Asynchronous Template Games
and the Gray Tensor Product of 2-Categories

Paul-André Melliès

Institut de Recherche en Informatique Fondamentale (IRIF)
CNRS & Université de Paris

ACM/IEEE LICS 2021  29 June → 02 July 2021
Main purpose

We want to combine two styles of game semantics:

- **template games** played on **categories** of positions and trajectories,
- **asynchronous games** played on **graphs** with **permutation tiles** indicating when two moves $m$ and $n$ are **independent** in the game.
Asynchronous graphs

An asynchronous graph is defined as a graph

$$G = (V, E)$$

equipped with a set of permutation tiles of the form

between coinital and cofinal paths of length 2.
— Axiom 1 —
All permutations are symmetric
— Axiom 2 —
All permutations are deterministic

\[ n' \cdot m' = n'' \cdot m'' \]
— Axiom 3 —

The cube axiom
The shuffle tensor product

The **shuffle tensor product**

\[ G \uplus H = (G \uplus H, \diamond_{G \uplus H}) \]

of two asynchronous graphs

\[ G = (G, \diamond_G) \quad \quad H = (H, \diamond_H) \]

is the asynchronous graph

- whose vertices \((x, y)\) are the pairs of vertices \(x \in G\) and \(y \in H\),
The shuffle tensor product

whose edges are of two kinds: the pairs

\[ (x, y) \xrightarrow{(u,y)} (x', y) \]

consisting of an edge in the graph \( G \)

\[ x \xrightarrow{u} x' \]

and of a vertex \( y \in H \); and pairs

\[ (x, y) \xrightarrow{(x,v)} (x, y') \]

consisting of an edge in the graph \( H \)

\[ y \xrightarrow{v} y' \]

and of a vertex \( x \in G \).
The shuffle tensor product

whoose permutation tiles are of three kinds:

1. two permutation tiles

for every pair of edges

\[ x \xrightarrow{u} x' \quad \text{and} \quad y \xrightarrow{v} y' \]

in the graphs \( G \) and \( H \) respectively;
The shuffle tensor product

2. a permutation tile

for every permutation tile

in the asynchronous graph $G$ and every vertex $y \in H$ ;
The shuffle tensor product

3. a permutation tile

for every permutation tile

in the asynchronous graph $H$ and every vertex $x \in G$. 
The category of asynchronous graphs

The category \textbf{Asynch} of asynchronous graphs has its morphisms

\[
f : (G, \diamond_G) \rightarrow (H, \diamond_H)
\]

graph homomorphisms

\[
f : G \rightarrow H
\]

transporting every permutation tile of \( G \) to a permutation tile of \( H \).

\textbf{Theorem.} The shuffle tensor product

\[
G, H \mapsto G \sqcup H : \text{Asynch} \times \text{Asynch} \rightarrow \text{Asynch}
\]

turns the category \textbf{Asynch} into a \textit{symmetric monoidal category}.
Basic illustration

For every label \( \text{token} \), we define the asynchronous graph

\[
\pm[\text{token}]
\]

with a unique vertex \( * \) and a unique edge

\[
\text{token} : * \rightarrow *
\]

together with a unique permutation tile
The template of games

The template of games \( \pm_{\text{game}} \) is the asynchronous graph

\[
\pm[O, P] = \pm[O] \uplus \pm[P]
\]

with one unique vertex \(*\) and two edges

\[
O, P : * \rightarrow *
\]

together with four permutation tiles expressing that all edges in \( \pm[O, P] \) are pairwise independent.
Asynchronous games

By definition, an asynchronous game

$$(A, \lambda_A)$$

is an asynchronous graph equipped with a polarity map

$$\lambda_A : (A, \diamond_A) \rightarrow \pm\text{game}$$
What this means...

The polarity map assigns a polarity $O$ or $P$ to each edge of the graph:

- an **Opponent move** $m : x \to y$ is an edge mapped to the polarity $O$,

- a **Player move** $m : x \to y$ is an edge mapped to the polarity $P$.

Polarities of moves are moreover preserved by permutation tiles:

\[
\lambda_A(m) = \lambda_A(m') \quad \text{and} \quad \lambda_A(n) = \lambda_A(n').
\]
Asynchronous games as 2-categories

At this stage, we revisit one main principle of template games:

think of games as categories with polarities

which we lift one dimension up to the following mantra:

think of asynchronous games as 2-categories with polarities
Asynchronous games as 2-categories

To that purpose, we use the basic observation that every asynchronous graph \((G, \diamond_G)\) generates a 2-category \(\langle G, \diamond_G \rangle\).

The 2-category \(\langle G, \diamond_G \rangle\) is defined in the following way:

- its objects = the vertices of the graph,
- its morphisms = the paths of the graph,
- its 2-cells = the reshufflings induced by the permutation tiles.
Reshufflings between paths

Definition: a reshuffling is a bijective function

$$\varphi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$$

which "keeps track" of a sequence of tiles on a path of length $n$.

Typically, the reshuffling $$\begin{pmatrix} 1 & \leftrightarrow & 2 \\ 2 & \leftrightarrow & 1 \end{pmatrix}$$ is associated to any permutation tile:
Reshufflings between paths

Similarly, the reshuffling on three indices

\[
\begin{pmatrix}
1 & \mapsto & 3 \\
2 & \mapsto & 2 \\
3 & \mapsto & 1
\end{pmatrix}
: \quad \{1, 2, 3\} \quad \rightarrow \quad \{1, 2, 3\}
\]

keeps track and identifies the two sequences of tiles:

Related to the braid equation and the Yang-Baxter equation
From asynchronous graphs to 2-categories, functorially...

The translation induces a functor

$$\langle - \rangle : \text{Asynch} \longrightarrow \text{TwoCat}$$

where \text{TwoCat} is the category of 2-categories and 2-functors.

Key observation:

The functor $\langle - \rangle$ comes equipped with a family of isomorphisms

$$\langle G \sqcup H \rangle \cong \langle G \rangle \boxtimes \langle H \rangle \quad \langle I \rangle \cong 1$$

and thus defines a symmetric monoidal functor

$$\langle - \rangle : (\text{Asynch}, \sqcup, I) \longrightarrow (\text{TwoCat}, \boxtimes, 1)$$

where we write $\boxtimes$ for the Gray tensor product of 2-categories.
Polarities and Gray comonoids

Here, something like an algebraic miracle occurs!

**Big surprise and main observation of the paper:**

**Fact.** A polarity structure on an asynchronous graph \((A, \diamond_A)\)

\[
\lambda_A : E_A \rightarrow \{O, P\}
\]

is the same thing as a **Gray comonoid structure**

\[
\text{comult} : \langle A, \diamond_A \rangle \rightarrow \langle A, \diamond_A \rangle \boxtimes \langle A, \diamond_A \rangle
\]

on the associated 2-category \(\langle A, \diamond_A \rangle\).
Illustration

Typically, the 2-category

\[ \triangledown \{ \text{token} \} = \langle \triangledown [\text{token}] \rangle \]

generated by the asynchronous graph

has exactly two Gray comonoid structures, noted \( \triangledown \{ O \} \) and \( \triangledown \{ P \} \).
The negative Gray comonoid

The comultiplication of the Gray comonoid

$$\text{comult}_O : \forall \{O\} \rightarrow \forall \{O_1\} \boxtimes \forall \{O_2\}$$

transports the unique edge $O$ to the path $O_2 \cdot O_1$ of length 2:

This is the scheduling of an Opponent move in a copycat strategy.
The positive Gray comonoid

The comultiplication of the Gray comonoid

\[ \text{comult}_P : \pm \{ P \} \rightarrow \pm \{ P_1 \} \boxtimes \pm \{ P_2 \} \]

transports the unique edge \( P \) to the path \( P_1 \cdot P_2 \) of length 2.

This is the scheduling of a Player move in a copycat strategy.
The mixed Gray comonoid

By tensoring the two Gray comonoids

$$\pm \{ O, P \} = \pm \{ O \} \boxtimes \pm \{ P \}$$

one obtains the Gray comonoid whose comultiplication

$$\text{comult}_{OP} : \pm \{ O, P \} \longrightarrow \pm \{ O_1, P_1 \} \boxtimes \pm \{ O_2, P_2 \}$$

reflects the "mixed" scheduling of the usual copycat strategy:

![Diagram showing the mixed Gray comonoid](image-url)
Asynchronous template games

This leads us to the following definition:

**Definition.** An **asynchronous template game** is a Gray comonoid

\[ \text{comult} : A \rightarrow A \otimes A \quad \text{counit} : A \rightarrow 1 \]

equipped with a **polarity map** of Gray comonoids

\[ \lambda_A : A \rightarrow \pm \text{game} \]

where the **template of games** is defined as

\[ \pm \text{game} = \pm \{ O, P \} \]
Asynchronous strategies

Definition. An asynchronous strategy

\[ \sigma = (S, \text{coact}_\sigma, \lambda_\sigma) : (A, \lambda_A) \rightarrow (B, \lambda_B) \]

is a triple consisting of

- a 2-category \( S \) called the support of the strategy,
- a bicomodule structure \( \text{coact}_\sigma : S \rightarrow A \square S \square B \)
- a scheduling 2-functor \( \lambda_\sigma : S \rightarrow \pm \text{strat} \)

where the template of strategies is defined as

\[ \pm \text{strat} = \pm \text{game} \square \pm \text{game} = \pm \{ O_s, P_s, O_t, P_t \} \]
The template of strategies

Here, each of the four labels

\[ O_s \quad P_s \quad O_t \quad P_t \]

describes a specific kind of Opponent and Player move

- \( O_s \): Opponent move played at the source game
- \( P_s \): Player move played at the source game
- \( O_t \): Opponent move played at the target game
- \( P_t \): Player move played at the target game

which may appear on the interactive trajectory played by a strategy

\[ \sigma : A \rightarrow B. \]
Asynchronous strategies

One requires moreover that the diagram commutes:

```
\[
\begin{array}{c}
S \xrightarrow{\text{coact}_\sigma} A \boxtimes S \boxtimes B \\
\downarrow \lambda_\sigma \\
\n\downarrow \text{coact}_\perp
\end{array}
\]

The diagram says that the **scheduling 2-functor**

\[
\lambda_\sigma : S \longrightarrow \Diamond \text{strat}
\]

is a **map of Gray comodules** to the Gray comodule

\[
\text{coact}_\perp : \Diamond \text{strat} \longrightarrow \Diamond \text{strat} \boxtimes \Diamond \text{game}
\]
Asynchronous strategies

Equivalently, an asynchronous strategy

$$\sigma = (S, \text{coact}_\sigma, \lambda_\sigma) : (A, \lambda_A) \rightarrow (B, \lambda_B)$$

is a double cell

in the double category $\text{BiComod}$ of $\text{Gray comonoids}$ and $\text{bicomodules}$. 
The template of interactions

Key observation: the template of interactions

\[ \pm \text{int} = \left( \pm \text{game} \xrightarrow{\pm \text{strat}} \pm \text{game} \xrightarrow{\pm \text{strat}} \pm \text{game} \right) \]

coincides with the 2-category

\[ \pm \text{int} = \pm \{ O_s, P_s, OP, PO, O_t, P_t \} \]

generated by the six polarities of moves on three games:

- \( O_s \): Opponent move played at the source game
- \( P_s \): Player move played at the source game
- \( OP \): internal move played at the intermediate game
- \( PO \): internal move played at the intermediate game
- \( O_t \): Opponent move played at the target game
- \( P_t \): Player move played at the target game
The template of interactions

The template of interactions comes equipped with a 2-functor

\[ \text{hide} : \dashv \text{int} \rightarrow \dashv \text{strat} \]

which hides the internal polarities \( PO \) and \( OP \).

This defines a double cell:

\[
\begin{array}{ccc}
\dashv \text{game} & \dashv \text{strat} & \dashv \text{game} \\
\downarrow id & \downarrow \text{hide} & \downarrow id \\
\dashv \text{game} & \dashv \text{strat} & \dashv \text{game}
\end{array}
\]

in the double category \( \text{BiComod} \).
Composition of strategies

Now, suppose given a pair of asynchronous strategies

\[ \sigma = (S, \text{coact}_\sigma, \lambda_\sigma) : (A, \lambda_A) \rightarrow (B, \lambda_B) \]
\[ \tau = (T, \text{coact}_\tau, \lambda_\tau) : (B, \lambda_B) \rightarrow (C, \lambda_C) \]

represented as a pair of double cells:
Composition of strategies

The composite

$$\tau \circ \sigma = (S \boxtimes_B T, \text{coact}_{\tau \circ \sigma}, \lambda_{\tau \circ \sigma}) : (A, \lambda_A) \rightarrow (C, \lambda_C)$$

is obtained by composing the two cells horizontally:

and then composing vertically with the double cell hide.
Composition of strategies

\[
\begin{align*}
A & \quad S & \quad A \quad B & \quad T & \quad B \quad C \\
\lambda_A & \quad \lambda_\sigma & \quad \lambda_B & \quad \lambda_\tau & \quad \lambda_C \\
\pm \text{game} & \quad \pm \text{strat} & \quad \pm \text{game} & \quad \pm \text{strat} & \quad \pm \text{game} \\
\text{id} & \quad \text{hide} & \quad \text{id} & \quad \text{id} & \quad \text{id} \\
\pm \text{game} & \quad \pm \text{strat} & \quad \pm \text{game} & \quad \pm \text{strat} & \quad \pm \text{game}
\end{align*}
\]
Composition of strategies

This definition of composition implements the slogan that

\[
\text{composition} = \text{synchronization} + \text{hiding}
\]
Illustration

A nice diagrammatic way to represent a strategy

$$\sigma = (S, \text{coact}_\sigma, \lambda_\sigma) : (A, \lambda_A) \rightarrow (B, \lambda_B)$$

and its comodule structure

$$S \xrightarrow{\text{coact}_\sigma} A \Box S \Box B$$

is to draw them as follows:
Illustration

We may also consider an asynchronous strategy

\[ \tau = (T, \text{coact}_\tau, \lambda_\tau) : (B, \lambda_B) \rightarrow (C, \lambda_C) \]

and its comodule structure represented as
Illustration

The tensor of Gray comodules

\[ S \boxtimes B \triangleright \\sigma \boxtimes T \boxleftarrow \\tau S \boxtimes B \boxtimes T \]

may be understood as a synchronization scheme between strategies:
Illustration

In this diagrammatic representation, the equalizer

\[ S \boxtimes_B T \quad \xrightarrow{\text{equ}} \quad S \boxtimes T \]

is depicted as follows:

where the synchronized OP move \( m \) is mapped to the path \( n_1 \cdot m_2 \).
Illustration

In this diagrammatic representation, the equalizer

\[ S \boxtimes_B T \xrightarrow{\text{equ}} S \boxtimes T \]

is depicted as follows:

where the synchronized \( PO \) move \( n \) is mapped to the path \( n_2 \cdot m_3 \)
What about the identities?

The templates \( \#$game \) and \( \#$strat \) are related by the 2-functor

\[
\text{copycat} : \#$game \rightarrow \#$game \boxtimes \#$game
\]

defined by the comultiplication of the Gray comonoid \( \#$game \).

This defines a double cell

in the double category \( \text{BiComod} \).
The identity strategy
What makes everything work...

The family of 2-categories

\[ \ast [0] = \ast \text{game} \quad \ast [1] = \ast \text{strat} \quad \ast [2] = \ast \text{int} \]

defines an \textbf{internal category} \( \ast \text{asynch} \) in the monoidal category

\( (\text{TwoCat}, \boxtimes, 1) \)

of 2-categories equipped with the Gray tensor product.

An important point: we use here the definition by Marcelo Aguiar of

\textbf{an internal category in a monoidal category}

because the Gray tensor product is not the cartesian product of \textbf{TwoCat}. 

What makes everything work...

In other words, the horizontal map

$$\dagger \text{strat} : \dagger \text{game} \longrightarrow \dagger \text{game}$$

defines a **formal monad** $\dagger_{\text{asynch}}$ with multiplication and unit

in the double category $\textbf{BiComod}$ of **Gray comonoids** and **bicomodules**.
As an immediate consequence...

**Theorem A.** The construction just given defines a bicategory \( \text{Games}(\pm_{\text{asynch}}) \) of asynchronous games, strategies and simulations.
Main technical result of the paper

**Theorem B.** The bicategory of asynchronous games

\[ \text{Games}(\pm_{\text{asynch}}) \]

is symmetric monoidal closed.
Main technical result of the paper

**Theorem C.** The bicategory of asynchronous games 

\[ \text{Games}(\pm_{\text{asynch}}) \]

is **star-autonomous**.
Conclusion and perspectives

- asynchronous games played on 2-categories with
  - positions of the game as objects
  - trajectories as morphisms
  - reshufflings as 2-cells
- the Gray tensor of 2-categories as a shuffle tensor product
- an unexpected connection with bicomodules on Gray comonoids
- a model of differential linear logic based on homotopy
Thank you and stay safe!