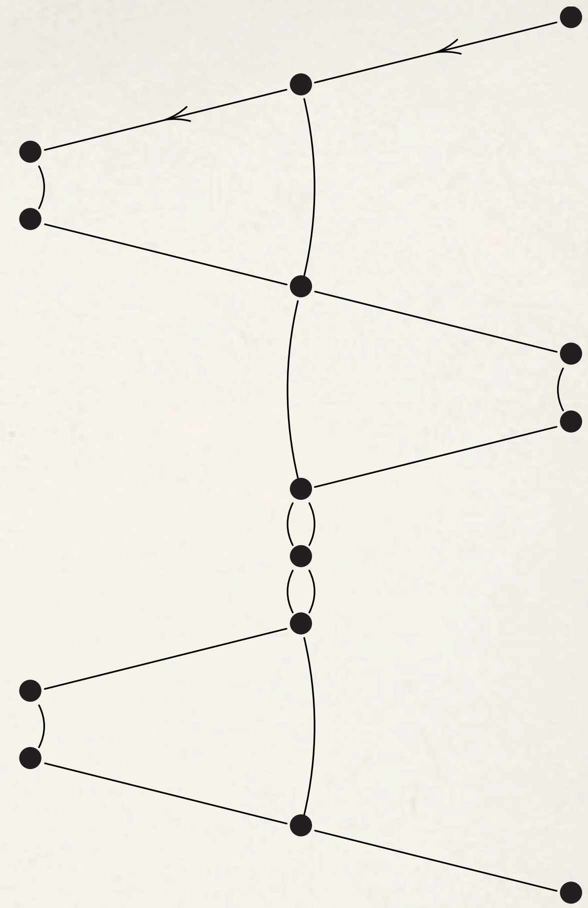


A graphical foundation for schedules

Guy McCusker — John Power — Cai Wingfield

University Of Bath

Bath–Swansea MathFound Seminar — March 22, 2012

Talk overview

- Schedules from Harmer et al.
- Joyal and Street’s framework
- Graphical definition
- Category of schedules

Schedules

Harmer, Hyland and Melliès, 2007
Categorical combinatorics for innocent strategies

- Paper constructs categories of games
- Describes some game semantic concepts via a distributive law
- Introduced notion of a $\multimap$-*scheduling function*.
  - Describes interleaving of plays in a game $A \multimap B$.

Schedules

Harmer, Hyland and Melliès, 2007
Categorical combinatorics for innocent strategies

- A **$\multimap$-scheduling function** is a function
  $$e : \{1, \ldots, n\} \to \{0, 1\}$$
  such that:
  $$e(1) = 1$$
  $$e(2k+1) = e(2k)$$
- I.e., binary strings
  - e.g. 1 00 11 11 00 1
  - e.g. 1 11 11 00 00 1
  - e.g. Prefixes

Schedules

Harmer, Hyland and Melliès, 2007
Categorical combinatorics for innocent strategies

- Schedules are pairs of embeddings

$$e_L : \{1, \ldots, |e|_0\} \hookrightarrow \{1, \ldots, |e|_0 + |e|_1\}$$
$$e_R : \{1, \ldots, |e|_1\} \hookrightarrow \{1, \ldots, |e|_0 + |e|_1\}$$

- Schedules are order relations

$$e_L(x) < e_R(y) \quad \subset \quad \{1, \ldots, |e|_0\}^{\text{even}} \times \{1, \ldots, |e|_1\}^{\text{even}}$$
$$e_R(y) < e_L(x) \quad \subset \quad \{1, \ldots, |e|_1\}^{\text{odd}} \times \{1, \ldots, |e|_0\}^{\text{odd}}$$

- **Compose** schedules by:
  - Composing corresponding order relations
  - Reconstructing function from composite

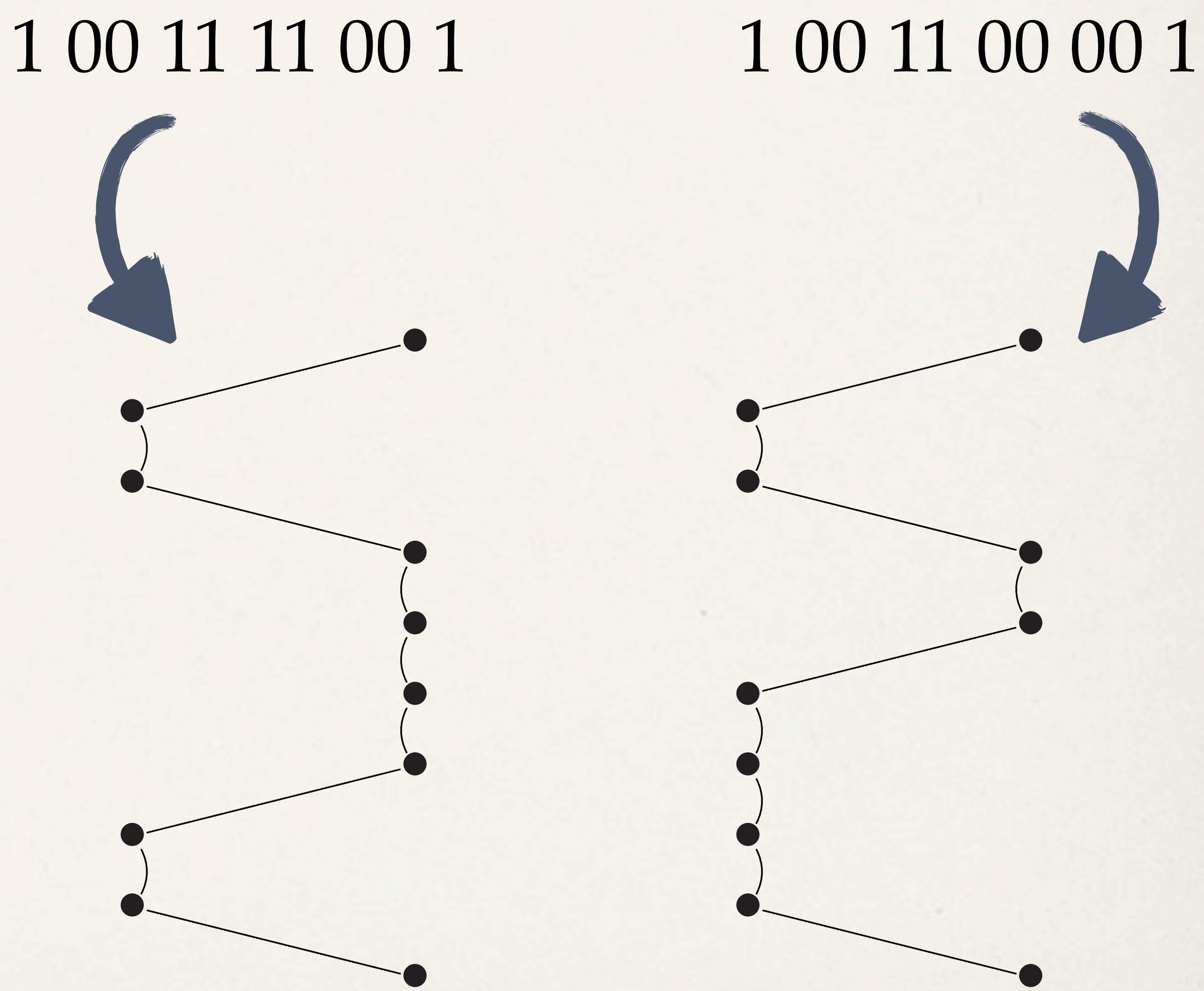
Schedules

Harmer, Hyland and Melliès, 2007
Categorical combinatorics for innocent strategies

- Composition is associative
  - Hard to prove!
- *Copycat* identities: prefixes of 1 00 11 00 11 00 11...
- Positive natural numbers and schedules form a category, Υ
  - Composition and identities are key

Schedules: what people do

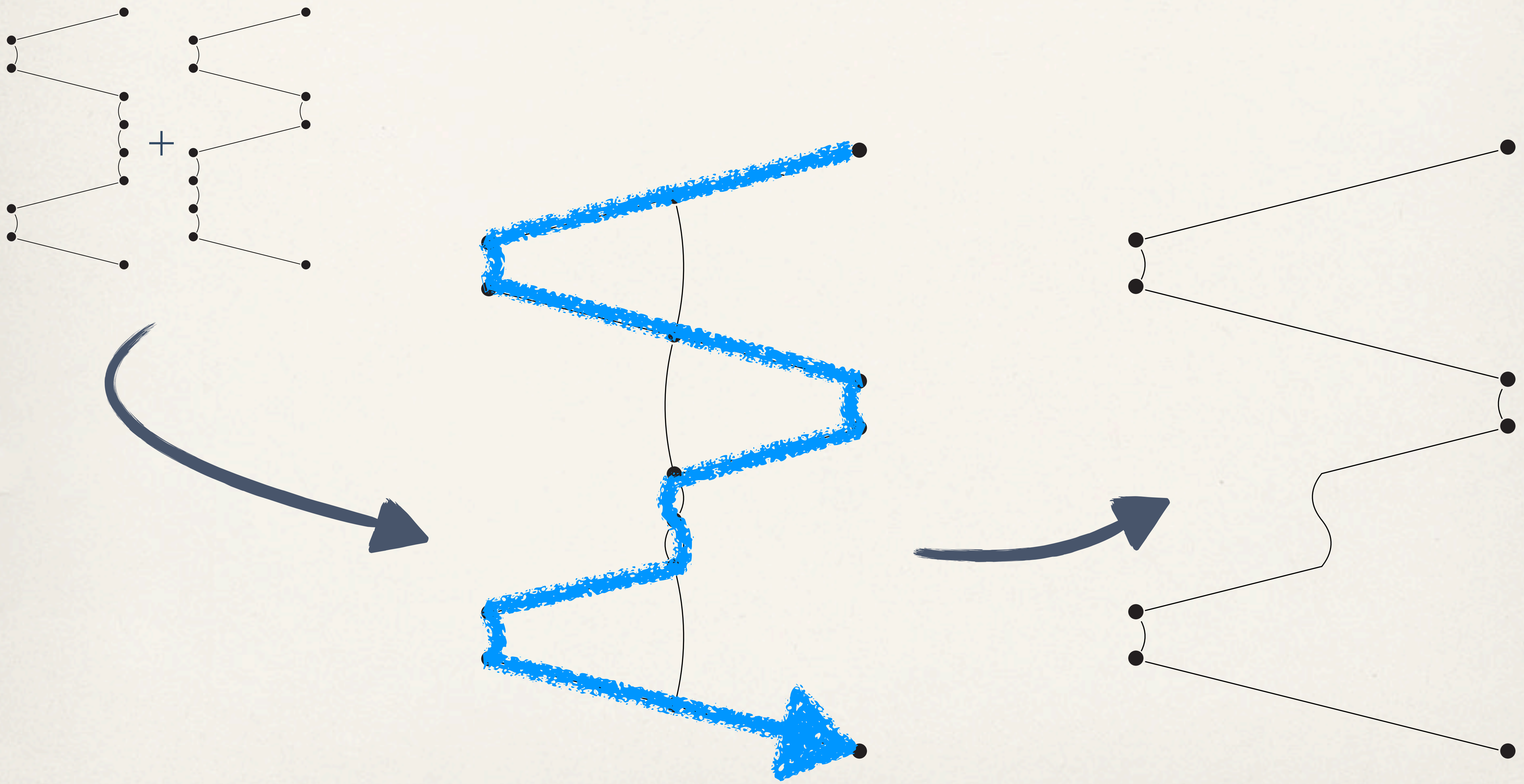
- Composing schedules via original definition is hard
- Use a graphical aid: *schedule diagrams*



Schedules: what people do

- Composition is a graphical exercise
  - Write schedules next to each other with nodes identified
  - Trace path “with momentum”

Schedules: what people do



Schedules: what people do

- Can we capture this and make it formal?
- Composition is easier...
  - ...can it help with schedules' other tricky properties?

PPGs

Joyal and Street, 1991
The geometry of tensor calculus I

- Set schedule diagrams in a general framework for diagrams
- Joyal and Street's *progressive plane graphs* for monoidal category string diagrams
  - Resembles what people draw
  - Operations on schedules are operations on PPGs
  - Compactness keeps things finite

PPGs

Joyal and Street, 1991
The geometry of tensor calculus I

- A **progressive plane graph** is a *progressive graph* which is *embedded* in the plane

$$\Gamma = (G, G_0)$$

- Hausdorff
- Edges are directed
- No cycles

- **Nodes**
- Separates graph into edges

$$\iota : \hat{\Gamma} \hookrightarrow \mathbb{R}^2$$

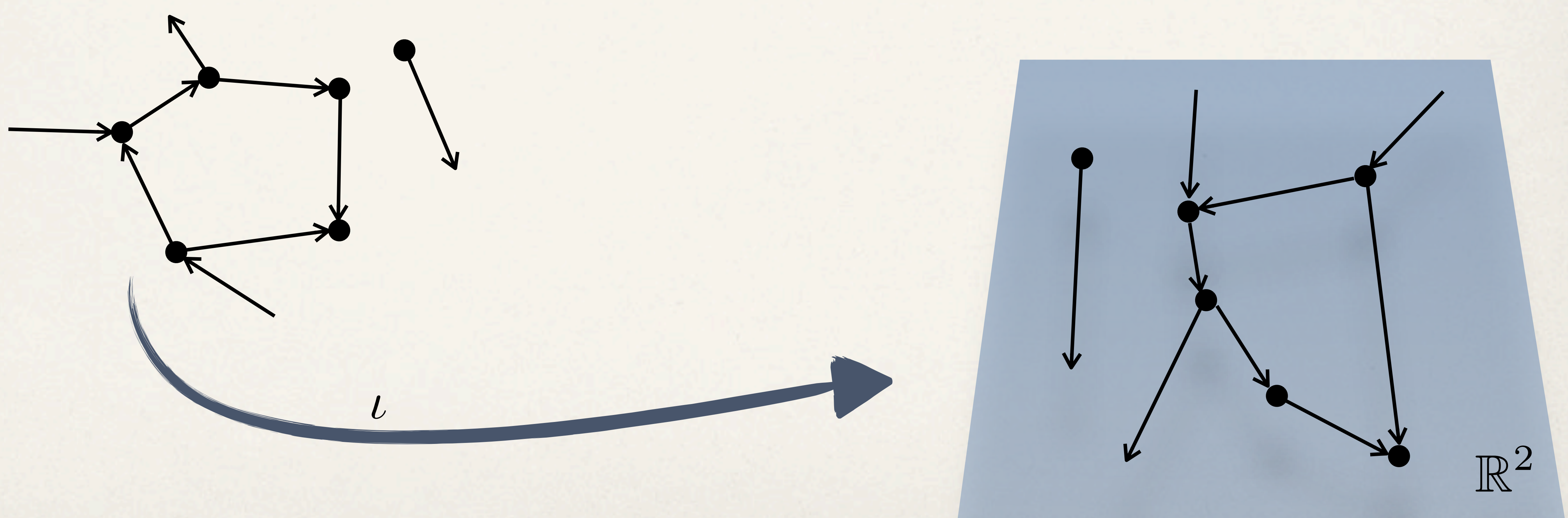
- Endpoint compactification
- Continuous injection
- $\longleftrightarrow$ projection injective on each edge
- Respects edge direction

PPGs

Joyal and Street, 1991
The geometry of tensor calculus I

- A **progressive plane graph** is a *progressive graph* which is *embedded* in the plane

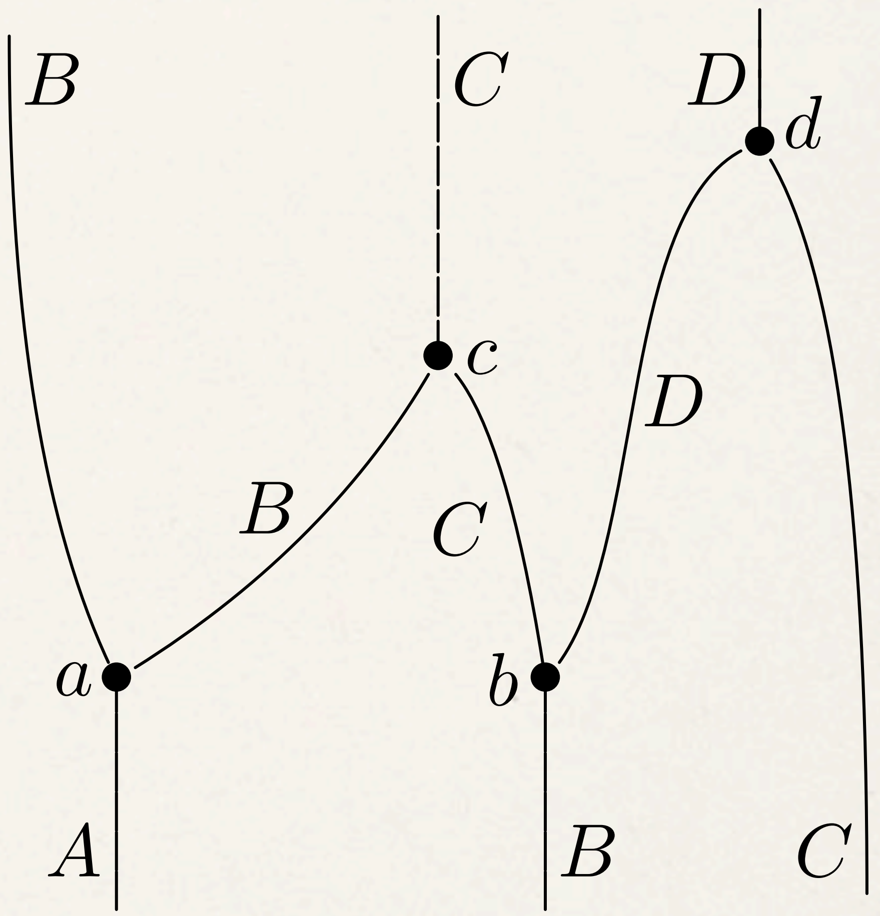
$$\Gamma = (G, G_0) \qquad \iota : \hat{\Gamma} \hookrightarrow \mathbb{R}^2$$



String diagrams

Joyal and Street, 1991
The geometry of tensor calculus I

- Example of how this is used elsewhere:
  - String diagrams for monoidal categories
  - PPGs have natural structure of free monoidal categories
  - Can be used to prove properties of monoidal structures



$$\|$$

$$(a \otimes b \otimes C) \circ (B \otimes c \otimes D \otimes C) \circ (B \otimes C \otimes d)$$

Schedules

- A **schedule**:

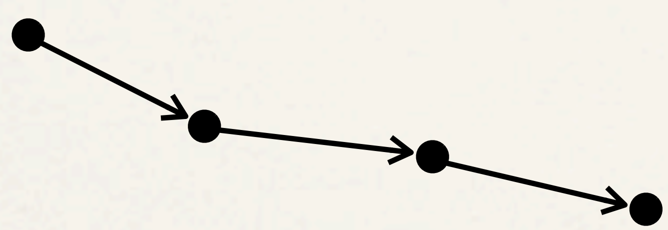
$$U = \{u_1, \ldots, u_m\}$$

$$V = \{v_1, \ldots, v_n\}$$

$$S_{m,n} = (U, V, \Sigma, \iota)$$

$$\Sigma = (S, U + V)$$

...



$$\Sigma = (S, P)$$

$$p_1 = v_1$$

$$\{p_{2k}, p_{2k+1}\} \subset U$$

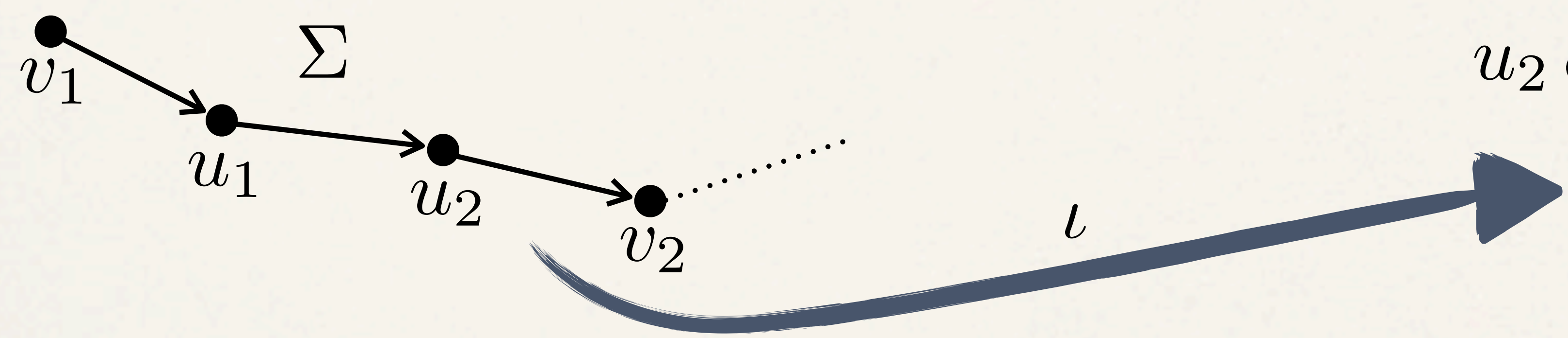
or

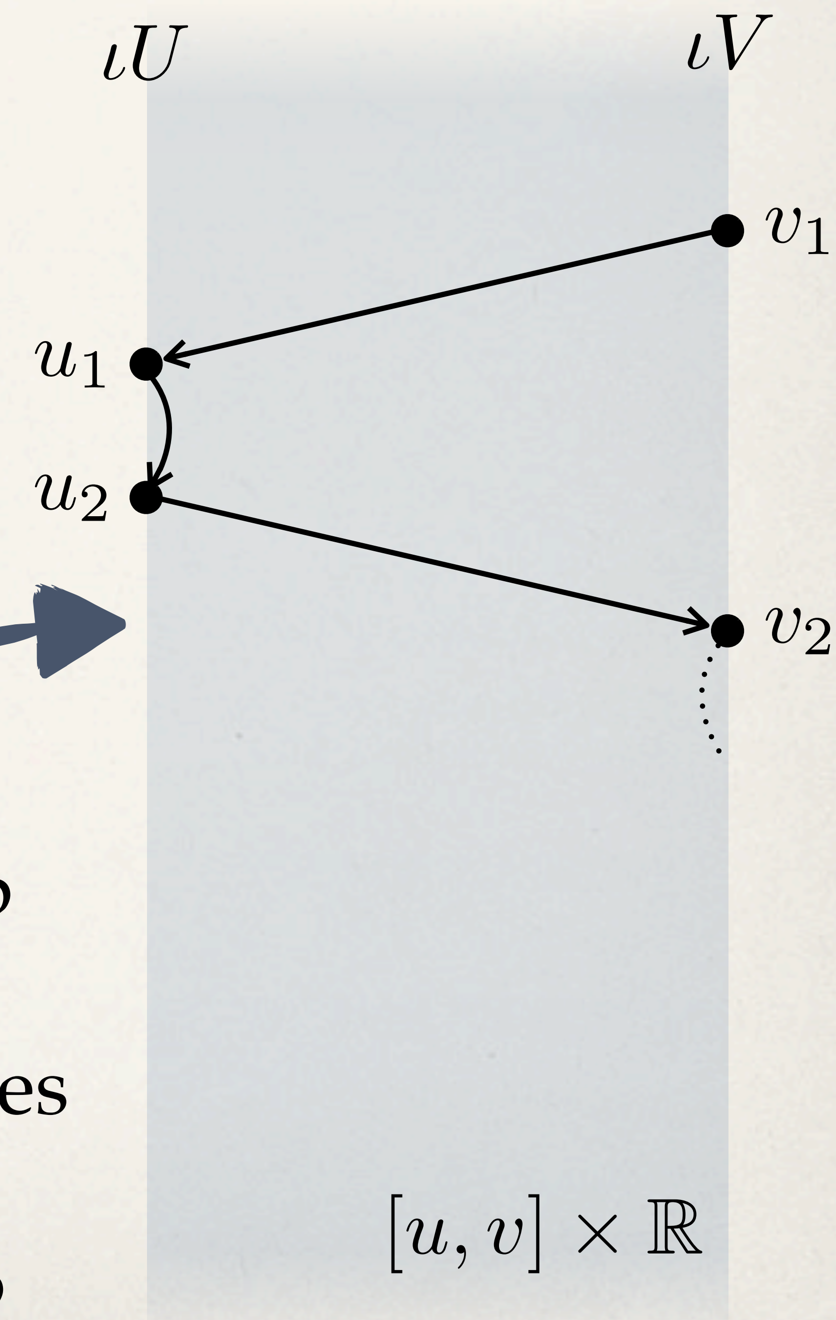
$$\{p_{2k}, p_{2k+1}\} \subset V$$

Schedules

- A **schedule**:

$$S_{m,n} = (U, V, \Sigma, \iota)$$

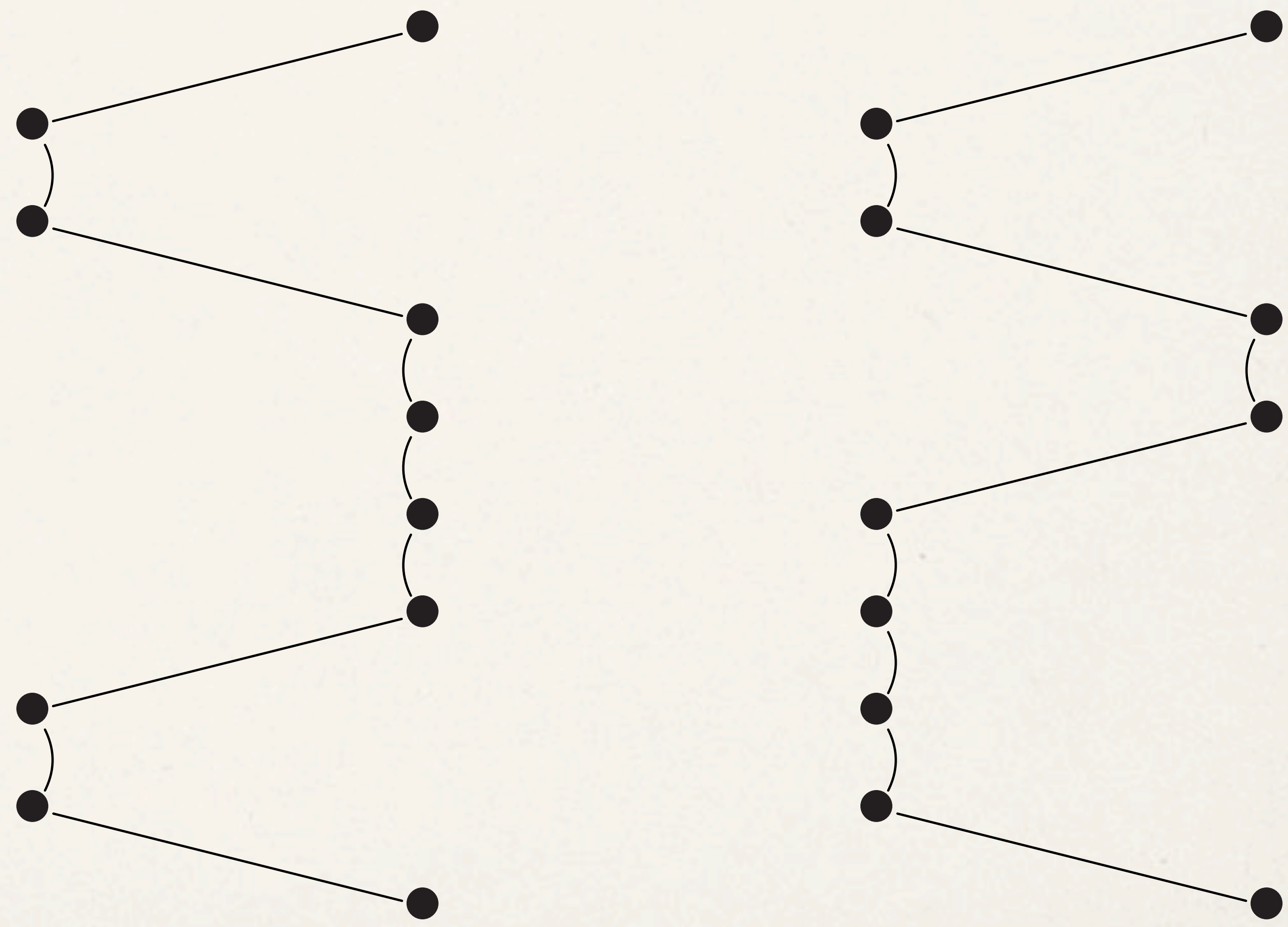




- Nodes into boundary of strip
- Downwards ordering of nodes
- Edges within interior of strip

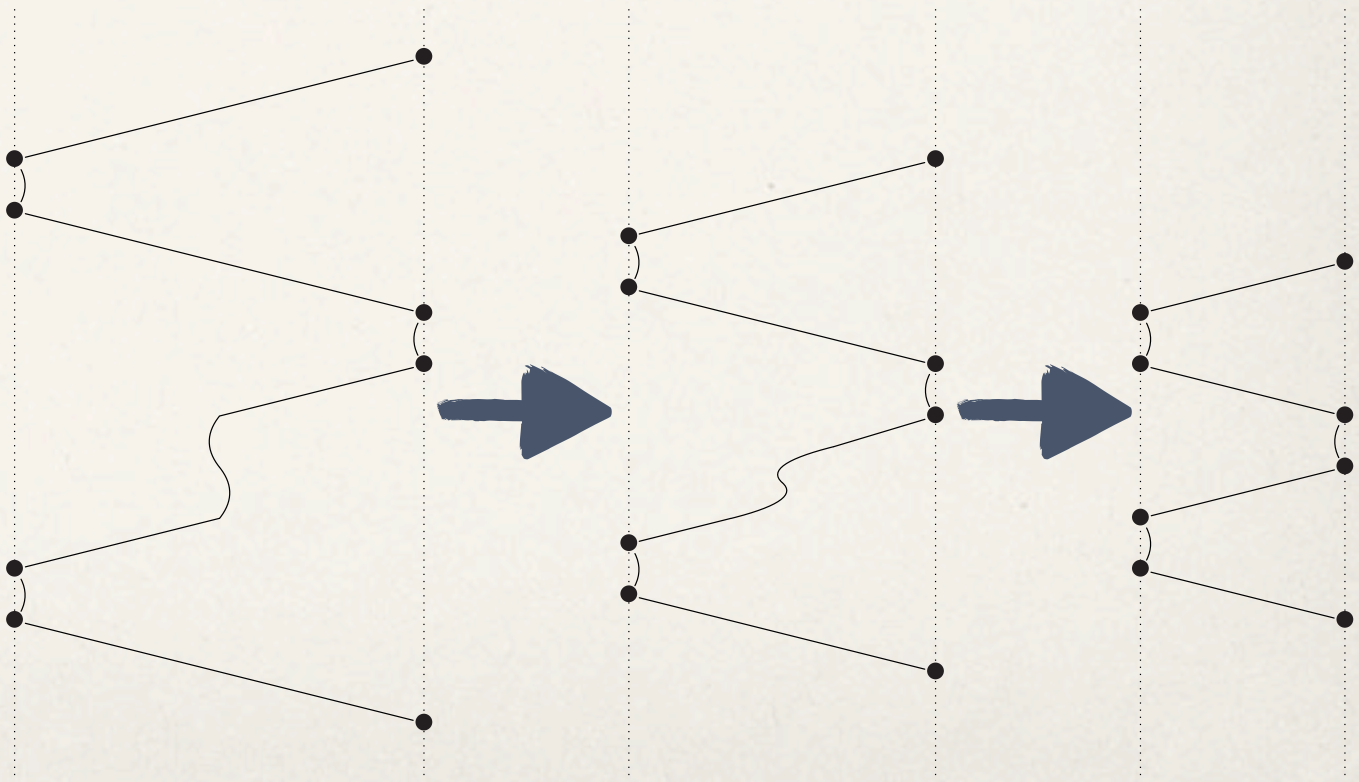
Schedules

- Examples:



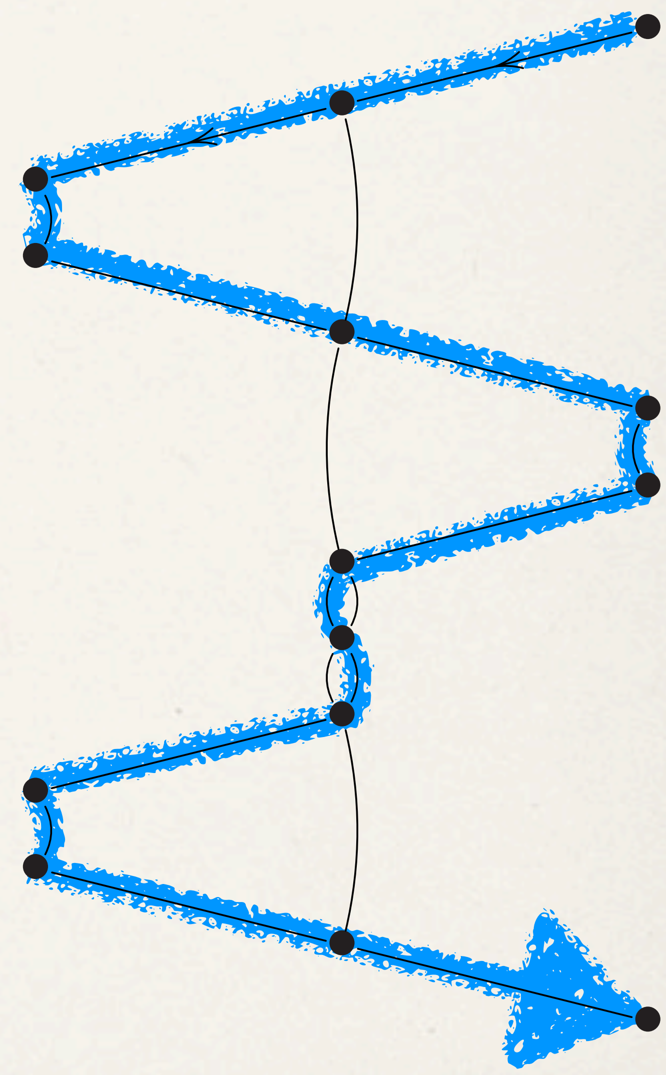
Deforming schedules

- When are two schedules "the same"?
- Consider equality of schedules to be *up to deformation*, such as:
  - Translation
  - "Piecewise" scaling
  - "Yanking" of zig-zags



Composition of schedules

- Capturing idea of "momentum"
- Two ways to think about it
  - (*Definition*) Algorithmically / inductively
    - Start top–right
    - Swap through internal nodes
    - Remove unpicked edges, internal nodes
  - (*Lemma*) Unique (up to deformation) path through all nodes

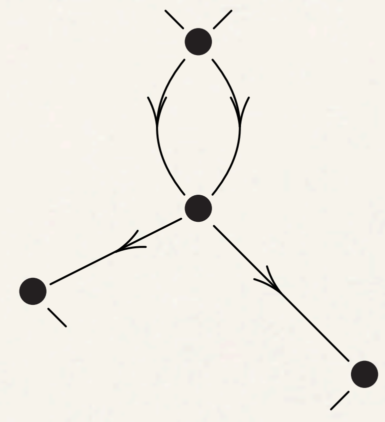


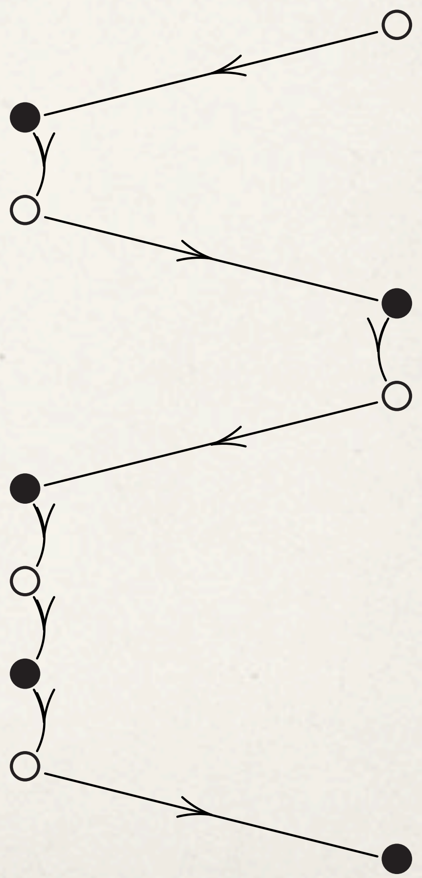
Composition of schedules

- Is this well-defined?
- (*Proposition*) Following this procedure produces a graph satisfying schedule conditions.
  - Removing internal nodes concatenates sequences of nodes on one side or the other
  - This preserves odd/evenness

Composition of schedules

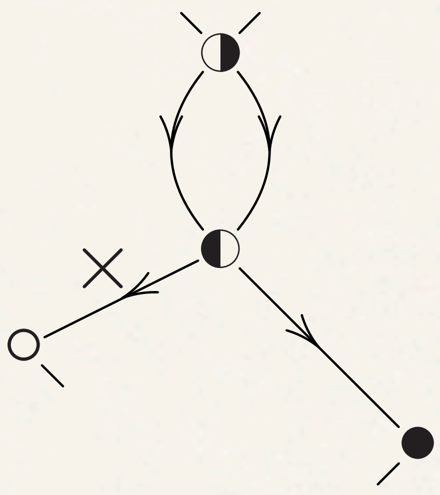
- Why can't we have these problematic scenarios?
  - Colour nodes ○/● (like O/P)
    - First right-hand node: ○
    - Nodes alternate ○/● along path
    - Nodes alternate ○/● down each side

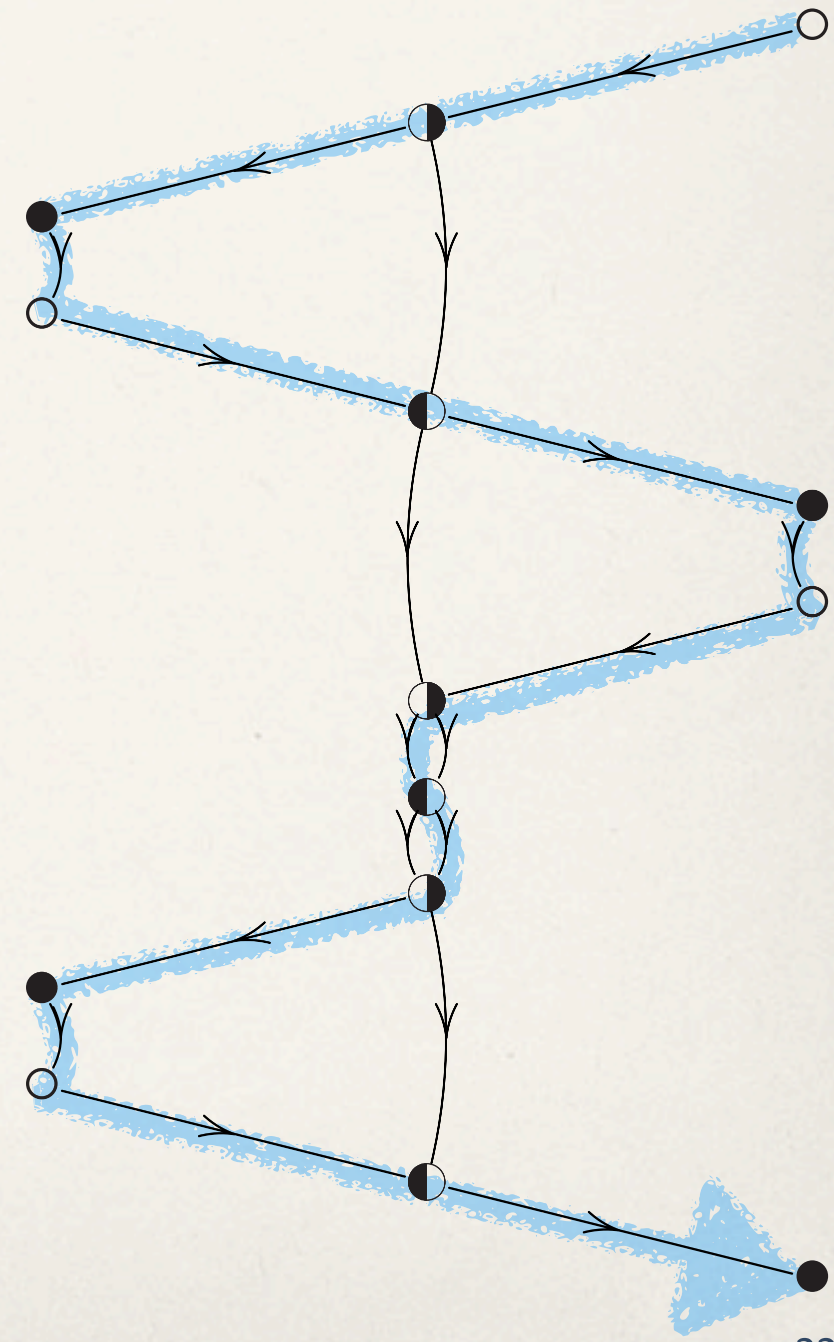




Composition of schedules

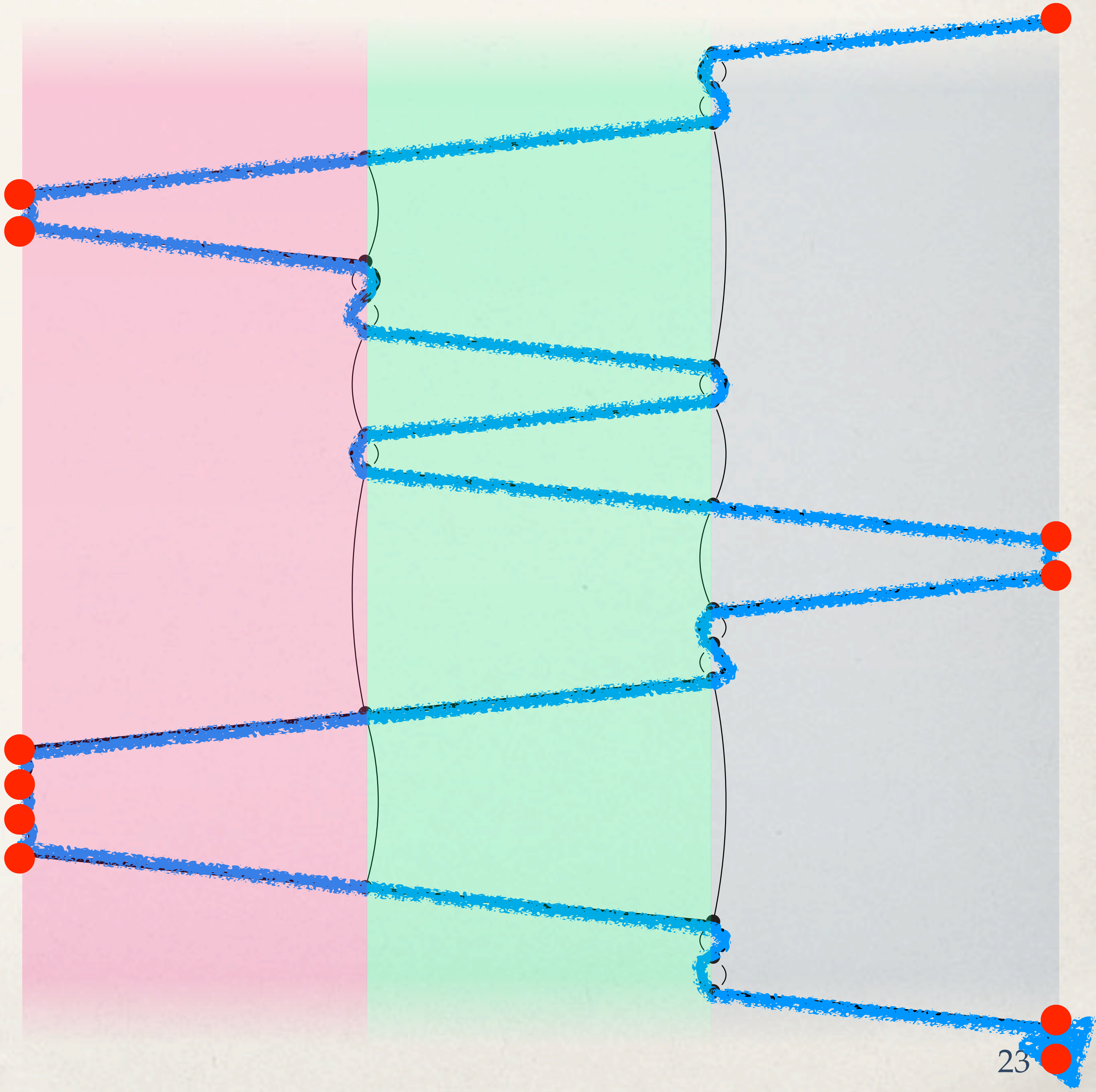
- Local encoding of global properties
- Internal nodes are two-coloured: ◑ or ◐
- Cross-schedule edges are ○→●
  - Problematic scenario is impossible
- When composing, remove ◑ or ◐





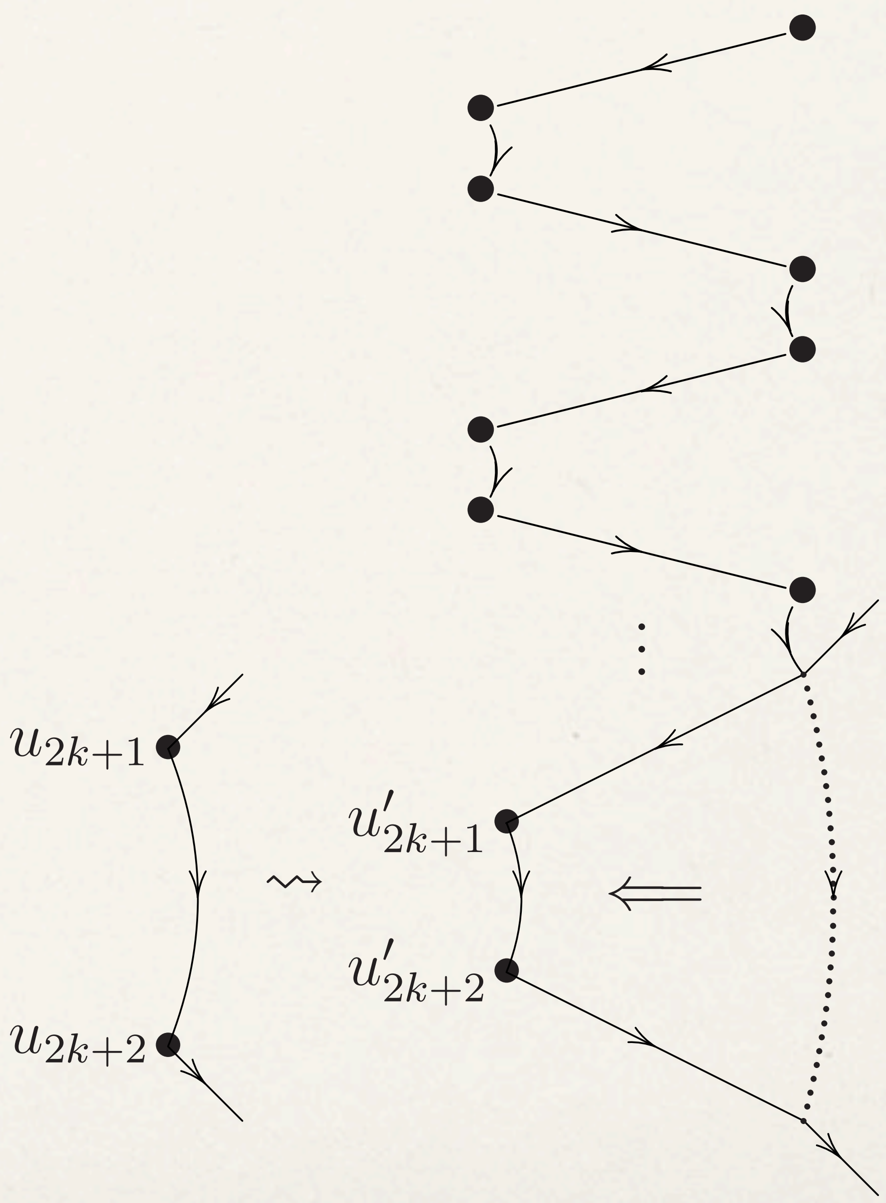
Composition of schedules

- (*Proposition*) Associativity is easy!
  - Write down three schedules
  - Composite is unique path through each node
  - Associating left/right is just discarding left/right set of unused edges and nodes first
- "Juxtaposition is associative"



Category of schedules

- (*Lemma*) *Copycat* schedules are identities
- (*Theorem*) Positive naturals and graphical schedules form a category, *Sched*.



Equivalence of categories

- (*Theorem*) Sched and Υ are equivalent as categories
  - Functor $C : \mathit{Sched} \longrightarrow \Upsilon$ is:
    - Identity on objects
    - Schedule $\longmapsto$ binary string recording left–right position
    - Composition is preserved
      - "Glueing cross-schedule edges is composing order relations on odd and even subsets"

Equivalence of categories

- Functor $G : \Upsilon \longrightarrow \mathit{Sched}$ is:
  - Identity on objects
  - Binary string $\longmapsto$ some canonical schedule construction
    - E.g. nodes at integer heights, edges are straight lines and circular arcs

Equivalence of categories

- $CG = \mathrm{id}$
- $GC \cong \mathrm{id}$
  - Schedules determined up to deformation by left–right position of nodes
    - Arrange nodes in order with unit vertical distances
    - Compact, simply-connected rectangles with nodes in corners
    - Endpoint-preserving homotopies relate any edges within a rectangle

Results

- Definitions relate directly to pictures and practice amongst researchers
- Demonstration of key properties rendered far simpler through careful definitions
- Relation to other work:
  - Schedules can also be characterised using the free adjunction $\mathcal{Adj}$
  - Cf. Melliès' 2-categorical string diagrams for adjunctions (in preparation)

Future work

- Other constructions from Harmer et al.
  - $\otimes$-scheduling functions.
  - Strategies
  - Pointer functions and heaps

Future work

- Definition of associative composition for more relaxed notions of scheduling
  - Our schedules are typed by numbers
  - Alternative notions of type may support broader classes of schedule

Future work

- Joyal and Street's framework can be expanded for other classes of diagram
- Hopefully our use of it will:
  - Provide common ground for future work
  - Contribute new categories of games and strategies