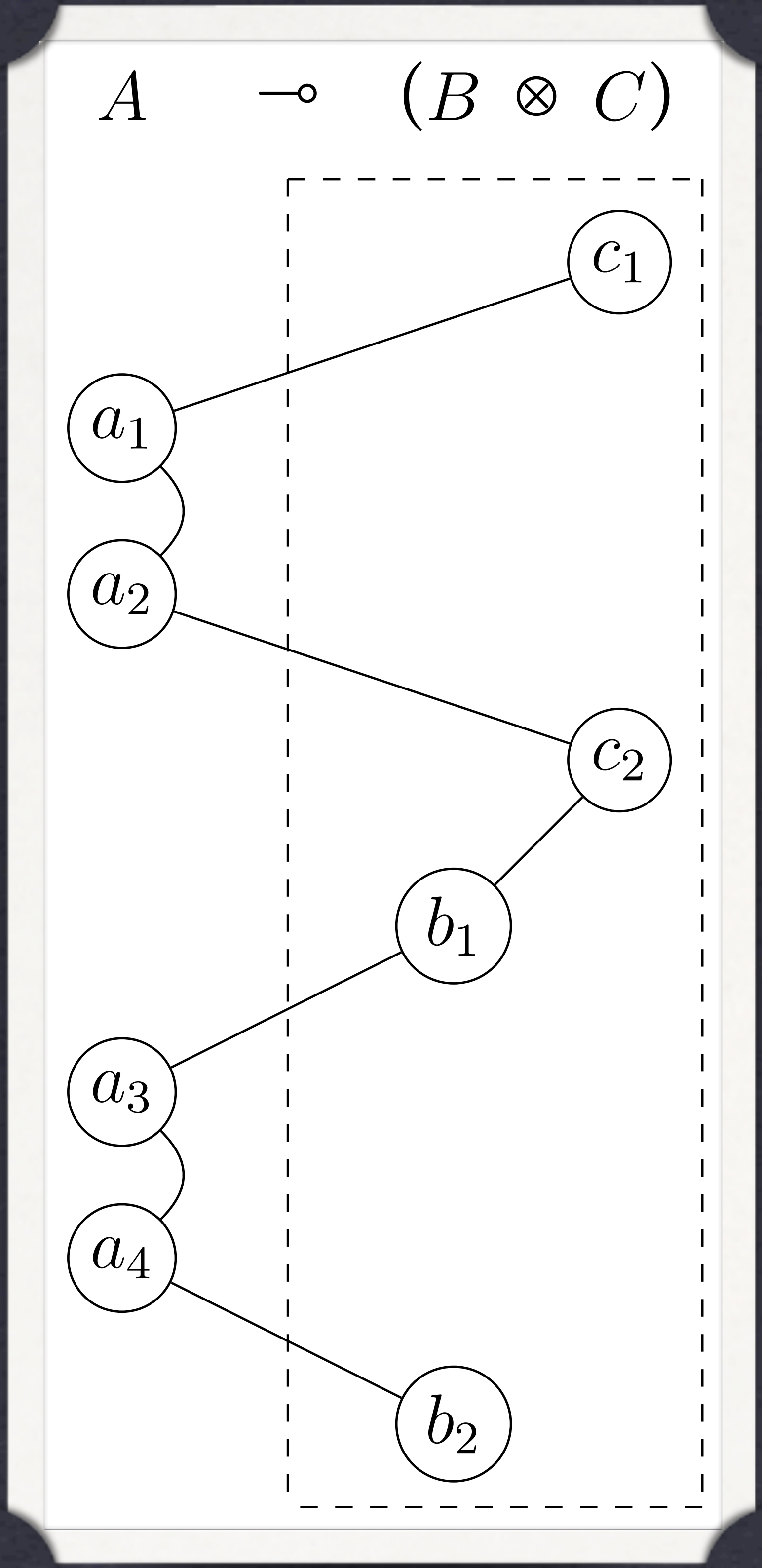
Graphical Foundations for Dialogue Games

Guy McCusker, John Power, Cai Wingfield
Mathematical Foundations of Computer Science, University of Bath

June 7th, 2013
Laboratoire d'Informatique de Paris Nord



This work is concerned with some of the mathematical devices that give structures to games.

In particular:

- Schedules for interleaving in $\multimap$ and $\otimes$
- Heaps for backtracking in !
- (Just schedules for today)

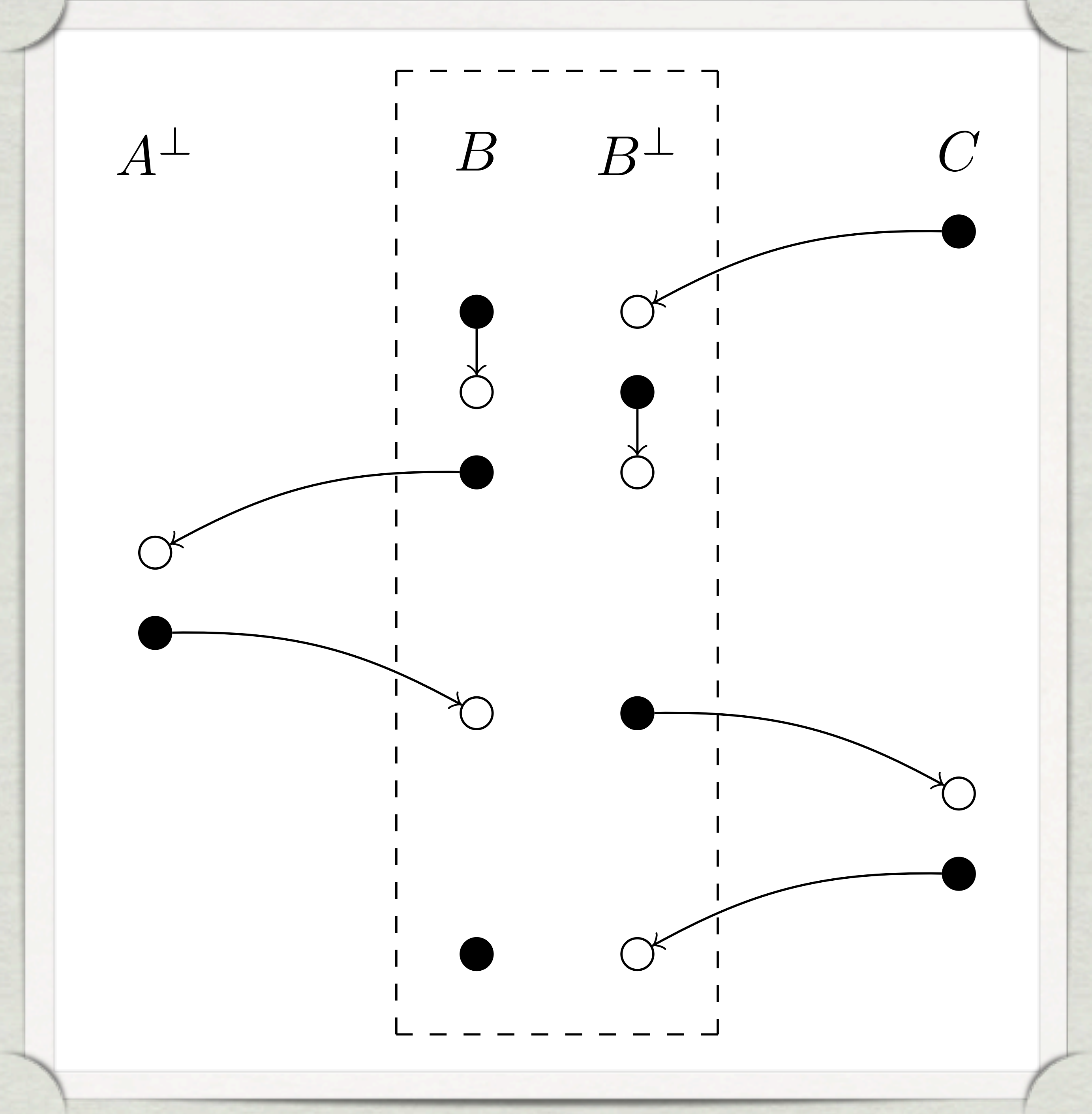
Schedules

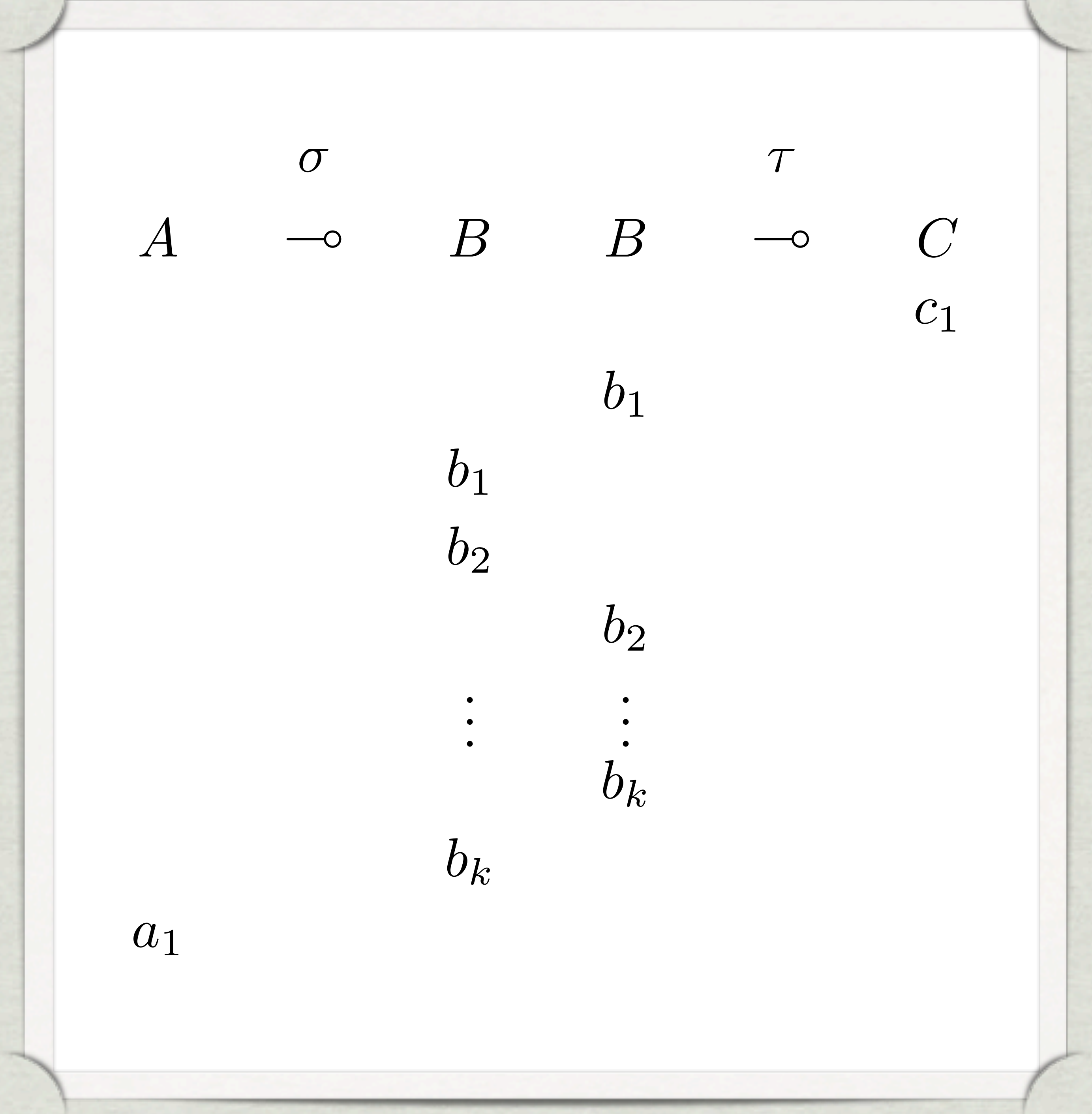
Schedules as an explicit interleaving structure were introduced by Harmer, Hyland and Melliès in 2007.

Their definitions are very combinatorial.

In practice, when describing a position of a game, or interactions between strategies, people tend to draw pictures.

Harmer, Hyland and Melliès. *Categorical combinatorics for innocent strategies*. LICS 2007.





“COMPOSITION OF STRATEGIES”

$(\mathbb{N}$	$\Rightarrow$	$\mathbb{N}$	$\Rightarrow$	$\mathbb{N})$	$\Rightarrow$	$\mathbb{N}$	
						q	O
				q			P
q							O
1							P
		q					O
		1					P
				n			O
						n	P

"$\lambda f.f\,1\,1$"

Aim of this work

These pictures allow intuitive descriptions and intuitive arguments to be made.

We'd like to let games *be given by* these diagrams.

We want to:

- Characterise the diagrams.
- Describe graphical arguments and methods.

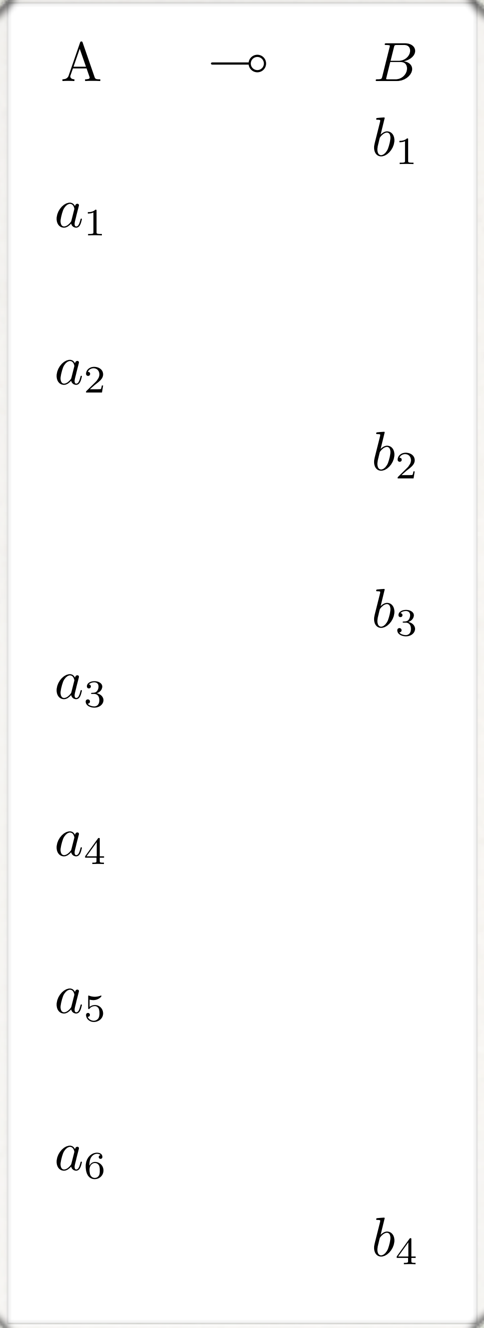
Then intuitive arguments become proofs in terms of the definitions.

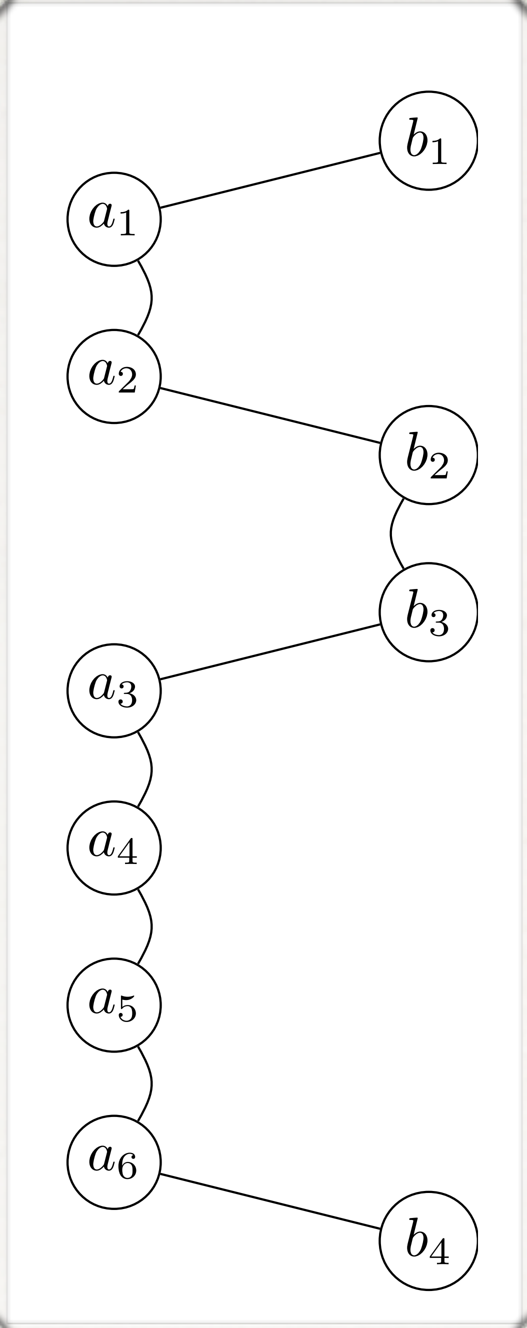
Outline

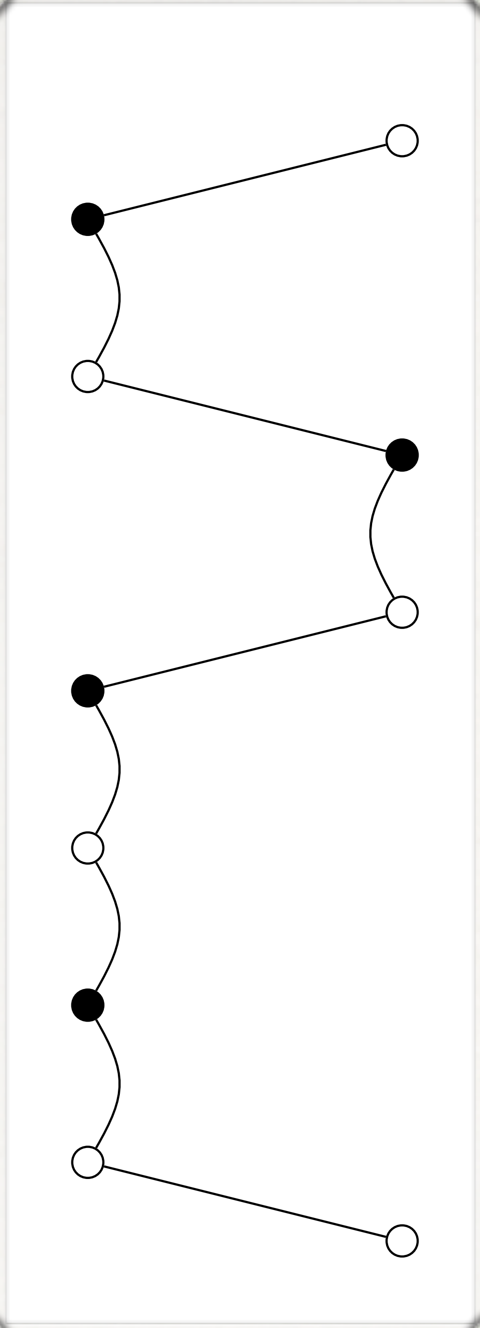
What I'll do here:

- Describe a general framework for plane graphs
- Characterise schedules diagrams for $\multimap$ and $\otimes$
- Show how schedules can describe plays in games
- Characterise interleaving graphs
- Show how interleaving graphs can describe games
- Describe how interleaving graphs relate to schedules
- Show how this permits arguments for categorical properties
- Mention other/future directions

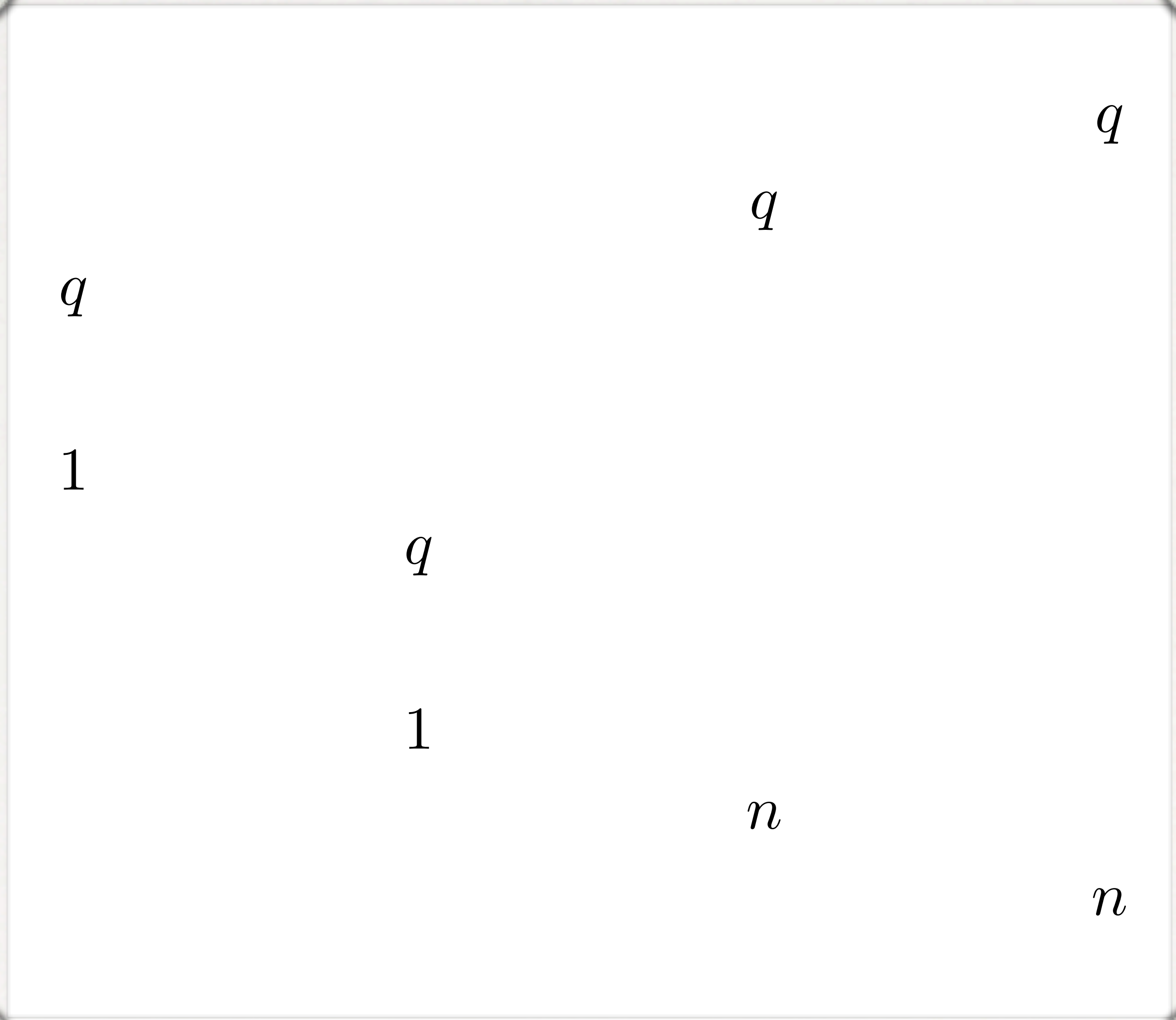
What pictures to characterise?

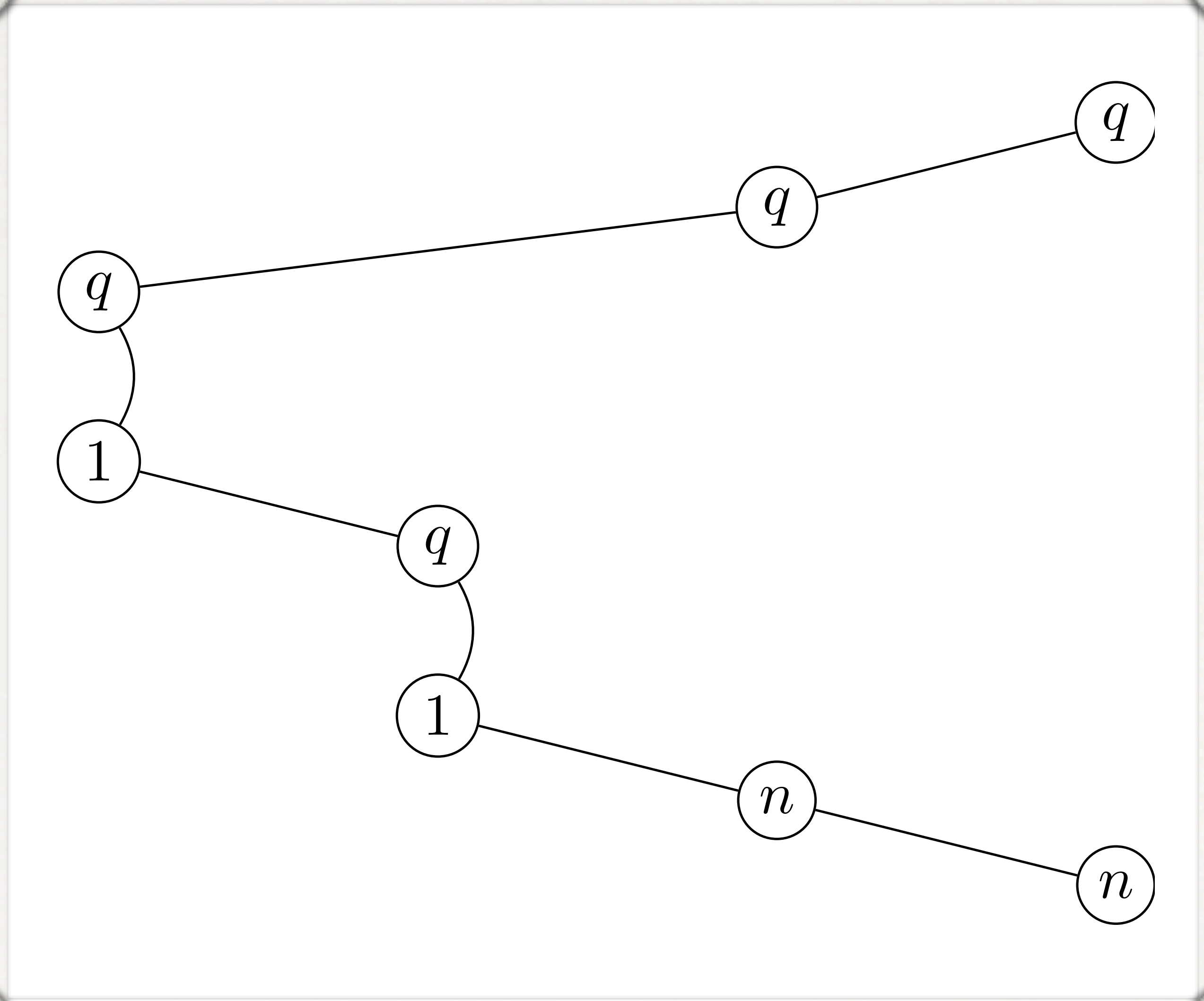






What pictures to characterise?





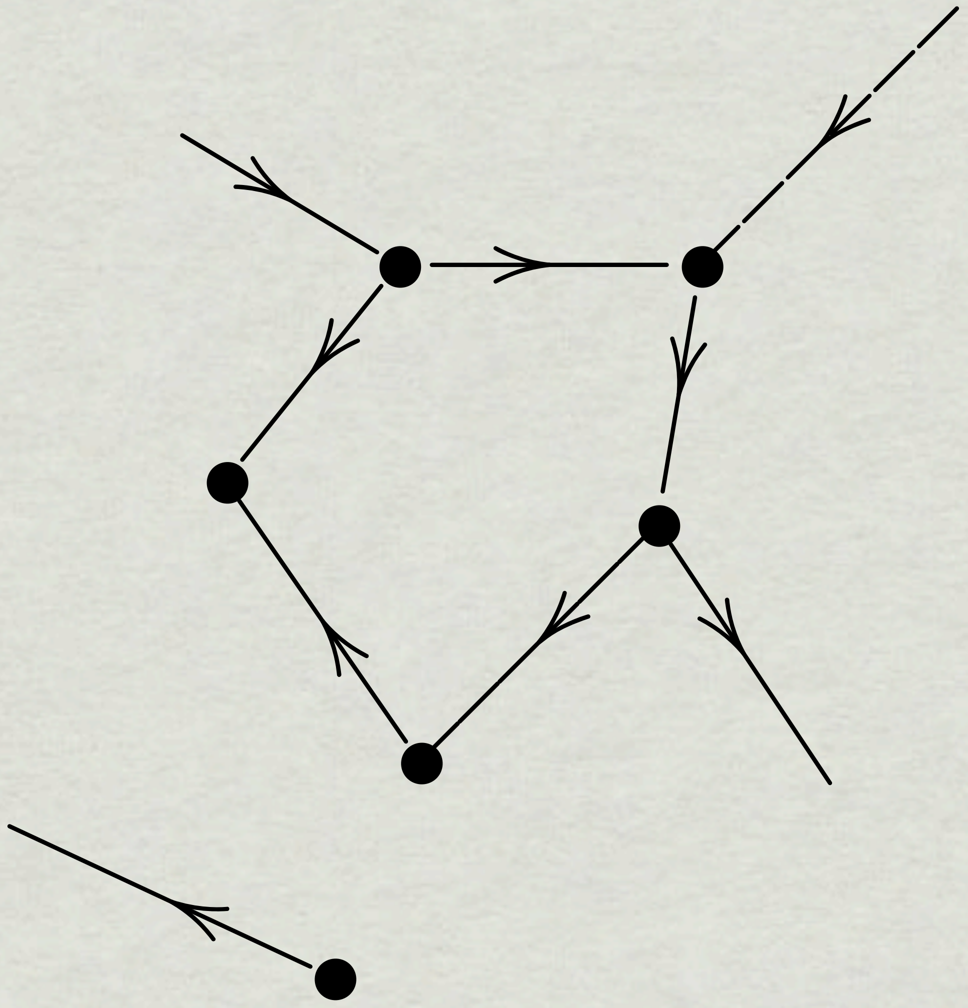
Progressive graphs

There are several definitions of *plane graph* we could use.

We chose Joyal and Street's *progressive plane graphs*.

A **progressive graph** $\Gamma = (G, G_0)$ consists of a Hausdorff space G, a finite subset $G_0 \subseteq G$ of **nodes** such that $G \setminus G_0$ is a finite collection of **edges**, each homeomorphic to $(0,1)$.

- Edges have directions.
- No cycles.



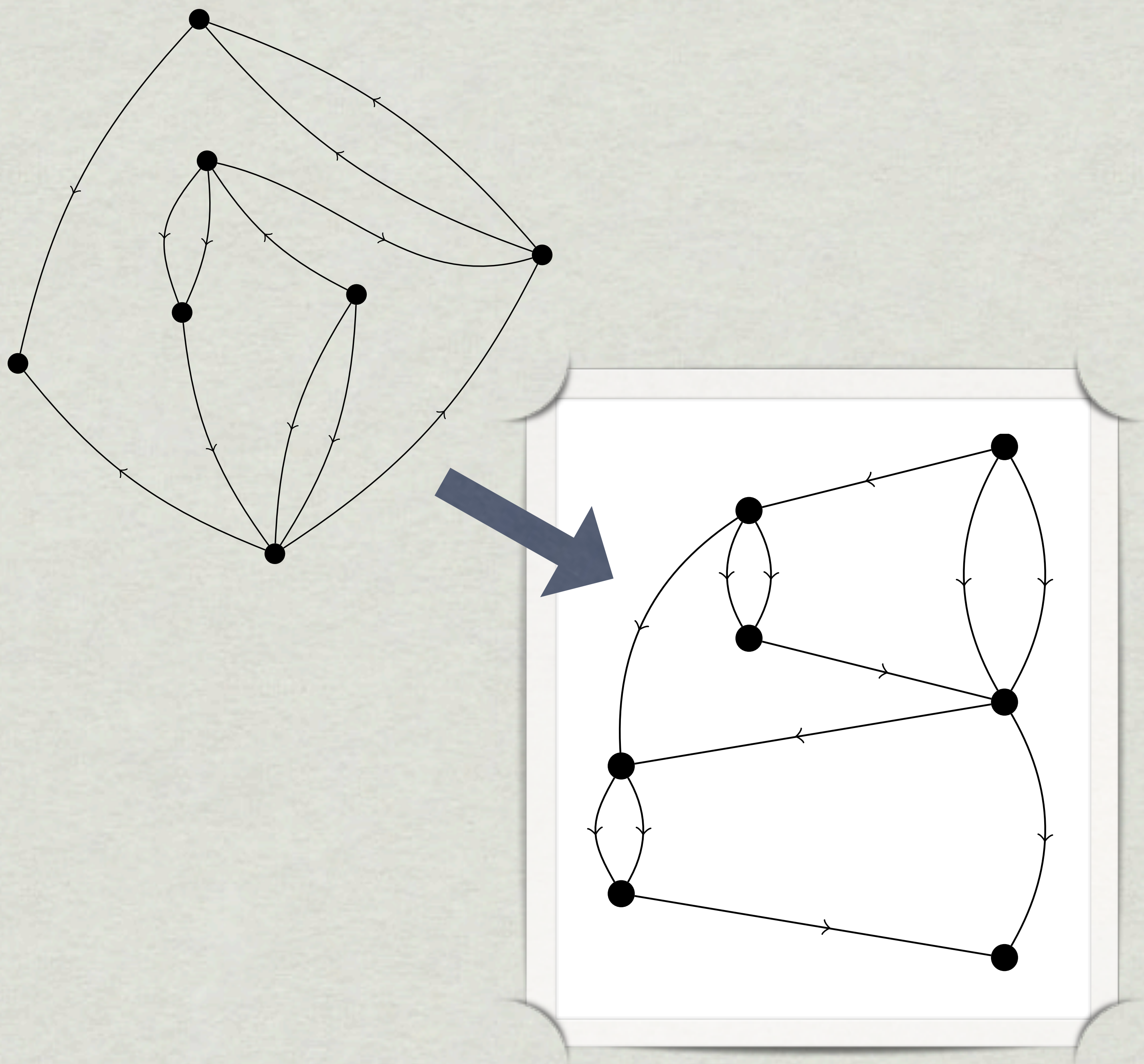
Joyal and Street. *The geometry of tensor calculus I*. Advances in Mathematics 1991.

Progressive plane graphs

A **progressive plane graph** is a progressive graph* Γ, together with an embedding $\iota : \Gamma \hookrightarrow \mathbb{R}^2$ such that:

- Edges point down: sources are higher than targets.
- The second projection $\mathbb{R}^2 \to \mathbb{R}$ is injective on each edge: no doubling-back.

Observe that images are compact subsets of the plane.



*We are assuming here that each edge of $\Gamma = (G, G_0)$ has two endpoints in G_0.

Deformation

Progressive graphs $\Gamma = (G, G_0)$ and $\Delta = (D, D_0)$ are **isomorphic** if $G \cong D$ induces a bijection on the nodes. Write it $\Gamma \cong \Delta$.

A ppg Γ,ι is a **deformation** of Δ,κ if $\Gamma \cong \Delta$ and there's a continuous function $h : G \times [0,1] \to \mathbb{R}^2$ such that:

- $h(G,0) = \iota G$
- $h(G,1) = \kappa D$
- $h(-,t)$ is an embedding of Γ as a ppg for each $t \in [0,1]$

Deformations allow us not to worry *exactly* what our pictures look like. We speak instead of deformation classes.

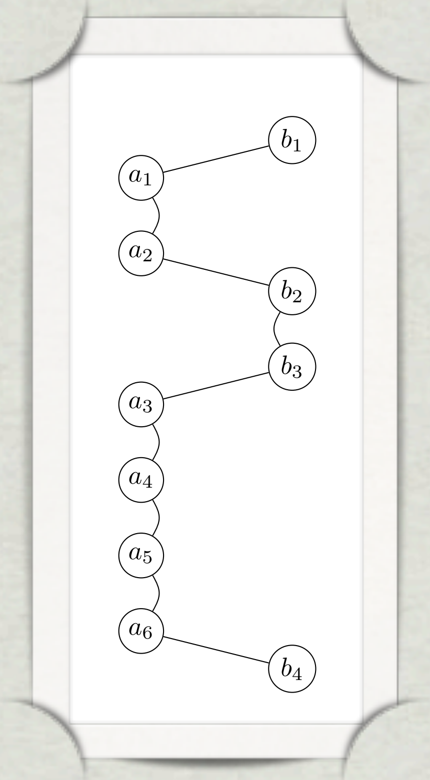
Our characterisations of schedule, interleaving graph, etc. will each be a specific refinement of ppg. Each refinement comes with an equivalently refined *deformation*.

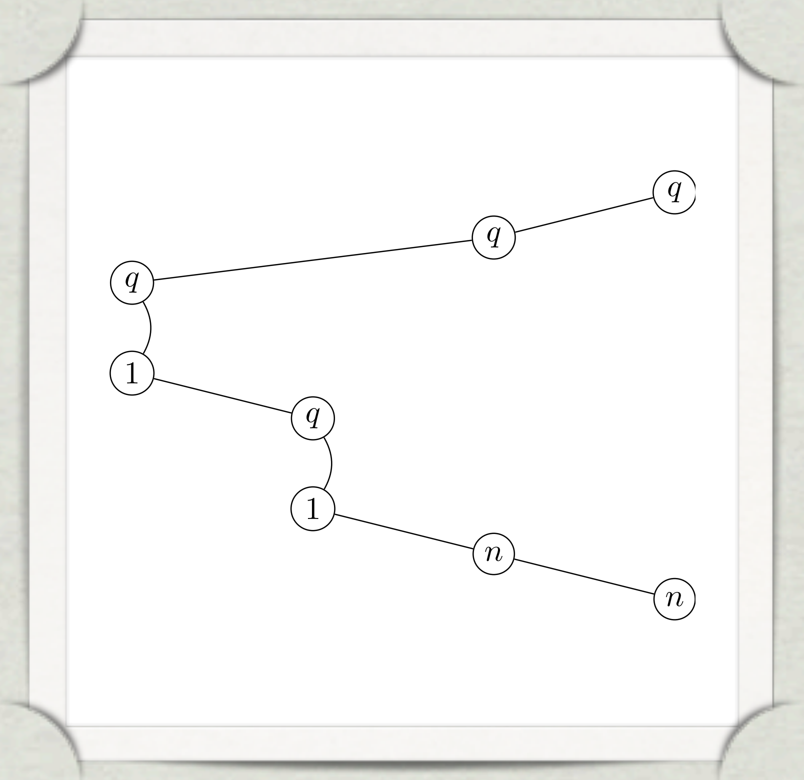
- ...
- $h(–,t)$ is an embedding as a *[insert refinement here]* graph for each $t \in [0,1]$

Paths

In this talk, our refinements are examples of *directed paths*.

- $G \cong [0,1]$.
- Each node has at most one in-edge, one out-edge.



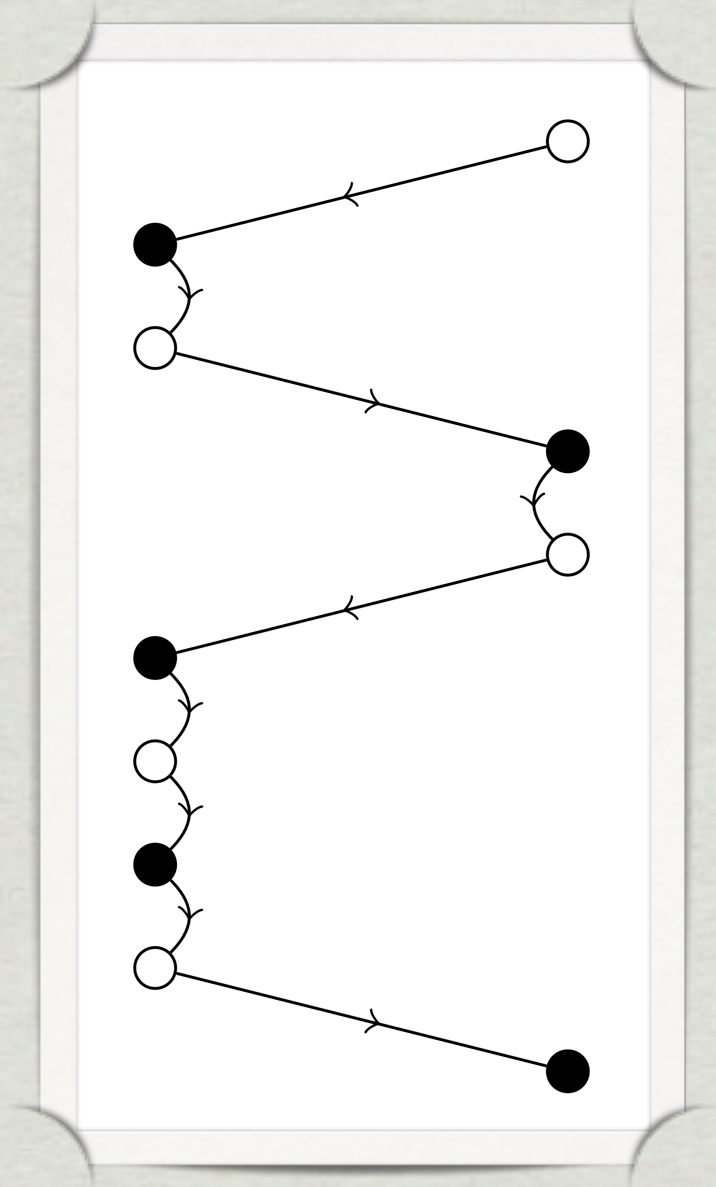


Schedules

A **⊸-schedule** $S : m \to n$ is a directed path ppg with $m+n$ nodes, positioned on left and right boundaries of some $[u_1,u_2] \times \mathbb{R}$, such that:

- First node on right
- Subsequent nodes in pairs

We can colour nodes by O/P status (parity) to make certain things easier.



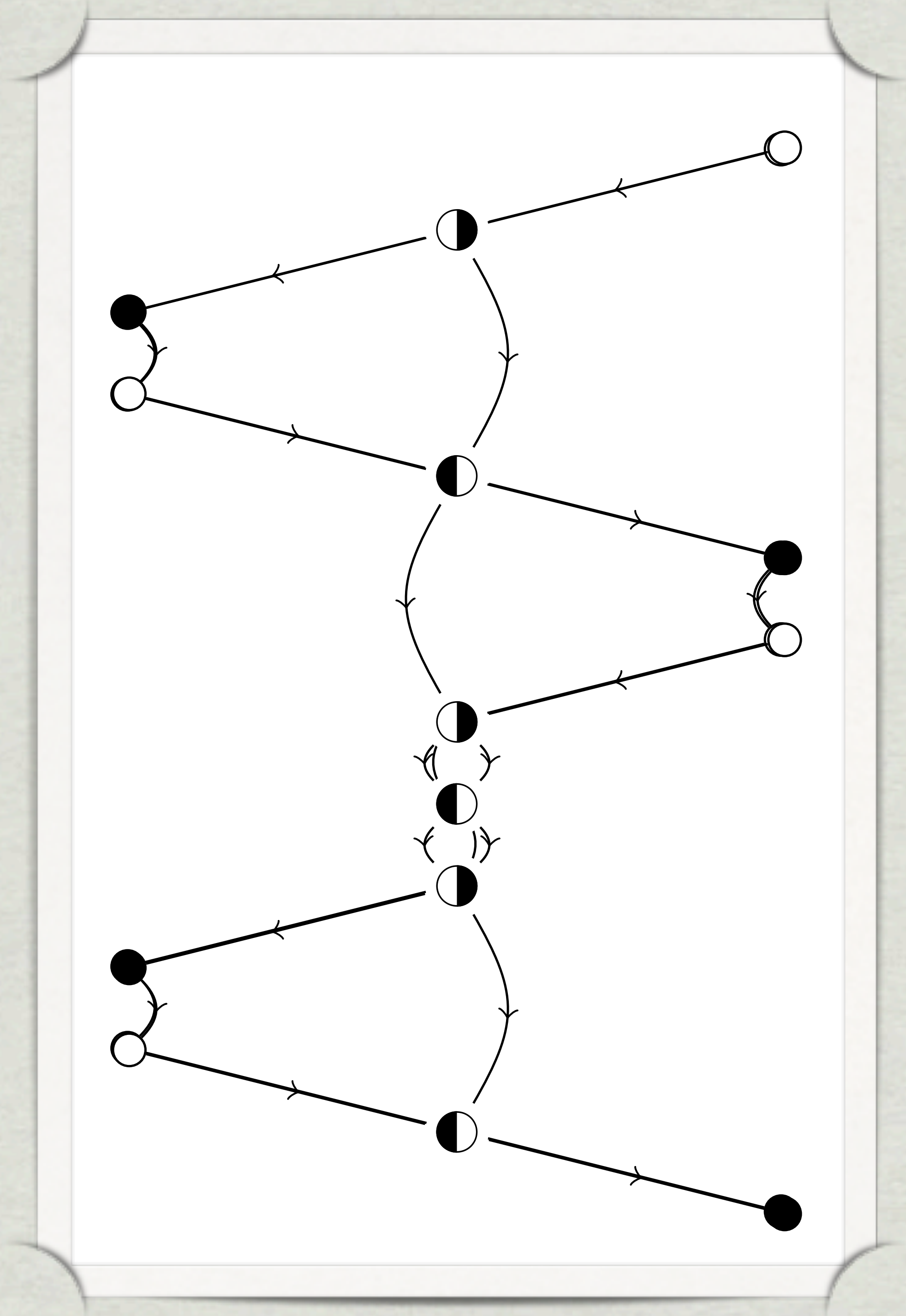
Composition

Composition of $\multimap$-schedules is a graphical procedure.

To **compose** $S : m \to n$ with $T : n \to r$; form a 2-fold composition diagram:

- Overlap middle nodes
- Remove central "⦚" edges and "declassify" nodes

Resultant $\multimap$-schedule is called $S \| T : m \to r$.

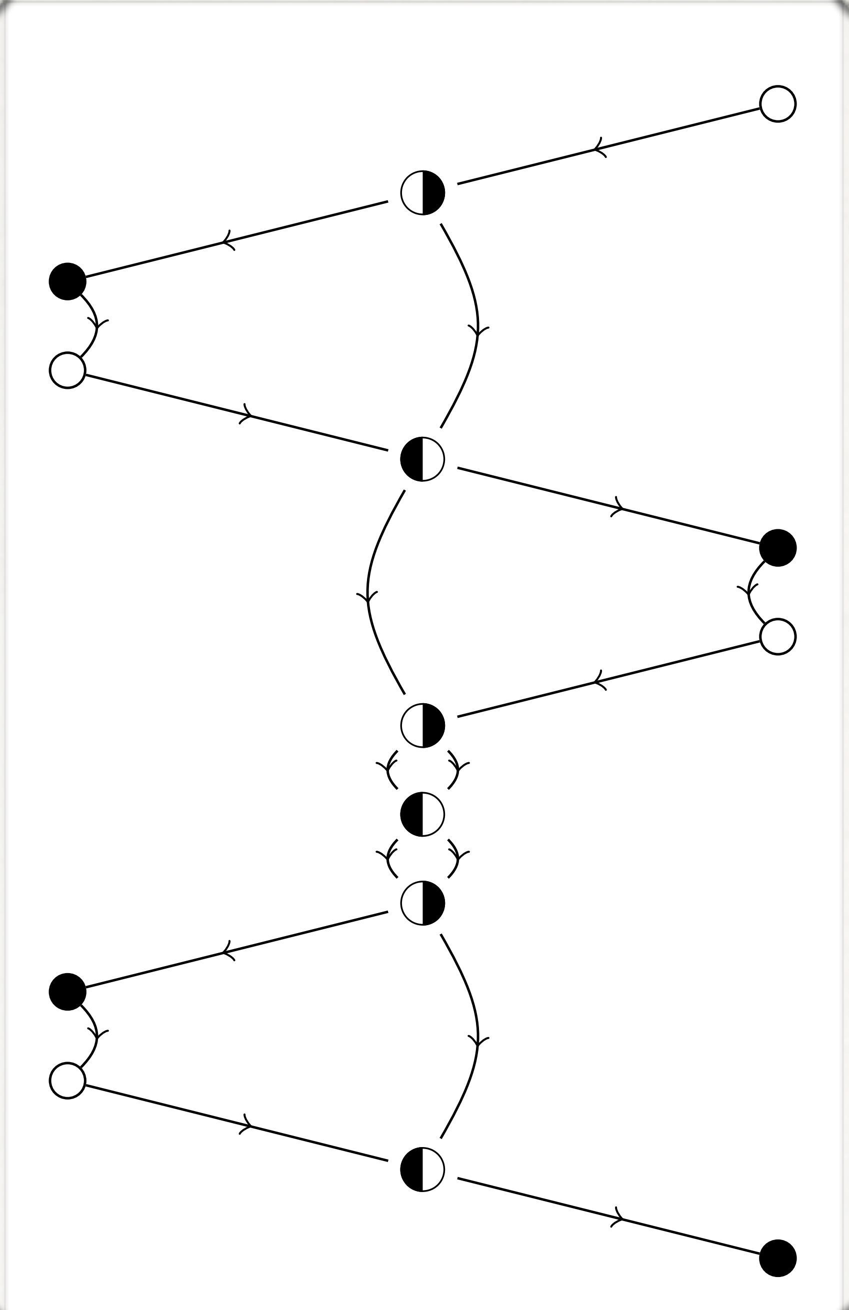


This process uses properties of graphs in the plane

We know that we approach the first internal node *from the right* and must leave *to the left*

We know, therefore, that the "⦚" edges exist

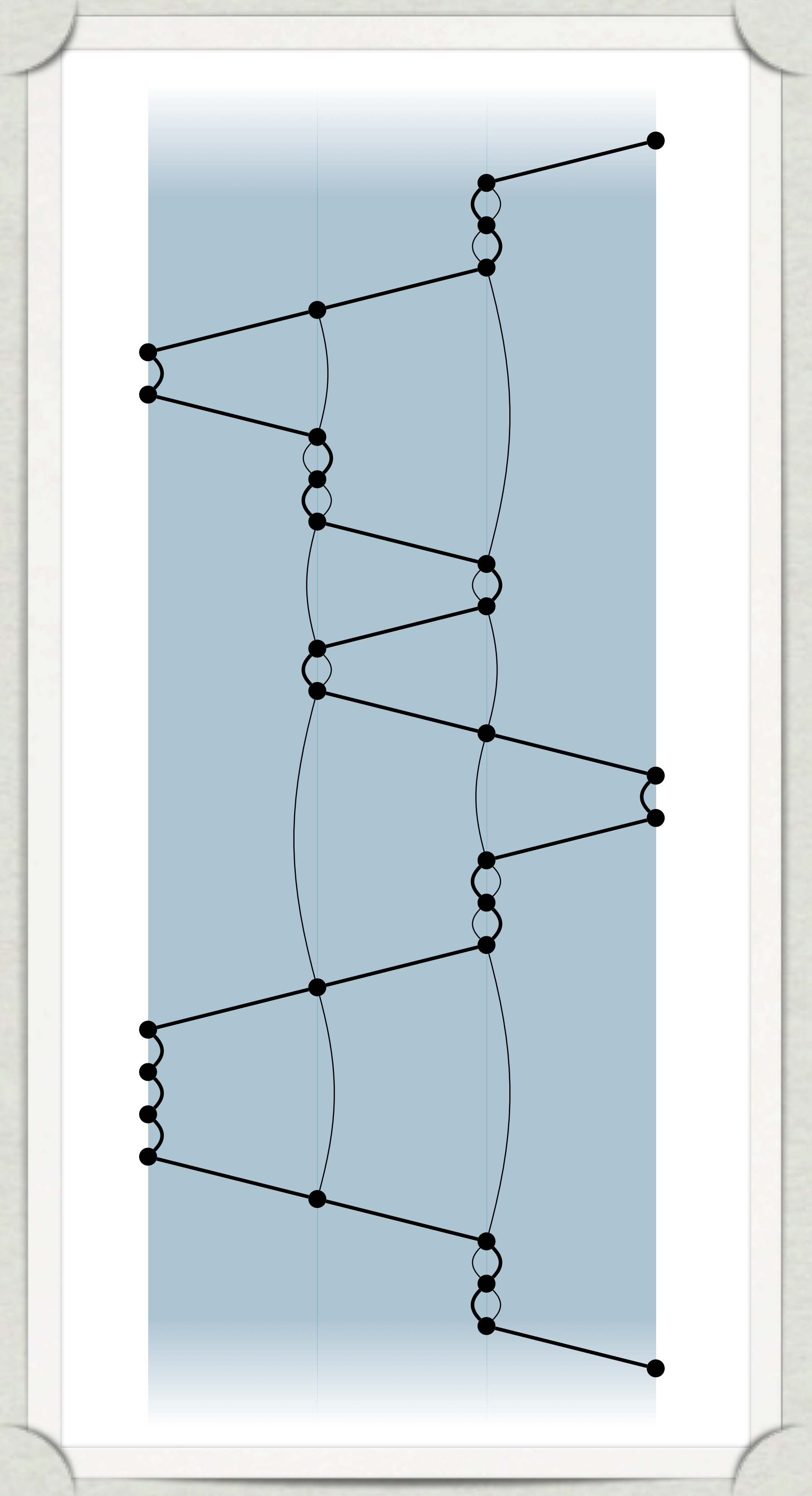
It's the "⦚" whose first edge leaves the first internal node *to the right*...



We can then argue graphically to get our first result.

Composition of ⊸-schedules is associative.

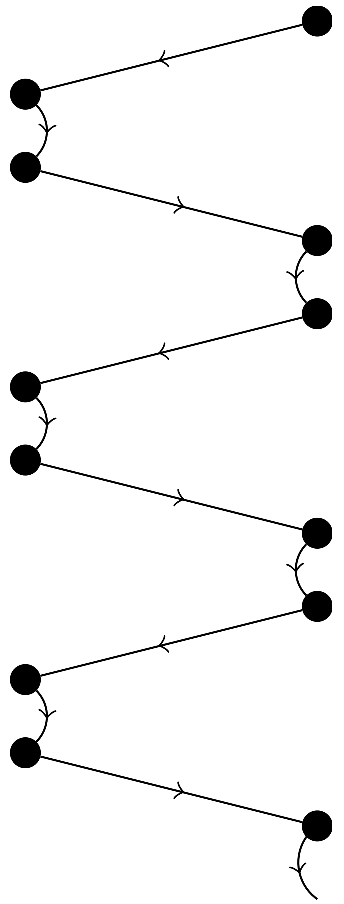
- Form 3-fold composition diagram.
- One order of composition removes right-hand "⦚" first...
- ...the other order removes left-hand "⦚" first

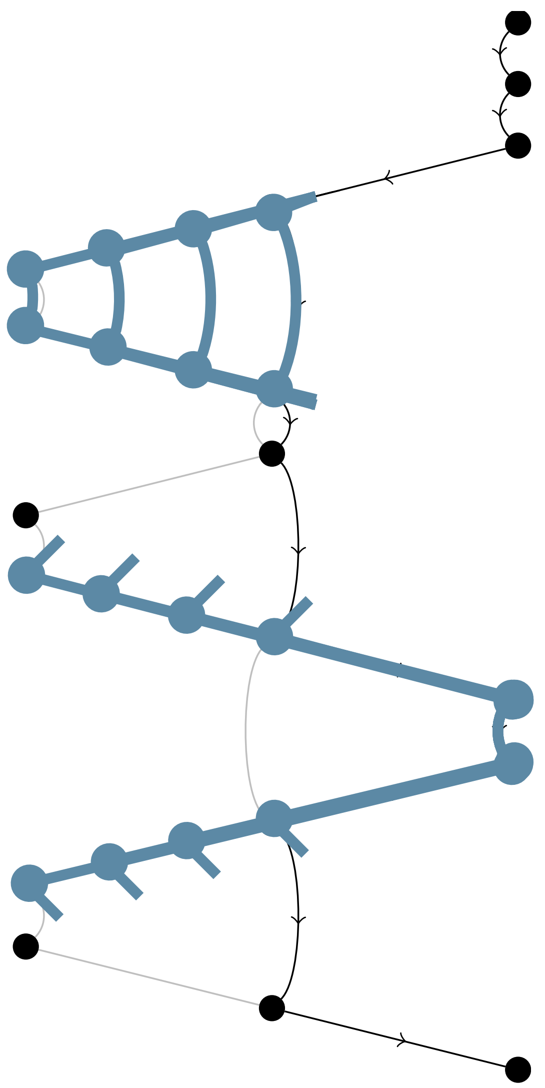


Copycat ⊸-schedules

These are the "most alternating" ⊸-schedules satisfying the conditions.

They're identities of composition.





Category of $\multimap$-schedules

Objects are nonnegative integers.

A morphism $m \to n$ is a schedule $S : m \to n$.

Composition and identities as described.

It's isomorphic to the category of schedules in Harmer et al.'s paper.

Now in games

A **game** A is a graded set with predecessor function; a forest.

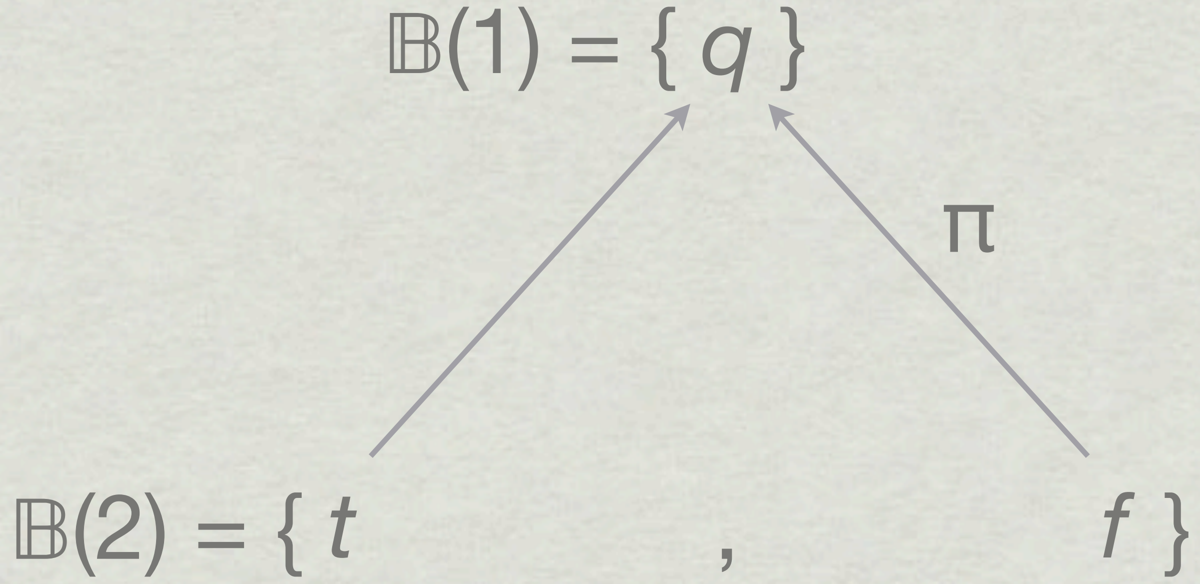
- $A(1) \xleftarrow{\pi} A(2) \xleftarrow{\pi} \cdots$

If i is odd, elements of $A(i)$ are **O-positions**, otherwise they are **P-positions.**

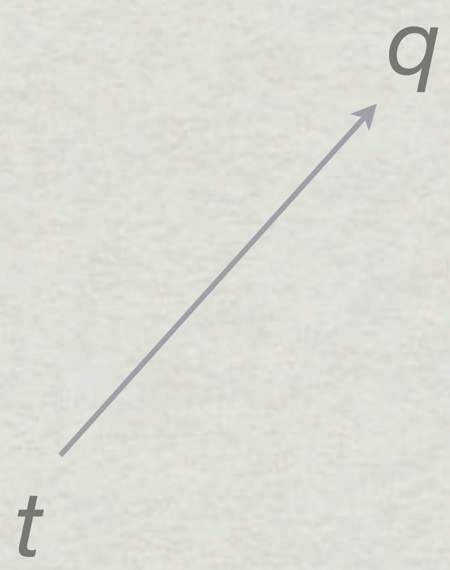
$A \setminus \pi(A)$ are **leaf positions.**

For a game A, a **strategy** σ on A, written $\sigma{:}A$, is a graded subset $\sigma(2k) \subseteq A(2k)$ such that:

- *Closed under double-predecessor*: $\pi^2(\sigma(2k+2)) \subseteq \sigma(2k)$
- *Deterministic*: $x,y \in \sigma(2k)$ and $\pi x = \pi y$ then $x = y$



𝔹, THE GAME OF BOOLEANS



THE STRATEGY *t*:𝔹

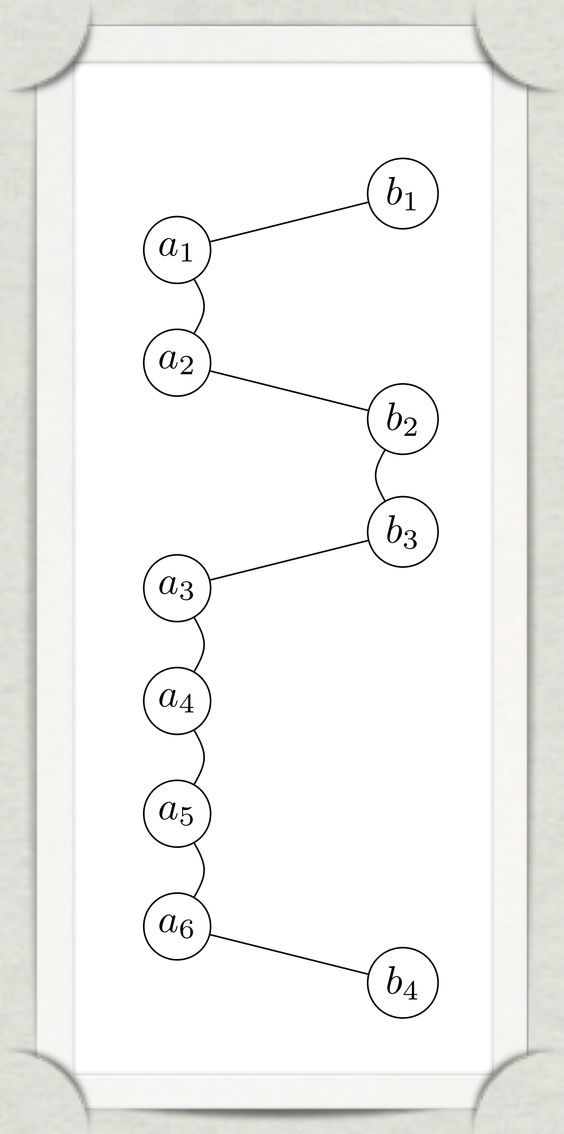
⊸-schedules in games

Suppose we have games A and B with a_m in $A(m)$ and b_n in $B(n)$, and a ⊸-schedule $S : m \to n$. Then we have a way to **label** all nodes of S.

- Label bottom-left node a_m
- Label bottom-right node b_n
- Label each node above using π_A and π_B

We denote such a labelled ⊸-schedule (S, a_m, b_n).

Labelled ⊸-schedules can be **composed**.



⊸-games

Given games A and B, the game $A \multimap B$ is given by the diagram

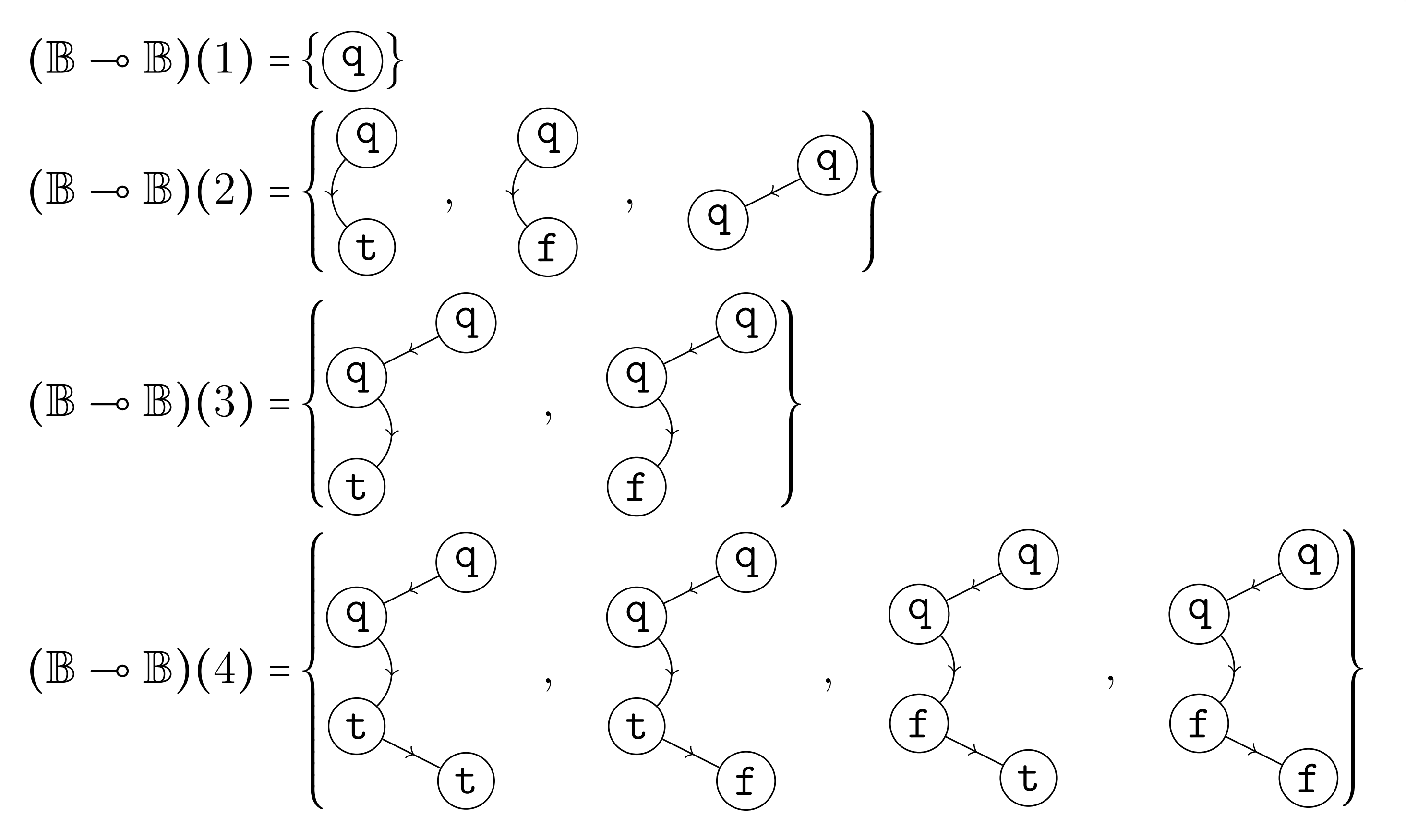
- $(A \multimap B)(1) \leftarrow (A \multimap B)(2) \leftarrow \cdots$

$(A \multimap B)(k)$ is the set of labelled ⊸-schedules

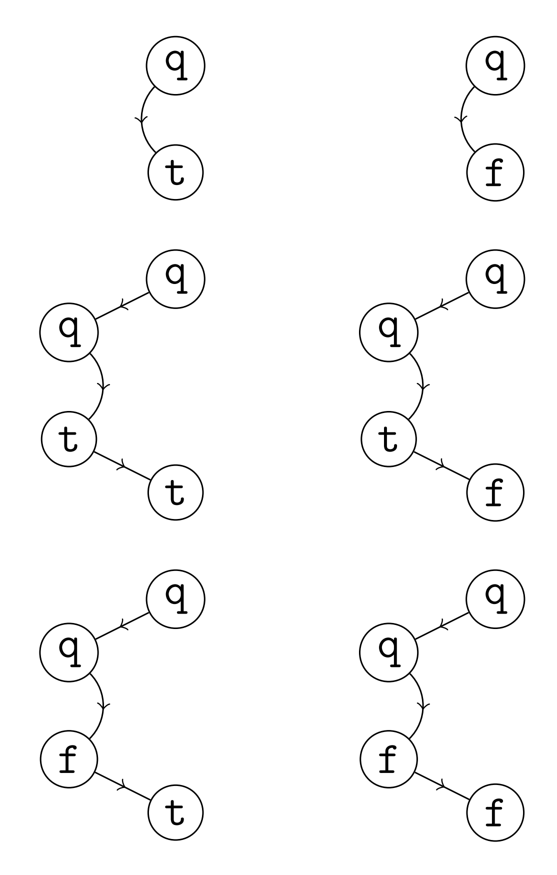
- $(\; S{:}m \to n \;, \; a_m \in \hat{A}(m) \;, \; b_n \in B(n) \;)$

such that $m + n = k$.

$\pi_{A \multimap B}$ is given by truncation.



THE GAME $\mathbb{B} \multimap \mathbb{B}$



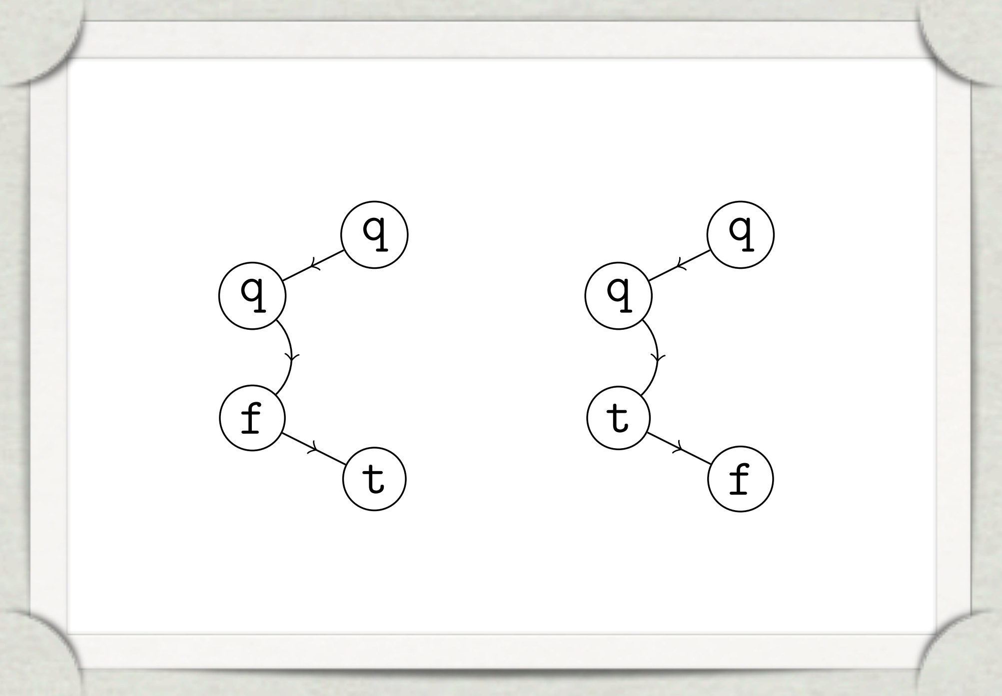
A GAME IS GIVEN BY ITS LEAF POSITIONS

Strategies

A game $A \multimap B$ can be given by a set of complete plays; labelled $\multimap$-schedules.

A strategy $\sigma : A \multimap B$ can be given by a set of labelled $\multimap$-schedules:

- Maximal even-length
- Even-length longest common truncation



THE STRATEGY $\neg : \mathbb{B} \multimap \mathbb{B}$

Composition of strategies

Strategies are **composed** by composing all composable pairs of labelled ⊸-schedules.

- $\sigma \| \tau = \{ (S \| T, a, c) \mid (S,a,b) \in \sigma,\ (T,b,c) \in \tau \}$

It's associative because composition of ⊸-schedules is.

Category of games

Objects are games.

Morphism $A \to B$ is a strategy on $A \multimap B$.

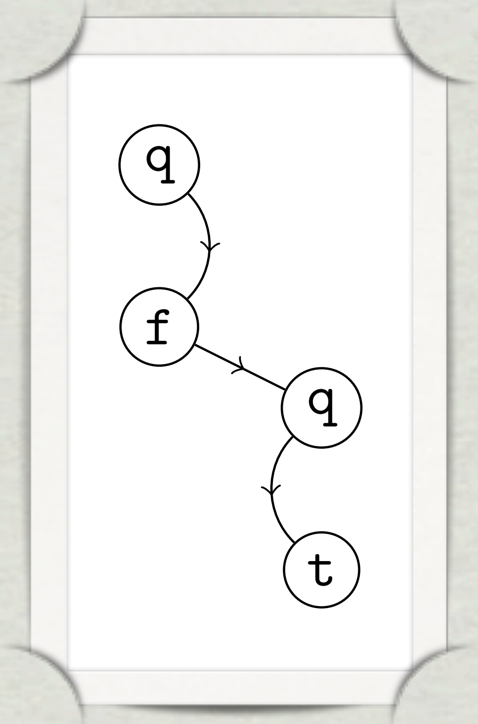
Identity morphism $A \to A$ is the strategy of labelled copycat schedules (S,a,a).

It's isomorphic to Harmer et al.'s category of games.

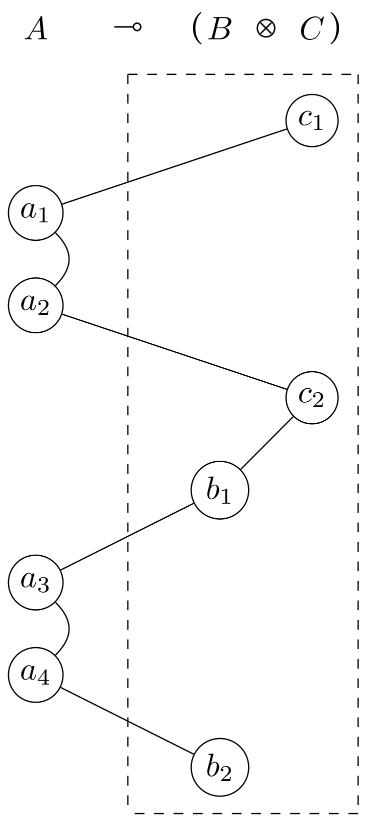
⊗-schedules and ⊗-games

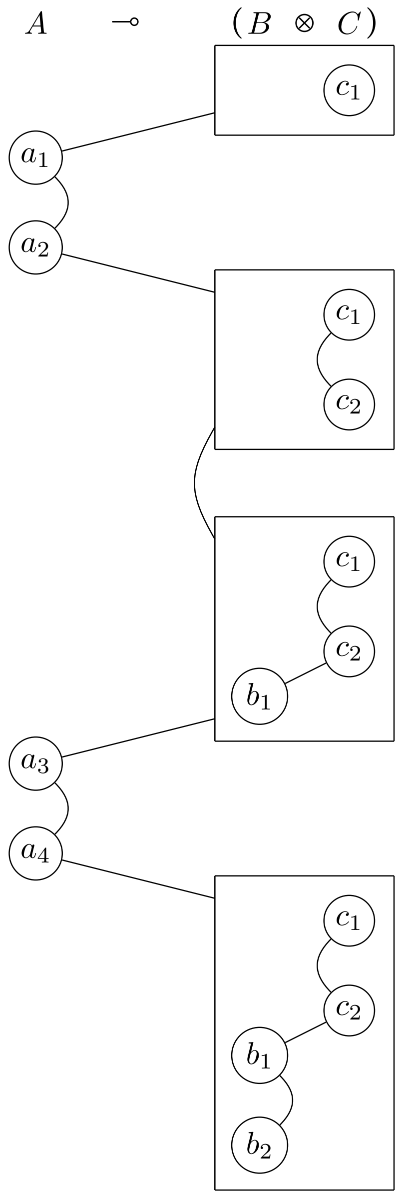
⊗-schedules are like ⊸-schedules, but don't have to start on the right.

For games A,B, the game $A \otimes$ B is given by sets of labelled ⊗-schedules under truncation.



Interleaving graphs

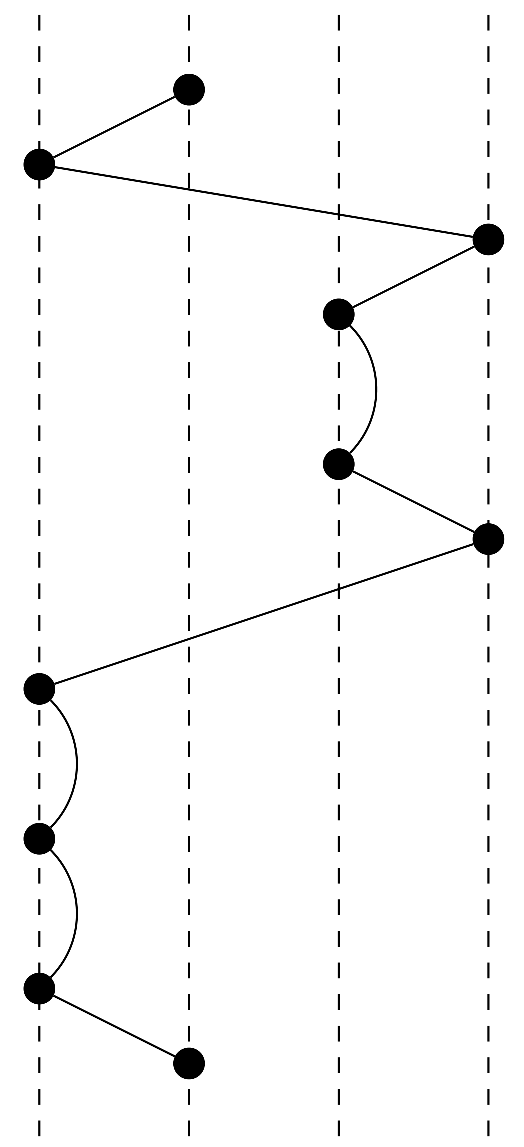




By our definitions, games with more than two components have positions given by schedules, whose nodes are labelled by schedules, whose nodes are labelled by schedules, whose nodes are labelled by...

We want to be able to use an ***n*-interleaving graph** representation for any appropriate *n*.

- How can we characterise which of these actually describe the interleaving in a game?
- How do they relate to the "nested schedule" representation?

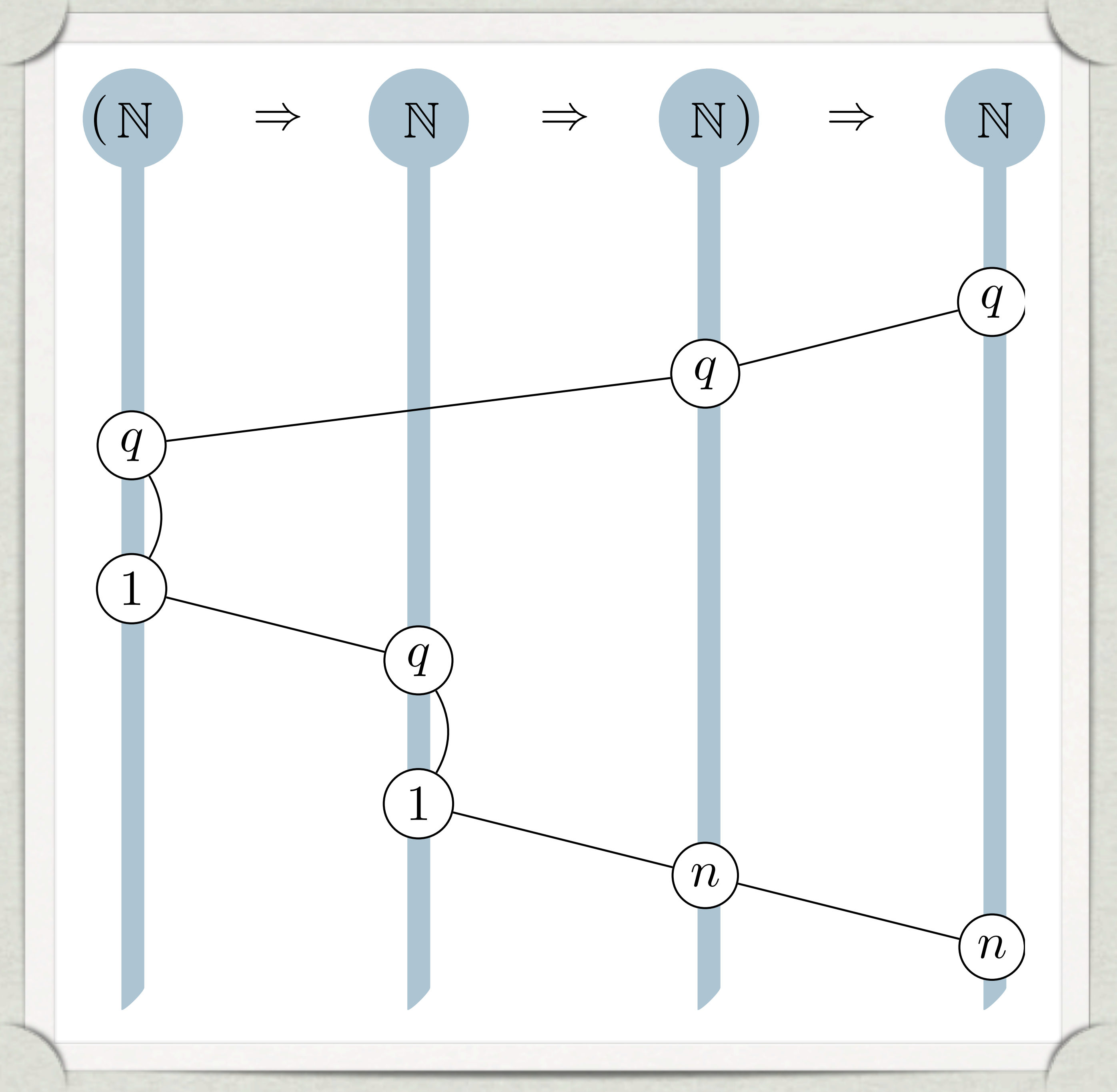


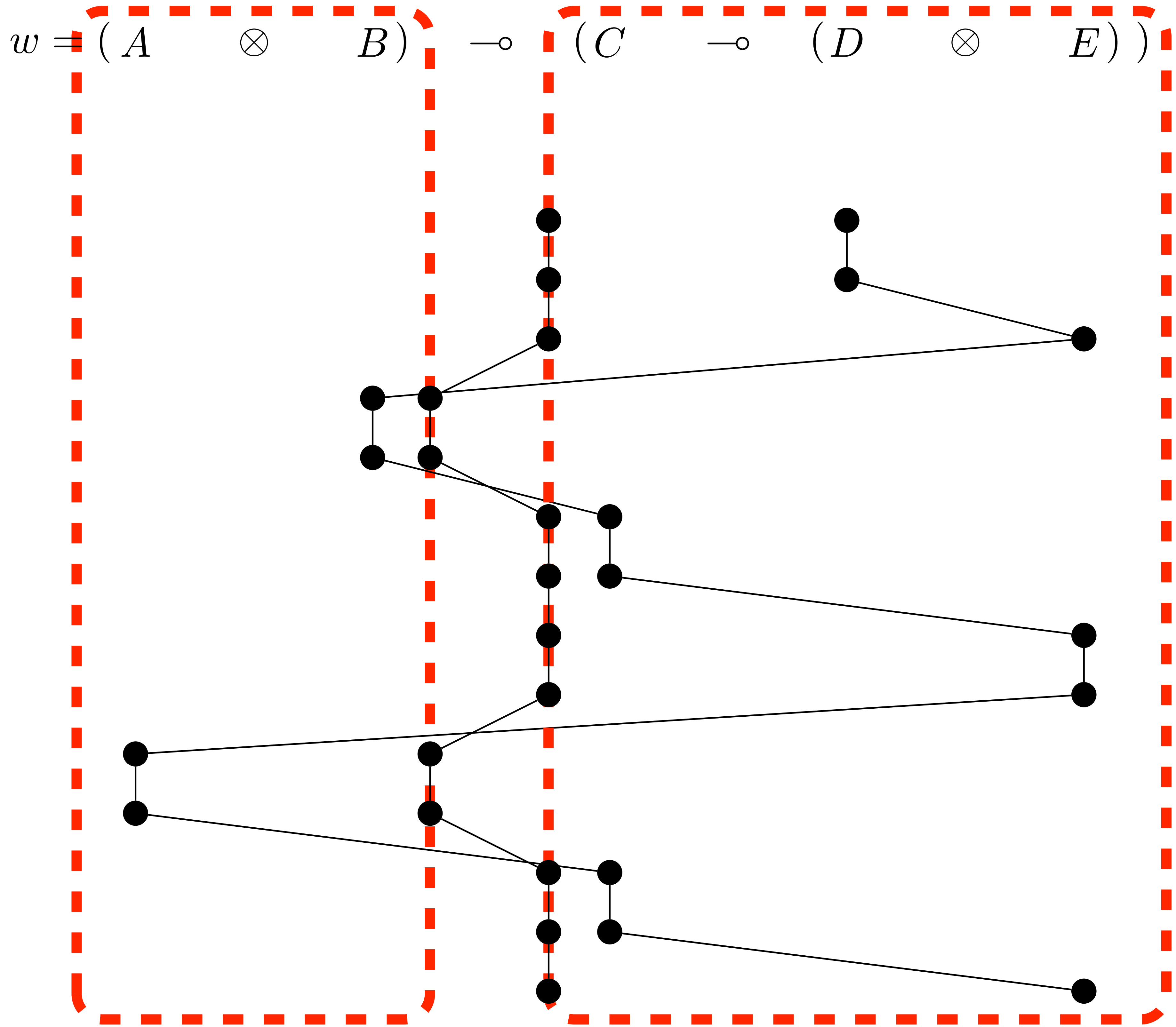
Suitability

Which interleaving graphs describe suitable interleavings?

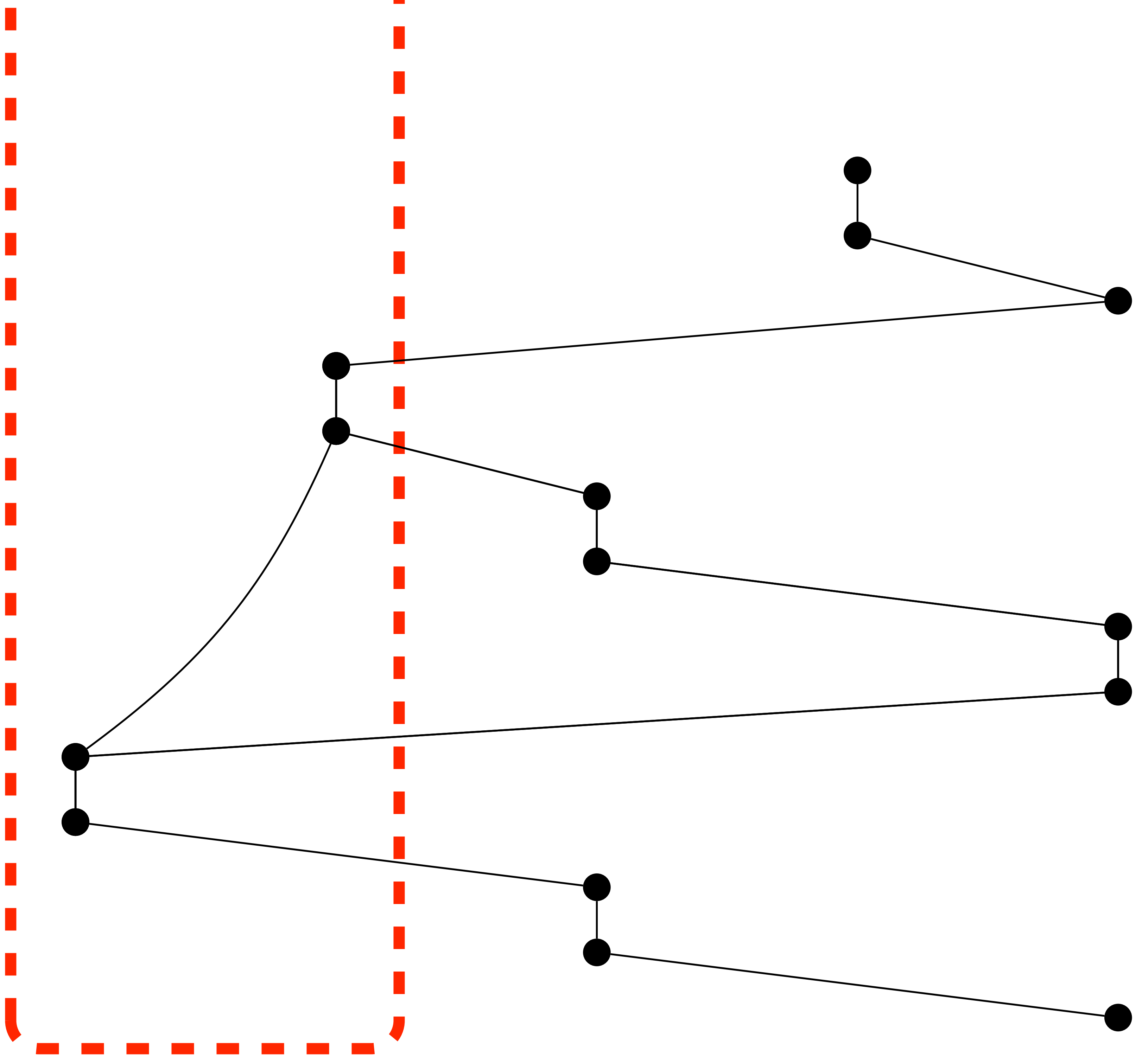
The columns of nodes in an interleaving graph correspond to letter symbols in the word describing the game.

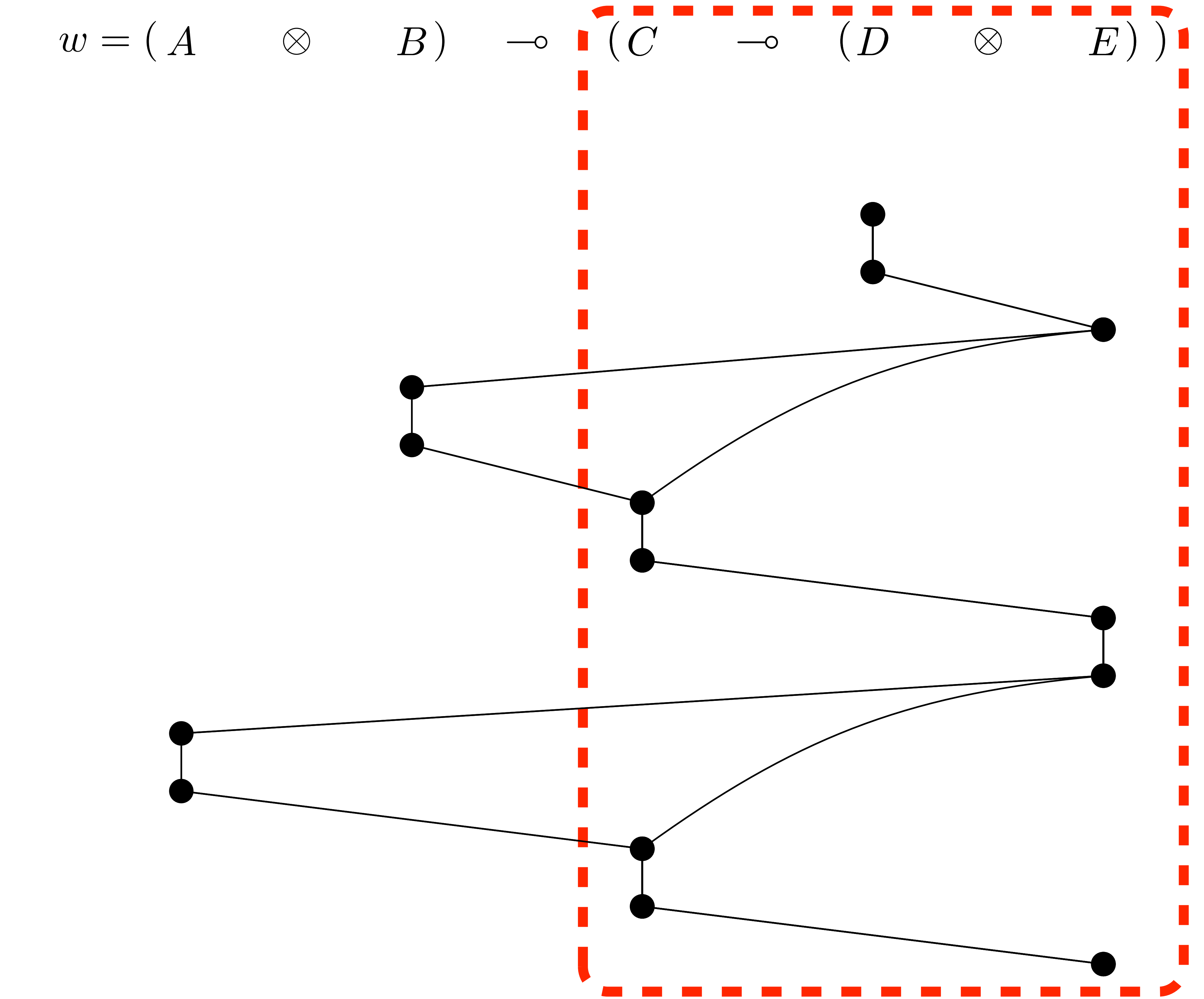
For a word *w* containing a symbol *X*, we'll speak of ***X*-nodes** of the graph.

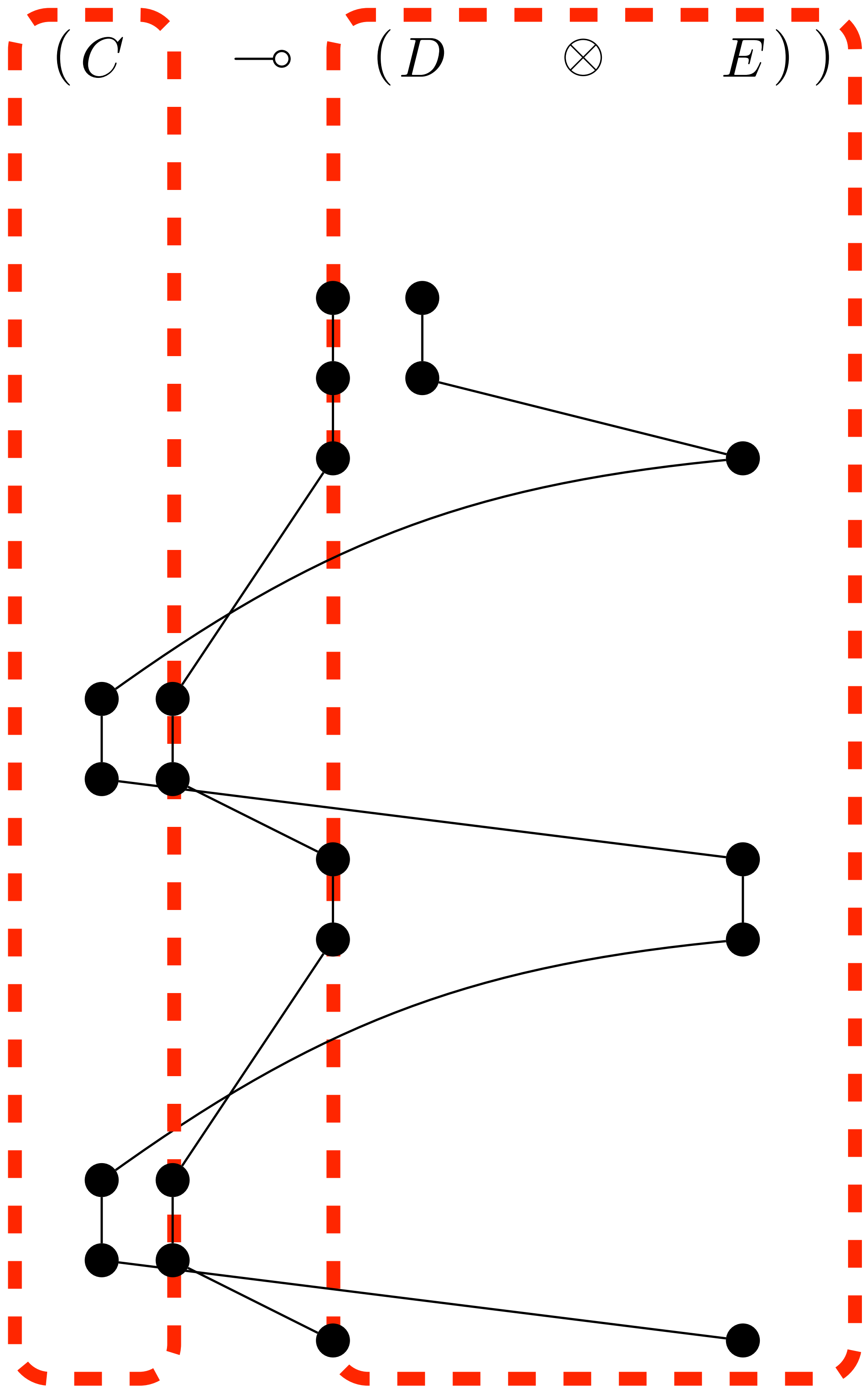



$w = (A \otimes B) \multimap (C \multimap (D \otimes E))$

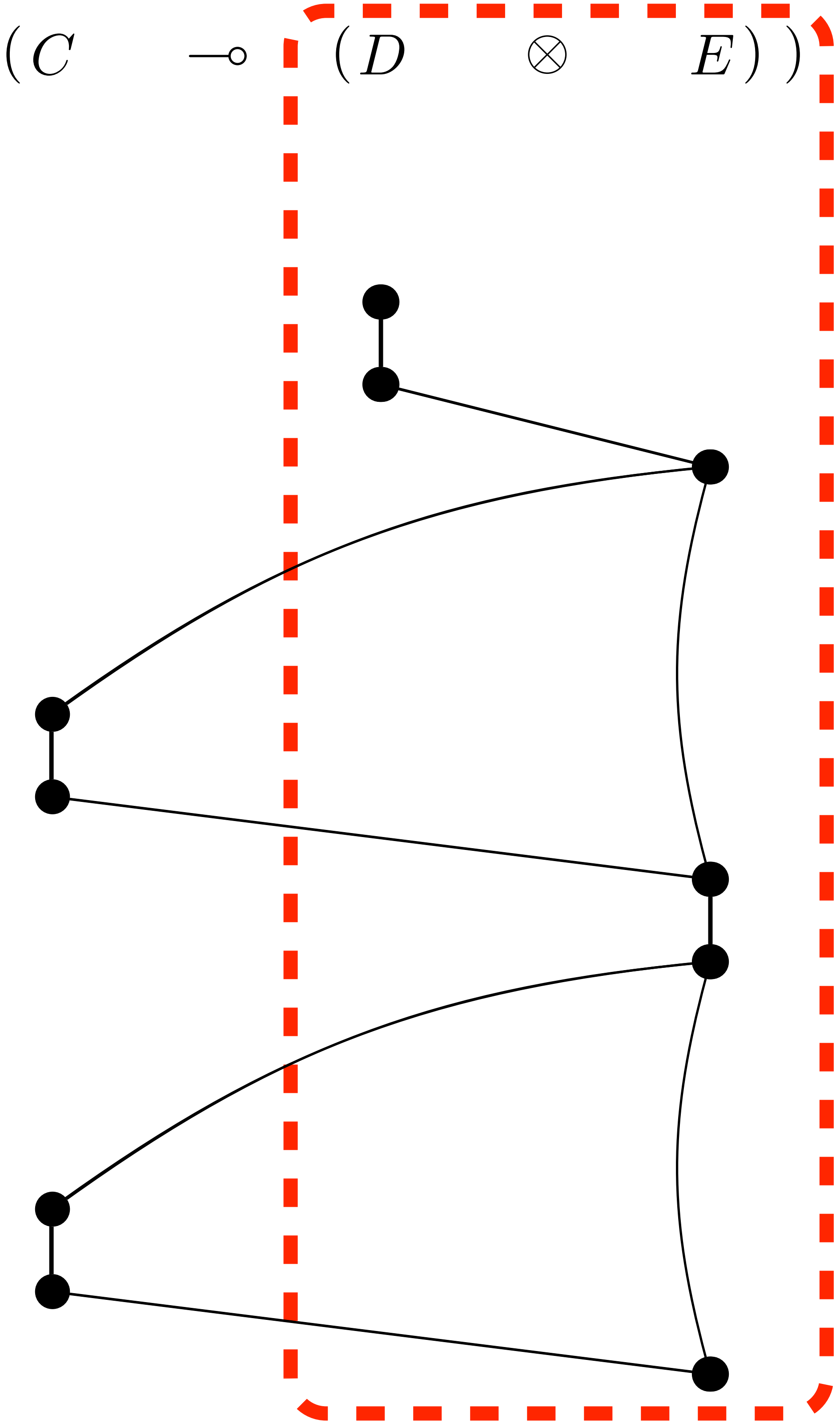
$$w = (A \otimes B) \multimap (C \multimap (D \otimes E))$$




$w = (A \otimes B) \multimap (C \multimap (D \otimes E))$


$(C \multimap (D \otimes E))$

$(C \multimap (D \otimes E))$



An n-interleaving graph Z is **suitable** for a word w with $|w| = n$, if:

- For a connective $\square$ of w, **restricting** Z to the bracketed subword $w_\square \sqsubseteq w$ and **segregating** over $\square$ gives a $\square$-schedule.
- For a bracketed subword $u \sqsubseteq w$, the corresponding restriction of Z is suitable for u.

Unfolded game

Suppose a game A is built from games A_1, ..., A_n using $\multimap$ and $\otimes$ as in a word w.

The **unfolded game** $\tilde{A}$ is is given by labelled interleaving graphs suitable for w, under truncation.

How does A relate to $\tilde{A}$?

We'll transform a position of A into a position of $\tilde{A}$ by repeatedly "unfolding" a schedule labelling one of A's nodes.

This will turn out to be an isomorphism of games.

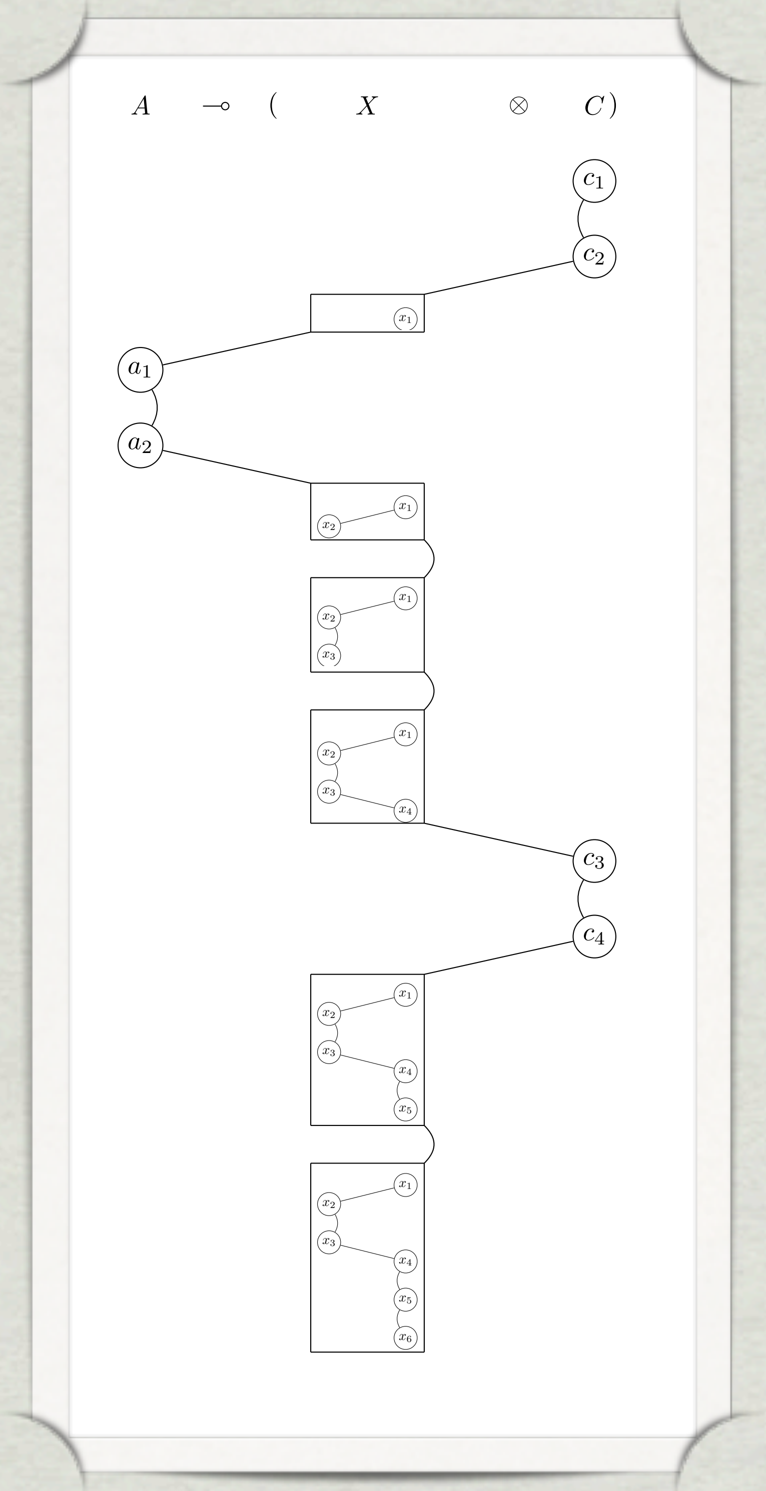
- "Folded" games have nice associative composition
- "Unfolded" games resemble intuitions, literature

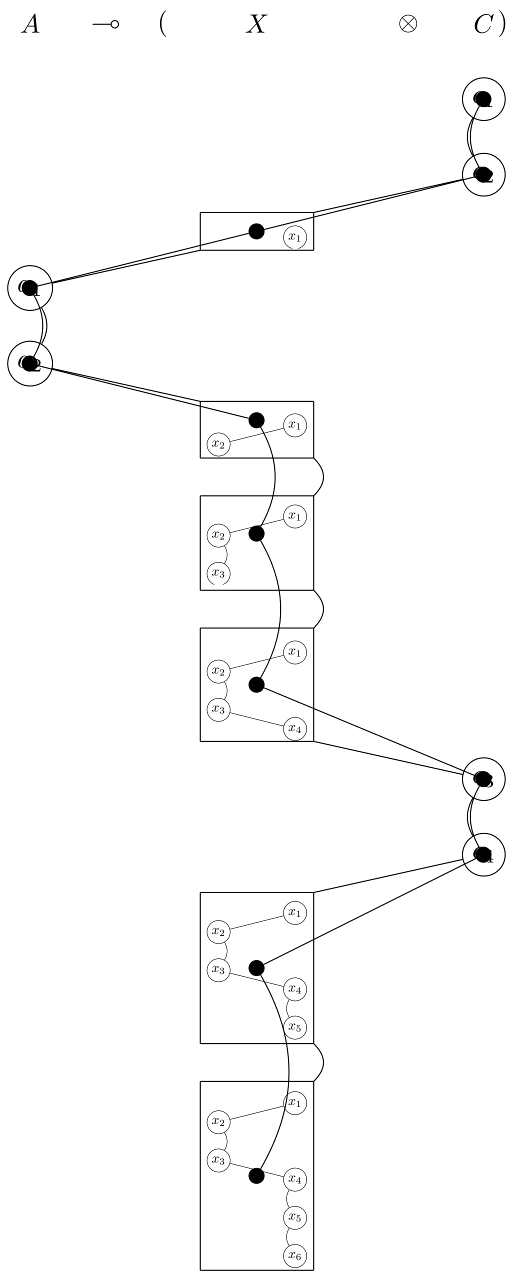
Unfolding

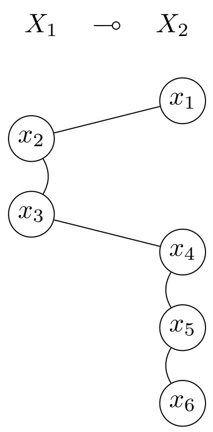
A □-schedule is a 2-interleaving diagram suitable for the word
A □ B.

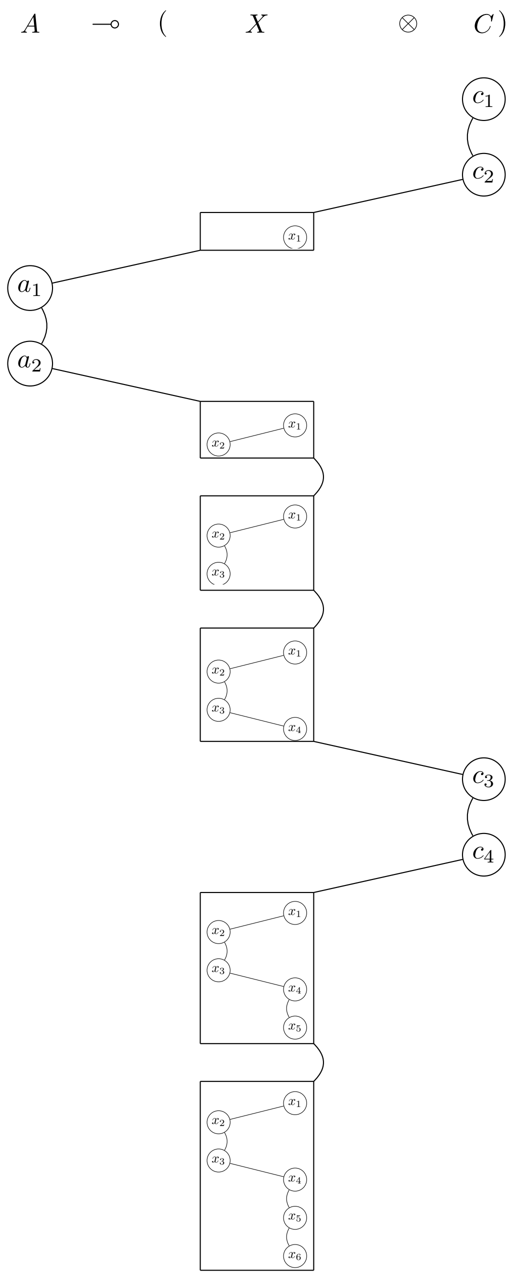
Let w be a word containing letter *X* with $|w| = n$.

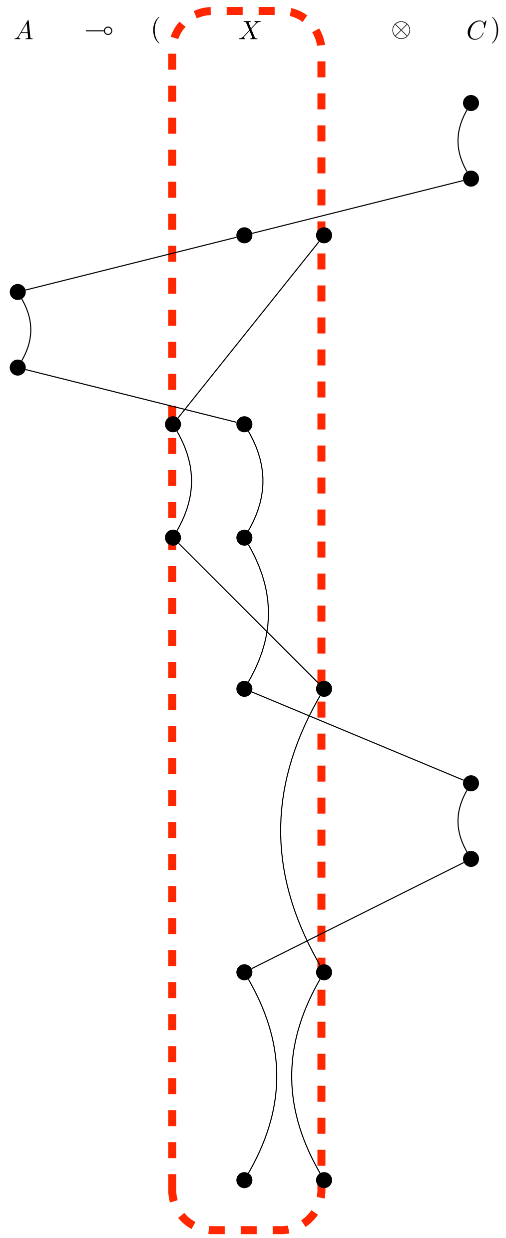
Let *Z* be an *n*-interleaving graph suitable for w whose *X*-nodes are labelled with a schedule.

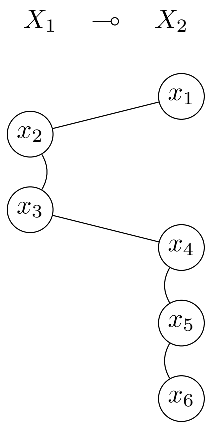


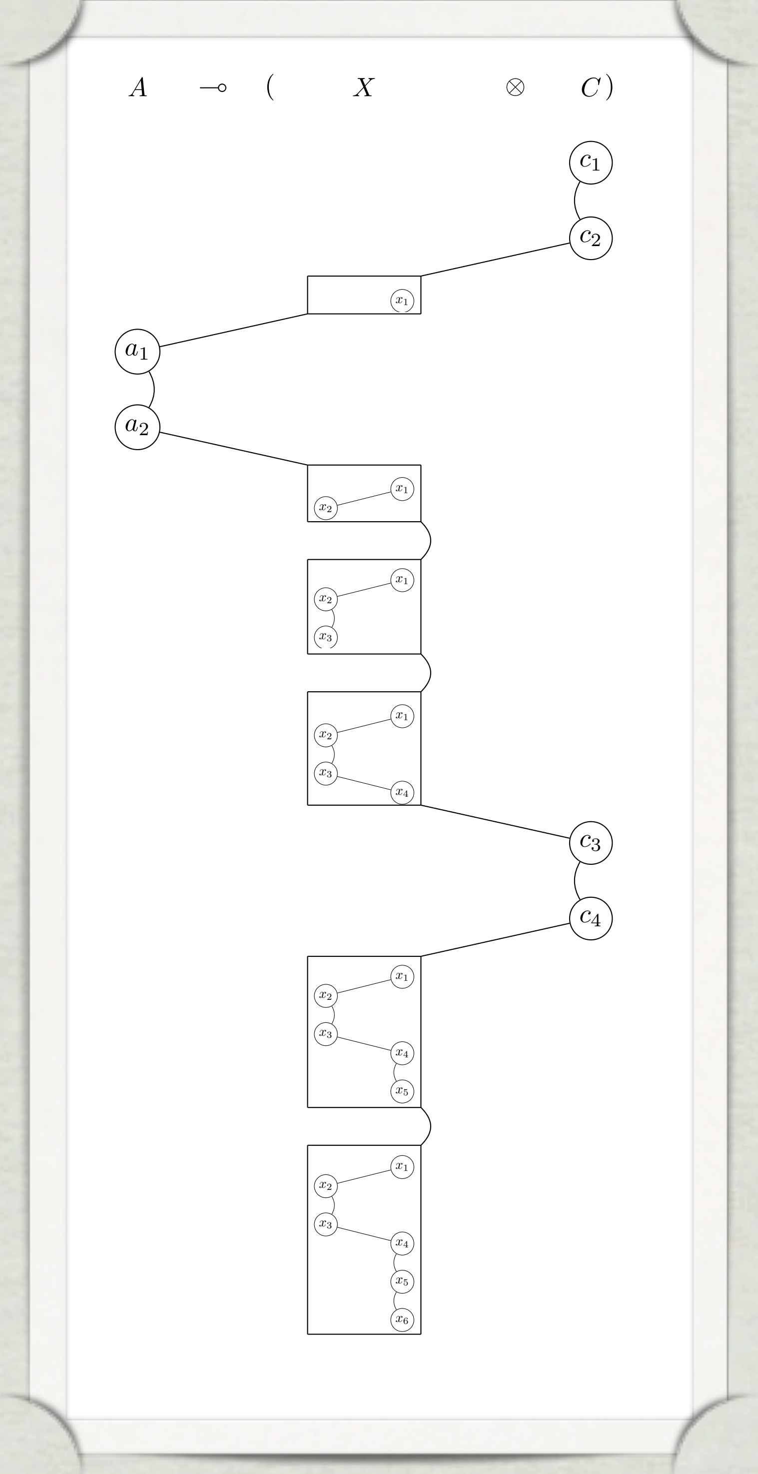

$A \multimap (X \otimes C)$

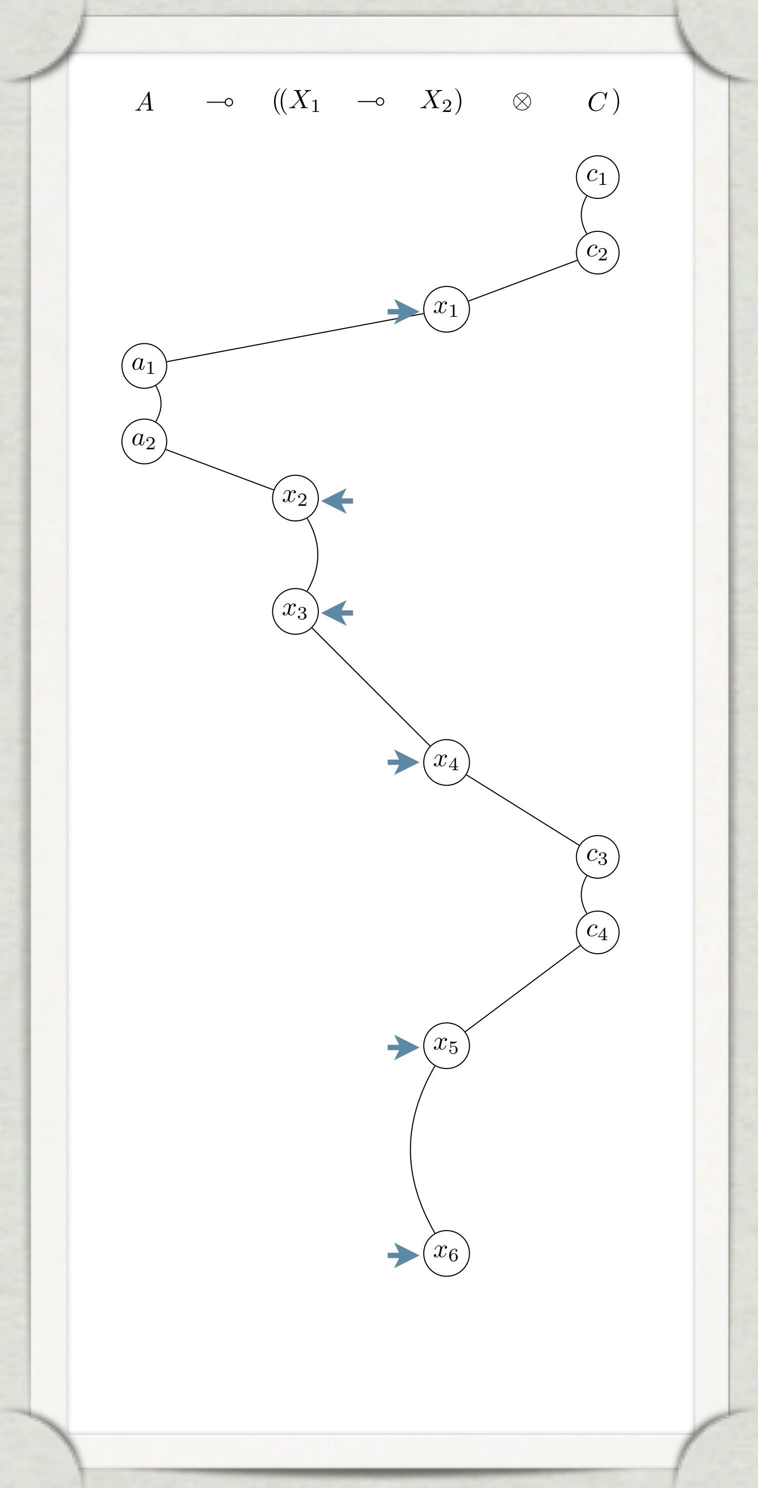

$X_1 \multimap X_2$

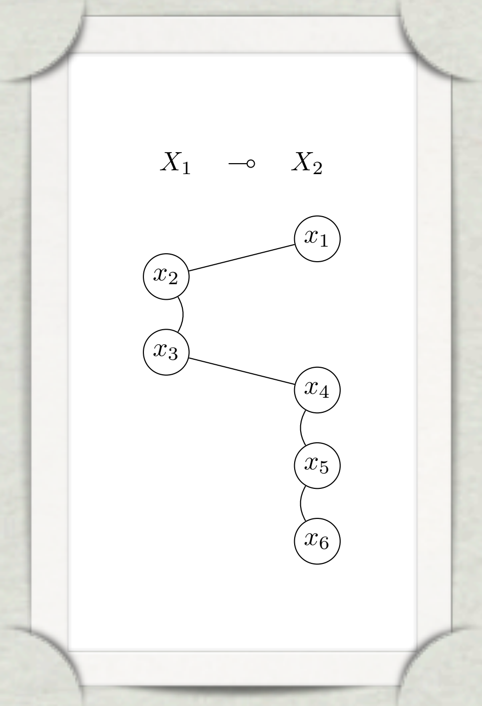

A ⊸ (X ⊗ C)
c_1
c_2
a_1
a_2
c_3
c_4


A ⊸ (X ⊗ C)


X_1 ⊸ X_2
x_1
x_2
x_3
x_4
x_5
x_6


A ⊸ (X ⊗ C)
c1
c2
a1
a2
c3
c4


A ⊸ ((X1 ⊸ X2) ⊗ C)
c1
c2
x1
a1
a2
x2
x3
x4
c3
c4
x5
x6


X1 ⊸ X2
x1
x2
x3
x4
x5
x6

One step of this is an isomorphism of games.

- Unfolding respects truncation

We can do this repeatedly to produce an interleaving graphs so long as we have labels which are schedules.

The process is confluent.

- There are no critical pairs

(There’s also a **folding** process.)

An isomorphism of games

Two games are isomorphic if their unfolded forms are isomorphic.

- A game is isomorphic to its unfolded form

Now we can think of any multi-component game as having positions which are labelled interleaving graphs.

Also works with strategies.

- To compose, fold first, compose as a $\multimap$-schedule, then unfold

This isomorphism is useful because our intuitive arguments become proofs.

E.g. The game $(A \otimes B) \multimap C$ looks the same as the game $A \multimap (B \multimap C)$ when unfolded.*

- “The first move is in C, subsequent moves come in pairs in A, B or C.”

So the category of games is monoidal closed.

A symmetry for $\otimes$ can be given by a labelled 4-interleaving graph suitable for $(A \otimes B) \multimap (B \otimes A)$.

*Haven’t actually shown here that $\otimes$ is a tensor product, but it is!

Recap

A category of games whose plays are labelled diagrams.

Familiar, “deceptively” intuitive arguments give proofs in terms of the actual definitions.

Our framework is well-grounded in the literature and very extensible.

In particular, we’ve seen:

- Schedules for $\multimap$ and $\otimes$
- Interleaving graphs
- An isomorphism of graphical representations

Our arguments use fundamental properties of graphs and of the plane.

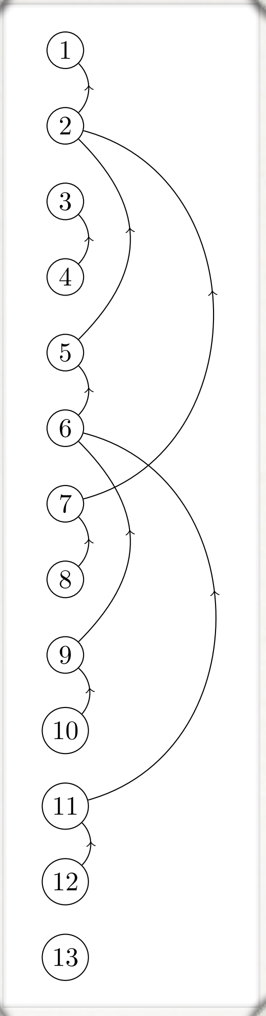
What else?

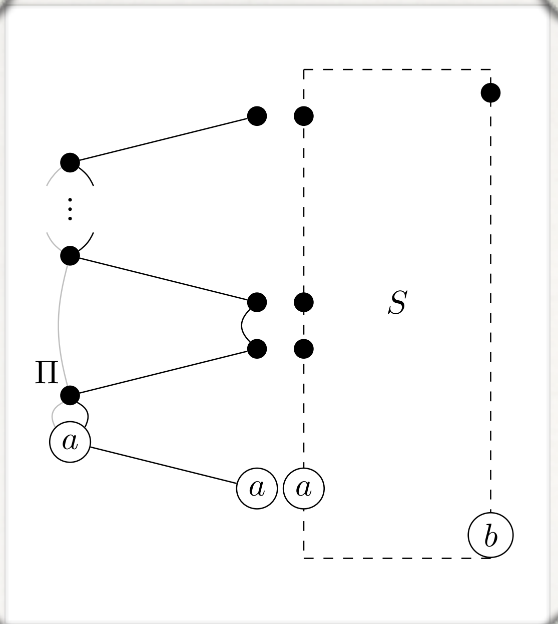
Pointer graphs for backtracking.

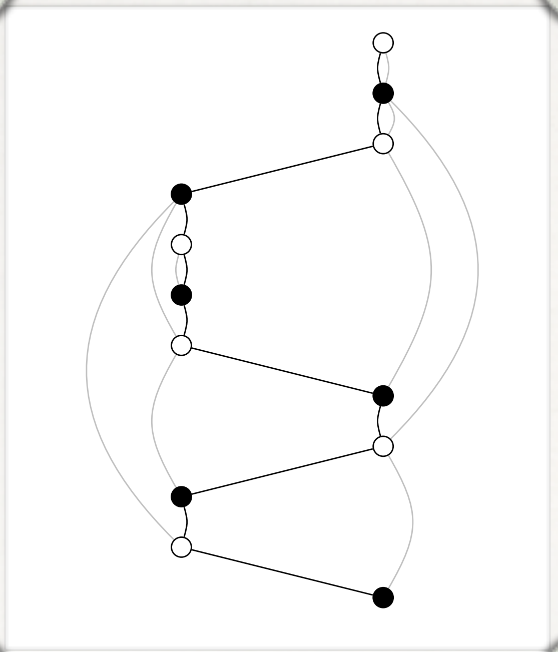
Combining pointers and schedules.

Composing and decomposing threads.

! as a comonad on the category of games.







What next?

I still have things to look at immediately.

- ? as a monad on the category of games
- Distributive laws for ? and !

Hopefully this approach will prove useful in general.