preserving.

3

SCHEDULES AND Δ

Case 1 even even

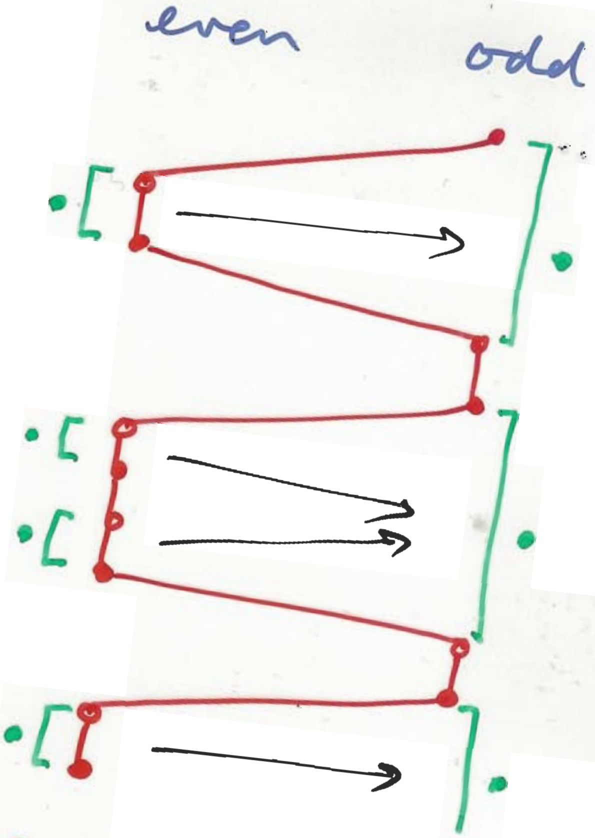


So $2m$ to $2n$

is $\Delta(m,n)$

SCHEDULES AND Δ

Case 2

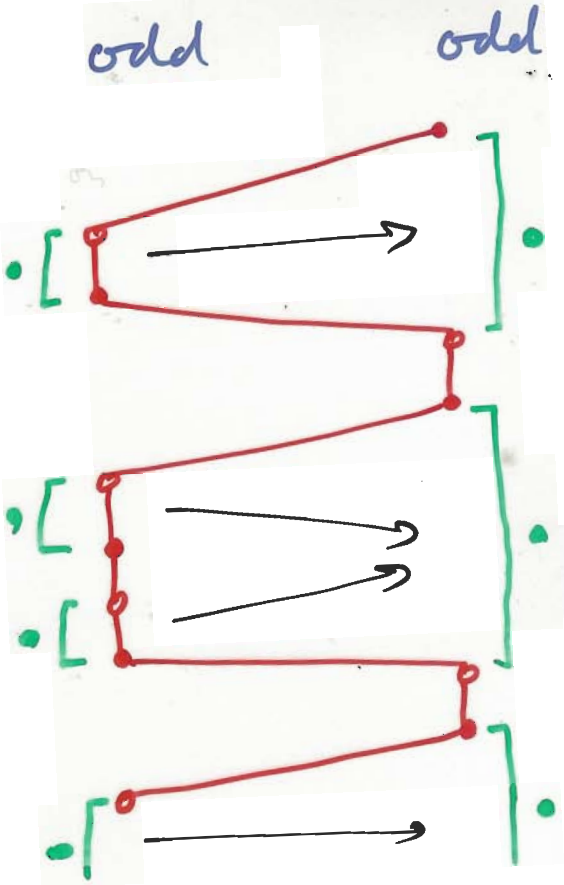


So is $2m$ to $2n+1$

$\Delta(m, n+1) \cong \Delta^{T}(m+1, n+1)$

SCHEDULES AND Δ

Case 3



So $2m+1$ to $2n+1$ is $\sim \Delta^T(m+1, n+1)$

CATEGORY Υ

of schedules

Objects $\quad 2m \quad m$

$\qquad\qquad 2n+1 \quad n^+$

Maps
$$\begin{aligned}
\Upsilon(m,n) &= \Delta(m,n)\\
\Upsilon(m,n^+) &= \Delta(m,n+1)\\
&= \Delta^T(m+1,n+1)\\
\Upsilon(m^+,n^+) &= \Delta^T(m+1,n+1)\\
\Upsilon(m^+,n) &= \emptyset
\end{aligned}$$

Composition 'obvious': this is the lax colimit of the profunctor $F^*:\Delta^T \nrightarrow \Delta$ from the free $F:\Delta\to\Delta^T$

(Another explanation of composition in games.)

PULLING BACK O-POINTERS

Evidently Υ embeds in Posets

O-pointers on a sequence of length

$2n$	are	$H \longrightarrow n$
$2n+1$		$H \longrightarrow n+1$

identity on objects maps in Posets. (Heaps.) (H a forest)

Take the pullback in Posets.

DUALITY ON Δ

$2 \in \Delta$ and $\Delta(m, 2) = m+1$
in the pointwise order.

'Hence' $\Delta(m, n) \cong \Delta_\perp^T(n+1, m+1)$

Also $\Delta^T(m, 2) = m$ in the
pointwise order $(m \geq 1)$.

'Hence' $\Delta^T(m, n) \cong \Delta^T(n, m)$

Dual view of Υ

$$\Upsilon(m, n) = \Delta_\perp^T(n+1, m+1)$$

$$\Upsilon(m, n^+) = \Delta^T(n+1, m+1)$$

$$\Upsilon(m^+, n^+) = \Delta^T(n+1, m+1)$$

PUSHING FORWARD P-POINTERS

The dual view of $\mathbb{V}$ gives an embedding $\mathbb{V}^{op}$ in Posets.

P-pointers on a sequence of length

$$\begin{array}{lcl} 2n & & H \longrightarrow n+1 \\ & \text{are} & \\ 2n+1 & & H \longrightarrow n+1 \end{array}$$

identity on objects maps in Posets. (Heaps) (H a forest)

Again take a pullback in Posets

CONCLUSIONS

Abstract methods come into the treatment of game semantics for proofs and programs at two levels:

to construct categories of structures

to handle inherent combinatorial structure.

Some think innocent strategies too close to proofs / programs. The reconstruction demonstrates real content in the full completeness / abstraction results

I hope this style of approach will extend.