An Asynchronous Game Semantics for Linear Logic

Samuel Mimram

CIE'10

July 2nd 2010

A program is a text in a programming language which will evolve during time.

We have to give a meaning to this language!

Denotational semantics

A **model** interprets

- a type A as a *computation space* $[\![A]\!]$
- a program $f : A \Rightarrow B$ as a *transformation* $[\![f]\!] : [\![A]\!] \to [\![B]\!]$

Denotational semantics

A **denotational model** interprets

- a type A as a *computation space* $[\![A]\!]$
- a program $f : A \Rightarrow B$ as a *transformation* $[\![f]\!] : [\![A]\!] \to [\![B]\!]$
- in a way such that the interpretation of programs is invariant under reduction

$$\boxed{\text{denotational semantics} \quad = \quad \text{program invariants}}$$

Interactive semantics

Here, a program will be modeled by its
interactive behavior
i.e. by the way it reacts to information provided by its
environment.

$$\begin{array}{rcl} (\texttt{fun } x \to \texttt{not } x)\texttt{false} & \rightsquigarrow & \texttt{true} \\ (\texttt{fun } x \to \texttt{not } x)\texttt{true} & \rightsquigarrow & \texttt{false} \end{array}$$

$\Rightarrow$ Game Semantics!

How can we extend game semantics to concurrent languages?

Game semantics

An *interactive trace semantics*:

- types are interpreted by **games**
- programs are interpreted by **strategies**

Game semantics

An *interactive trace semantics*:

- types are interpreted by **games**
  - a poset $(M, \leq)$ of *moves*
  - a *polarization function* $\lambda : M \to \{O, P\}$
- programs are interpreted by **strategies**

Game semantics

An *interactive trace semantics*:

- types are interpreted by **games**
  - a poset $(M, \leq)$ of *moves*
  - a *polarization function* $\lambda : M \to \{O, P\}$
- a **play** is a sequence $m_1 \cdot m_2 \cdots m_k$ of moves which is
  - respecting order:
    all the moves below a given move m_i occur before m_i
  - alternating: $m_1 \cdot m_2 \cdot m_3 \cdot m_4 \cdots$
- programs are interpreted by **strategies**

Game semantics

An *interactive trace semantics*:

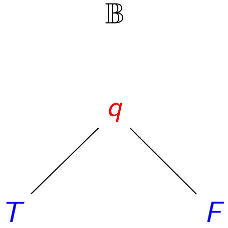
- types are interpreted by **games**
  - a poset $(M, \leq)$ of *moves*
  - a *polarization function* $\lambda : M \to \{O, P\}$

- a **play** is a sequence $m_1 \cdot m_2 \cdots m_k$ of moves which is
  - respecting order:
    all the moves below a given move m_i occur before m_i
  - alternating: $m_1 \cdot m_2 \cdot m_3 \cdot m_4 \cdots$

- programs are interpreted by **strategies**

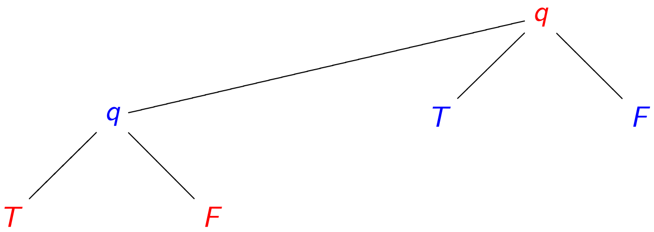
$$\text{strategy} \quad = \quad \text{set of plays closed under prefix}$$

Booleans



Booleans

$$\mathbb{B} \quad \Rightarrow \quad \mathbb{B}$$



The negation

The strategy interpreting negation $\texttt{not} : \mathbb{B} \Rightarrow \mathbb{B}$ is

$$[\![\texttt{not}]\!] \quad = \quad \{\, q \cdot q \cdot T \cdot F,\ q \cdot q \cdot F \cdot T,\ \ldots \,\}$$

$$\mathbb{B} \qquad \Rightarrow \qquad \mathbb{B}$$

The negation

The strategy interpreting negation $\texttt{not} : \mathbb{B} \Rightarrow \mathbb{B}$ is

$$[\![\texttt{not}]\!] \quad = \quad \{\, q \cdot q \cdot T \cdot F,\ q \cdot q \cdot F \cdot T,\ \ldots \,\}$$

$$\mathbb{B} \qquad \Rightarrow \qquad \mathbb{B}$$

$$\qquad\qquad\qquad\qquad q$$

The negation

The strategy interpreting negation $\texttt{not} : \mathbb{B} \Rightarrow \mathbb{B}$ is

$$[\![\texttt{not}]\!] \quad = \quad \{\, q \cdot q \cdot T \cdot F,\ q \cdot q \cdot F \cdot T,\ \ldots \,\}$$

$$\mathbb{B} \qquad \Rightarrow \qquad \mathbb{B}$$

$$\qquad\qquad\qquad\qquad q$$

$$q \qquad\qquad\qquad\qquad$$

The negation

The strategy interpreting negation $\texttt{not} : \mathbb{B} \Rightarrow \mathbb{B}$ is

$$[\![\texttt{not}]\!] \quad = \quad \{\, q \cdot q \cdot T \cdot F,\ q \cdot q \cdot F \cdot T,\ \ldots \,\}$$

$$\begin{array}{ccc} \mathbb{B} & \Rightarrow & \mathbb{B} \\ & & q \\ q & & \\ T & & \end{array}$$

The negation

The strategy interpreting negation $\texttt{not} : \mathbb{B} \Rightarrow \mathbb{B}$ is

$$[\![\texttt{not}]\!] \quad = \quad \{\, q \cdot q \cdot T \cdot F,\; q \cdot q \cdot F \cdot T,\; \ldots \,\}$$

$$\begin{array}{ccc} \mathbb{B} & \Rightarrow & \mathbb{B} \\ & & q \\ q & & \\ T & & \\ & & F \end{array}$$

The negation

The strategy interpreting negation $\texttt{not} : \mathbb{B} \Rightarrow \mathbb{B}$ is

$$[\![\texttt{not}]\!] \quad = \quad \{\, q \cdot q \cdot T \cdot F,\ q \cdot q \cdot F \cdot T,\ \ldots \,\}$$

$\mathbb{B}$	$\Rightarrow$	$\mathbb{B}$
		q
q		
F		
		T

A category of games and strategies

We can thus build a category whose

- objects A are *games*
- morphisms $\sigma : A \to B$ are *strategies*

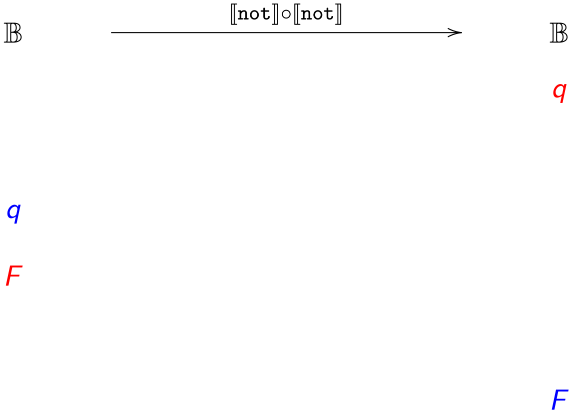
A category of games and strategies

For example, the composite $[\![\texttt{not}]\!] \circ [\![\texttt{not}]\!] : \mathbb{B} \to \mathbb{B}$ is

$\mathbb{B}$ $\xrightarrow{[\![\texttt{not}]\!]}$	$\mathbb{B}$	$\mathbb{B}$ $\xrightarrow{[\![\texttt{not}]\!]}$	$\mathbb{B}$
			q
	q	q	
q			
F			
	T	T	
			F

A category of games and strategies

For example, the composite $[\![\texttt{not}]\!] \circ [\![\texttt{not}]\!] : \mathbb{B} \to \mathbb{B}$ is



A category of games and strategies

For example, the composite $[\![\mathtt{not}]\!] \circ [\![\mathtt{not}]\!] : \mathbb{B} \to \mathbb{B}$ is

$$\mathbb{B} \xrightarrow{[\![\mathtt{not}]\!] \circ [\![\mathtt{not}]\!]} \mathbb{B}$$

$\mathbb{B}$	$\mathbb{B}$
	q
q	
T	
	T

A category of games and strategies

For example, the composite $[\![\mathtt{not}]\!] \circ [\![\mathtt{not}]\!] : \mathbb{B} \to \mathbb{B}$ is

$$\mathbb{B} \xrightarrow{[\![\mathtt{not}]\!] \circ [\![\mathtt{not}]\!]} \mathbb{B}$$

q (right)

q (left)

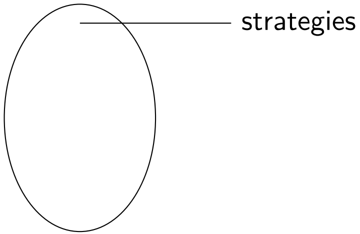
T (left)

T (right)

$$[\![\mathtt{not}]\!] \circ [\![\mathtt{not}]\!] \quad = \quad \{q \cdot q \cdot T \cdot T,\ q \cdot q \cdot F \cdot F,\ \cdots\} \quad = \quad [\![\mathrm{id}_{\mathbb{B}}]\!]$$

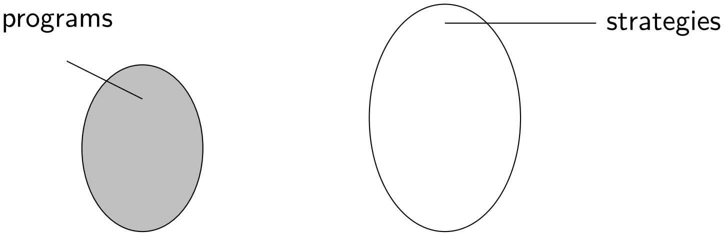
Definable strategies

We have to characterize **definable** strategies
(= strategies which are the interpretation of a program)



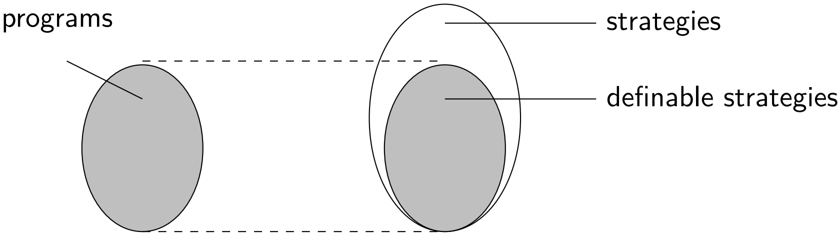
Definable strategies

We have to characterize **definable** strategies
(= strategies which are the interpretation of a program)



Definable strategies

We have to characterize **definable** strategies
(= strategies which are the interpretation of a program)



Definable strategies

We have to characterize **definable** strategies
(= strategies which are the interpretation of a program)

Two series of work laid the foundations of game semantics:

- fully abstract models of PCF [HON,AJM]
  definable strategies: bracketing and innocence conditions
  extended later on: references, control, non-determinism, . . .
- fully complete models of MLL [AJ,HO]

Purposes of game semantics

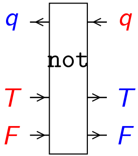
- Better understanding the core features of programming languages and logics

Purposes of game semantics

- Better understanding the core features of programming languages and logics
- Compositional model checking

Purposes of game semantics

- Better understanding the core features of programming languages and logics
- Compositional model checking
- Synthesis of electronic circuits



How do we extend those results
to *concurrent programming languages*?

Three flavors of conjunction

$\mathbb{B} \quad \times \quad \mathbb{B} \quad \Rightarrow \quad \mathbb{B}$

q

q_L

left conjunction T_L

q_R

F_R

F

Three flavors of conjunction

$$\mathbb{B} \quad \times \quad \mathbb{B} \quad \Rightarrow \quad \mathbb{B}$$

q

q_R

right conjunction F_R

q_L

T_L

F

Three flavors of conjunction

$\mathbb{B} \quad \times \quad \mathbb{B} \quad \Rightarrow \quad \mathbb{B}$

q

q_R

parallel conjunction $\quad q_L$

F_R

T_L

F

Three flavors of conjunction

$\mathbb{B} \quad \times \quad \mathbb{B} \quad \Rightarrow \quad \mathbb{B}$

q

q_L

parallel conjunction $\quad q_R$

T_L

F_R

F

Towards asynchronous game semantics

In order to represent such strategies we have to

- take in account **non-alternating plays**

Towards asynchronous game semantics

In order to represent such strategies we have to

- take in account **non-alternating plays**
- represent concurrency by interleavings modulo an equivalence relation, in the spirit of Mazurkiewicz traces:

asynchronous game semantics

Towards asynchronous game semantics

In order to represent such strategies we have to

- take in account **non-alternating plays**
- represent concurrency by interleavings modulo an equivalence relation, in the spirit of Mazurkiewicz traces:

  asynchronous game semantics

- more generally try to bring closer game semantics and **concurrency theory**

The multiplicative-additive linear logic

We consider here MALL formulas (without units):

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}(\parr) \qquad \frac{\vdash \Gamma_1, A \qquad \vdash \Gamma_2, B}{\vdash \Gamma_1, \Gamma_2, A \otimes B}(\otimes)$$

$$\frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}(\&) \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}(\oplus_L) \quad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}(\oplus_R)$$

The multiplicative-additive linear logic

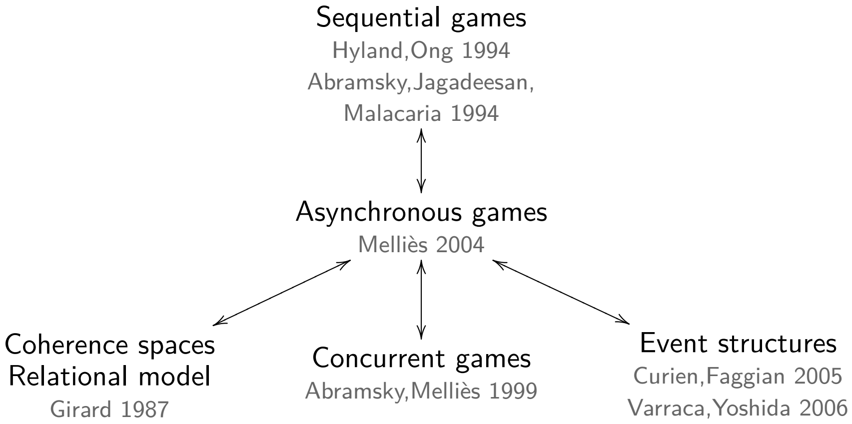
We consider here MALL formulas (without units):

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}(\parr) \qquad \frac{\vdash \Gamma_1, A \qquad \vdash \Gamma_2, B}{\vdash \Gamma_1, \Gamma_2, A \otimes B}(\otimes)$$

$$\frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}(\&) \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}(\oplus_L) \quad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}(\oplus_R)$$

- multiplicatives : concurrency / additives : non-determinism
- negative : Opponent / positive : Player

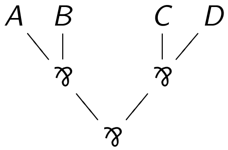
Unifying semantics of linear logic



Proofs explore formulas

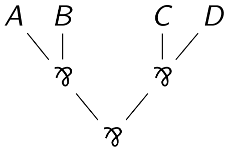
$$\frac{}{(A \parr B) \parr (C \parr D)}$$

Proofs explore formulas

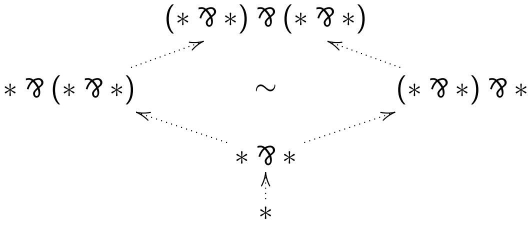


$$\frac{}{(A \parr B) \parr (C \parr D)}$$

Proofs explore formulas

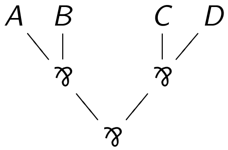


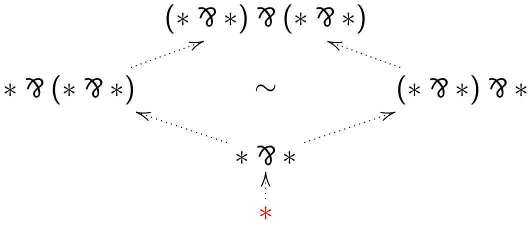
$$\frac{}{(A \parr B) \parr (C \parr D)}$$



Proofs explore formulas

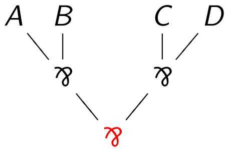
$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash A, B, C, D}}{\vdash A, B, C \parr D}\,(\parr)}{\vdash A \parr B, C \parr D}\,(\parr)}{\vdash (A \parr B) \parr (C \parr D)}\,(\parr)$$

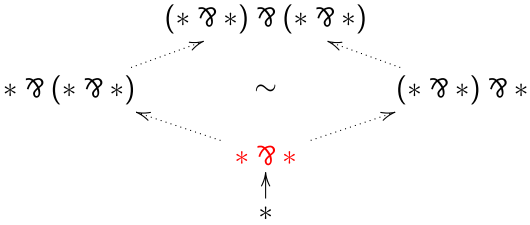




Proofs explore formulas

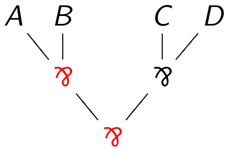
$$
\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash A, B, C, D}}{\vdash A, B, C \parr D}(\parr)}{\color{red}{\vdash A \parr B, C \parr D}}(\parr)}{\vdash (A \parr B) \parr (C \parr D)}(\parr)
$$

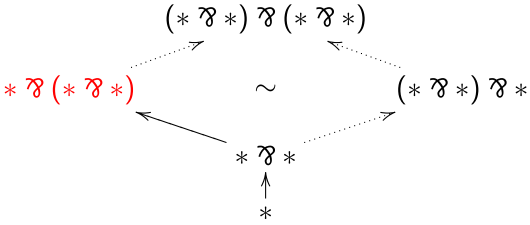




Proofs explore formulas

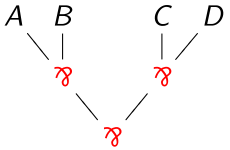
$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash A, B, C, D}}{\vdash A, B, C \parr D}(\parr)}{\vdash A \parr B, C \parr D}(\parr)}{\vdash (A \parr B) \parr (C \parr D)}(\parr)$$

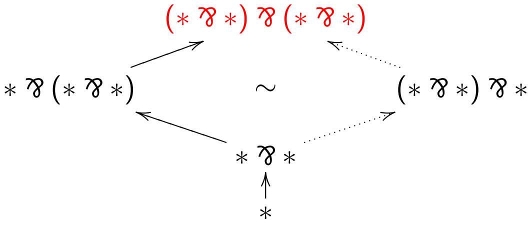




Proofs explore formulas

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash A, B, C, D}}{\vdash A, B, C \parr D}(\parr)}{\vdash A \parr B, C \parr D}(\parr)}{\vdash (A \parr B) \parr (C \parr D)}(\parr)$$





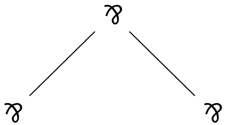
Proofs explore formulas

play	=	exploration of the formula
proof	=	exploration strategy

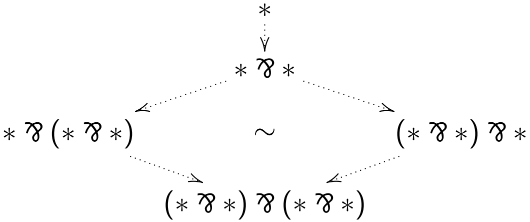
1. Associating an asynchronous game semantics to linear logic
2. Characterizing definable strategies in this semantics
3. Recovering preexisting models

From plays to Mazurkiewicz traces

partial order (event structure) vs transition graph

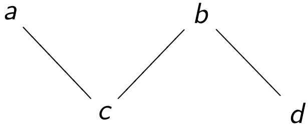


vs

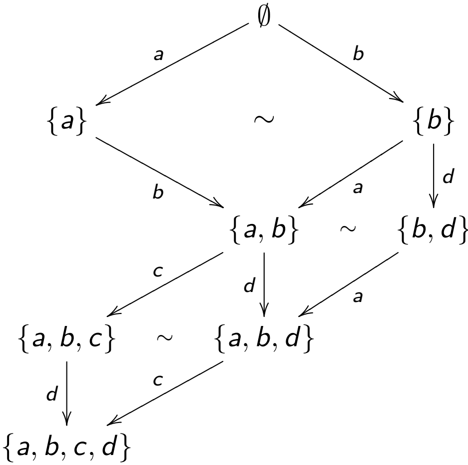


From plays to Mazurkiewicz traces

partial order vs transition graph



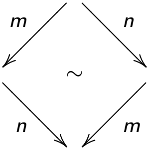
vs



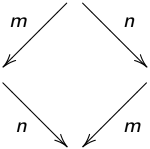
position = downward-closed set of moves

Asynchronous graphs: homotopy

plays

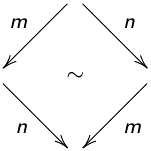


vs

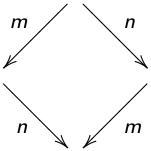


Asynchronous graphs: homotopy

plays

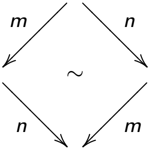
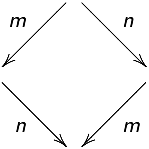


vs

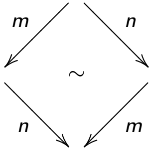
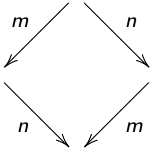


processes $m \| n$ $m \cdot n + n \cdot m$

Asynchronous graphs: homotopy

plays		vs	
processes	$m \| n$		$m \cdot n + n \cdot m$
linear logic	multiplicatives		additives

Asynchronous graphs: homotopy

plays		vs	
processes	$m \| n$		$m \cdot n + n \cdot m$
linear logic	multiplicatives		additives
geometry	possible deformation		hole

Asynchronous games

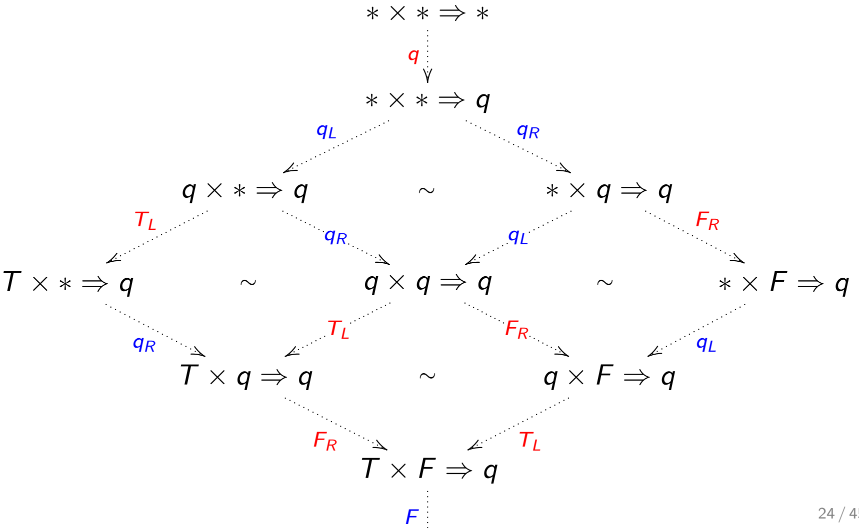
Definition
An **asynchronous game** is an asynchronous graph together with an initial position.

Definition
A **play** is a path in a game starting from the initial position.

Definition
A **strategy** $\sigma : A$ is a prefix closed set of plays on the asynchronous graph A.

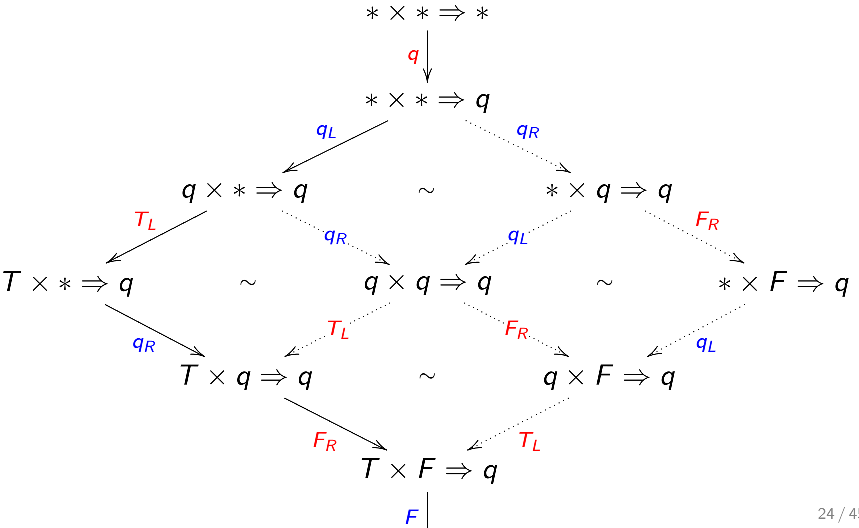
Asynchronous game semantics: conjunction

The game $\mathbb{B} \times \mathbb{B} \Rightarrow \mathbb{B}$ contains eight subgraphs:



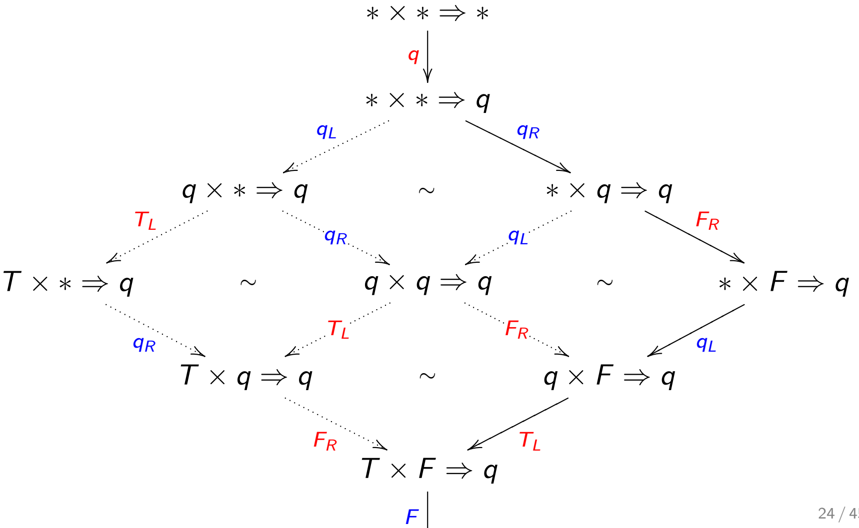
Asynchronous game semantics: conjunction

Left implementation of conjunction:



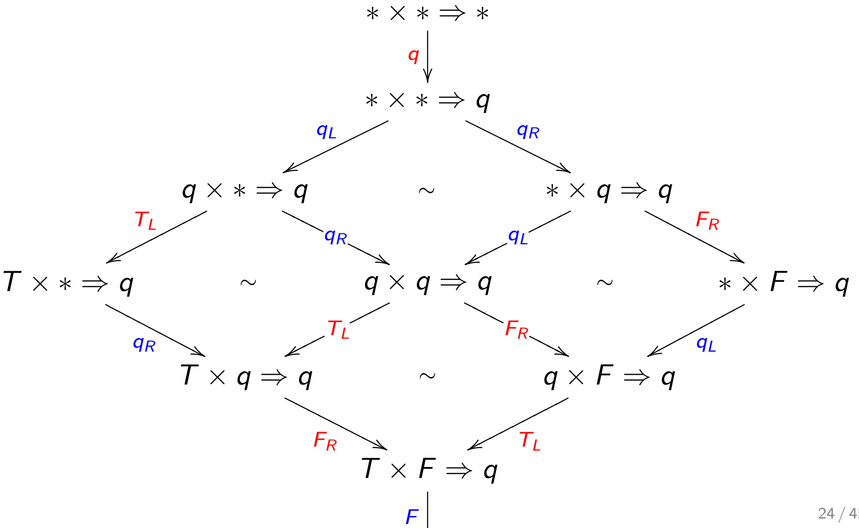
Asynchronous game semantics: conjunction

Right implementation of conjunction:



Asynchronous game semantics: conjunction

Parallel implementation of conjunction:

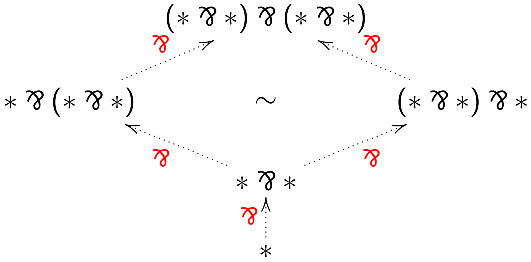


Interpreting formulas and proofs

By an easy inductive definition we associate

- an asynchronous game to every formula

$$\frac{}{(A \parr B) \parr (C \parr D)}$$

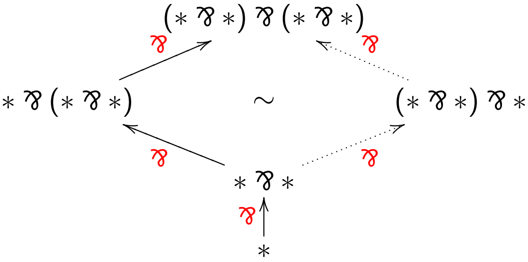


Interpreting formulas and proofs

By an easy inductive definition we associate

- an asynchronous game to every formula
- a strategy to every proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash A, B, C, D}}{\vdash A, B, C \parr D}(\parr)}{\vdash A \parr B, C \parr D}(\parr)}{\vdash (A \parr B) \parr (C \parr D)}(\parr)$$



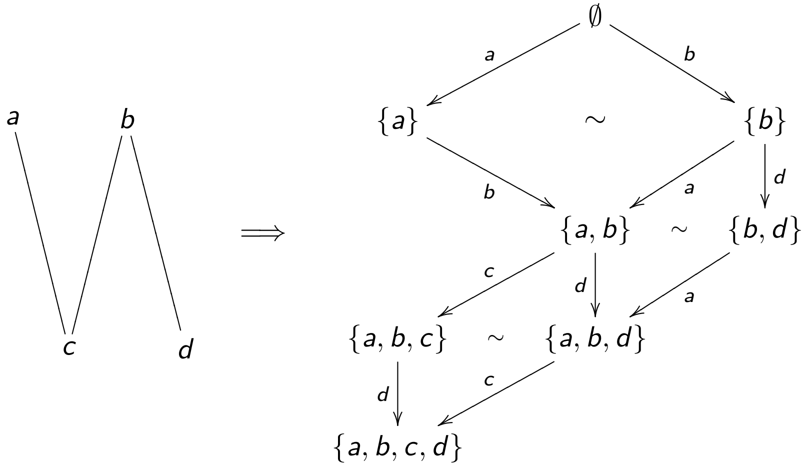
In order to characterize definable strategies,
we will impose conditions on our strategies.

In order to characterize definable strategies,
we will impose conditions on our strategies.

We will begin by some technical conditions which are necessary to regulate the strategies...

From sequentiality to causality

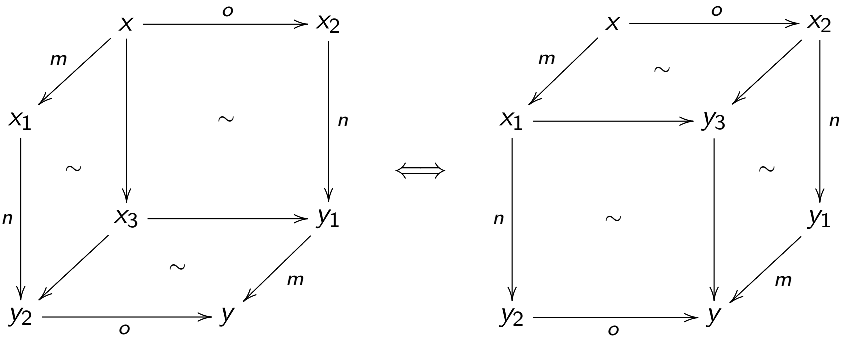
A game induces an asynchronous graph:



From sequentiality to causality

Conversely, one needs the Cube Property

The Cube Property



Theorem

Homotopy classes of paths are generated by a partial order on moves.

Proof: essentially Birkhoff duality theorem for finite posets.

Asynchronous games

Definition

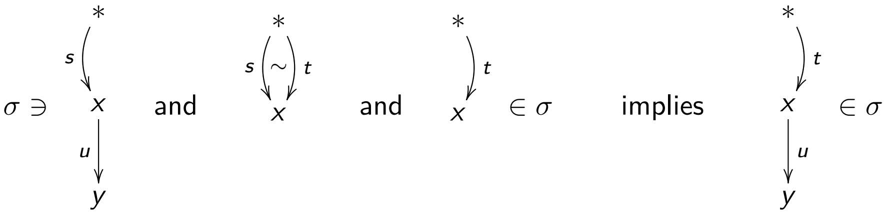
An **asynchronous game** is a pointed asynchronous graph satisfying the Cube Property.

Definition

A **strategy** $\sigma : A$ is a prefix closed set of plays on the asynchronous graph A.

Definition

A strategy σ is **positional** when its plays form a subgraph of the game:



Ingenuous strategies

We consider strategies which

1. are **positional**,

Ingenuous strategies

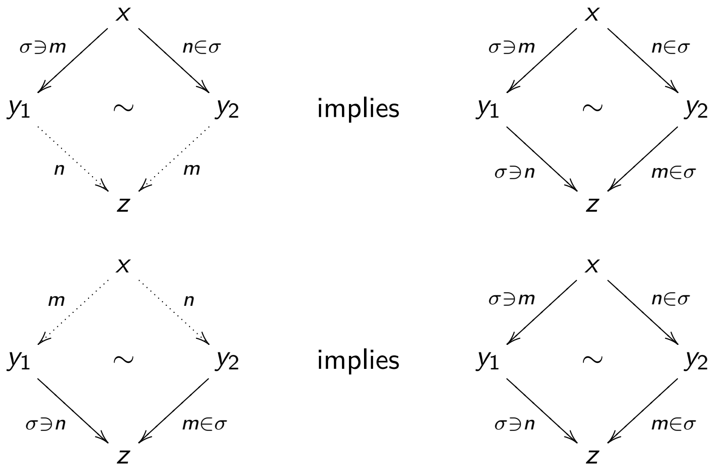
We consider strategies which

1. are **positional**,
2. satisfy the **Cube Property**,

Ingenuous strategies

We consider strategies which

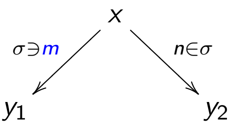
1. are **positional**,
2. satisfy the **Cube Property**,
3. satisfy



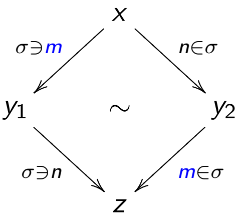
Ingenuous strategies

We consider strategies which

1. are **positional**,
2. satisfy the **Cube Property**,
3. satisfy . . .
4. are **deterministic**:



implies



where m is a Proponent move.

A model of MLL

Property

Asynchronous games and strategies form a $*$-autonomous category (which is compact closed).

This category still has “too many” strategies!

$$A \otimes B \quad = \quad A \parr B$$

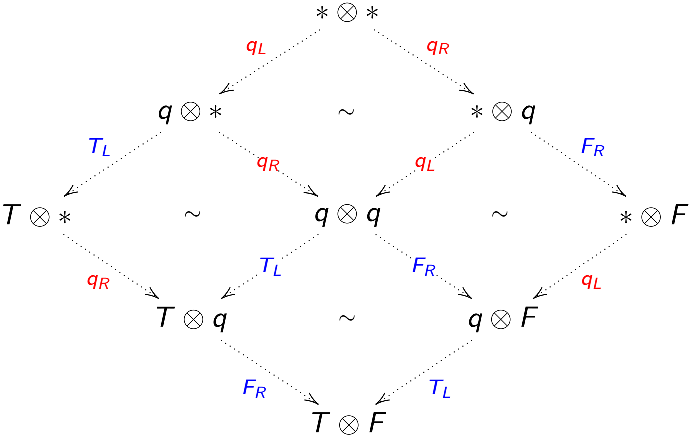
Halting positions

In the spirit of the relational model, a strategy σ should be characterized by its set σ° of halting positions.

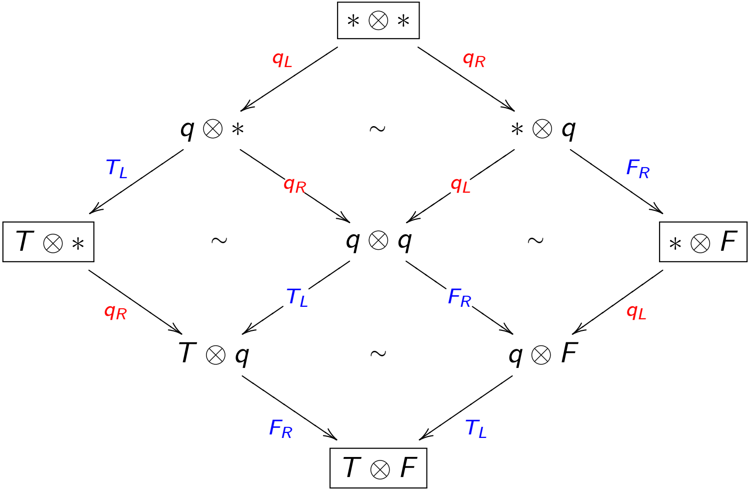
Definition

A **halting position** of a strategy σ is a position x such that there is no Player move $m : x \longrightarrow y$ that σ can play.

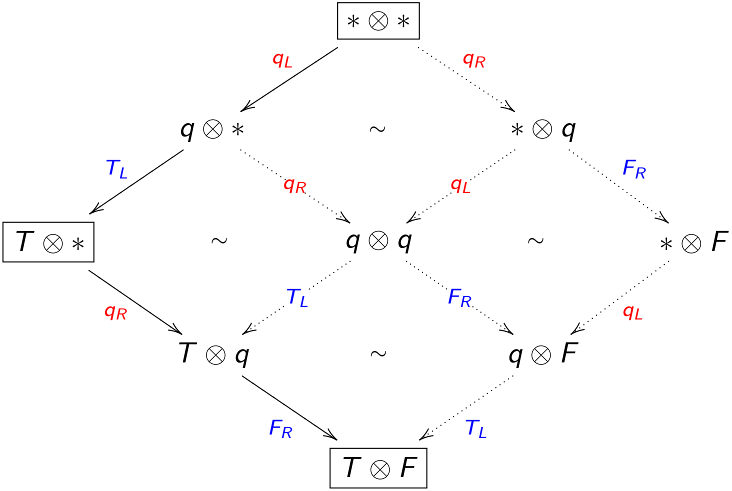
The game $\mathbb{B} \otimes \mathbb{B}$ contains the subgraph:



The pair `true` $\otimes$ `false`:



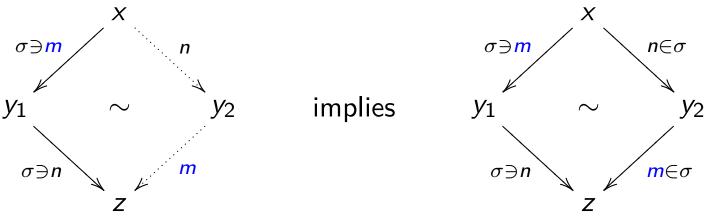
The left biased pair `true` ⩹ `false`:



Courteous strategies

Definition

An ingenuous strategy σ is **courteous** when it satisfies



where m is a Player move.

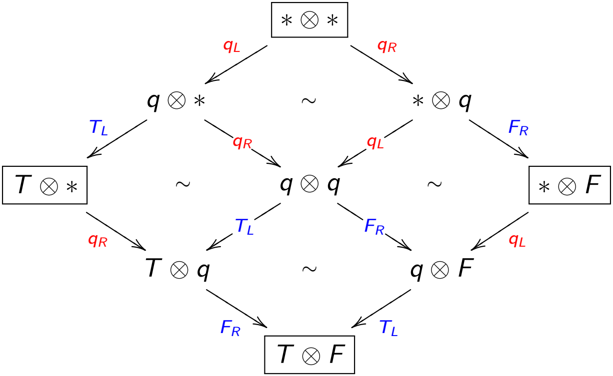
Theorem

A courteous ingenuous strategy σ is characterized by its set σ° of halting positions.

Concurrent strategies

The halting positions of such a strategy $\sigma : A$ are precisely the fixpoints of a **closure operator** on the positions of A.

- We thus recover the model of **concurrent strategies**.
- A semantical counterpart of the **focusing** property: strategies can play all their Player moves in one "cluster" of moves.



Focusing

Some introduction rules can be *permuted*:

$$\dfrac{\dfrac{\vdots}{\vdash A} \qquad \dfrac{\dfrac{\vdots}{\vdash B, C, D}}{\vdash B, C \parr D}(\parr)}{\dfrac{\vdash A \otimes B, C \parr D}{\vdash (A \otimes B) \parr (C \parr D)}(\parr)}(\otimes)$$

Some introduction rules can be *permuted*:

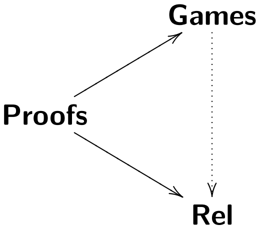
$$\dfrac{\dfrac{\dfrac{\vdots}{\vdash A} \qquad \dfrac{\dfrac{\vdots}{\vdash B, C, D}}{\vdash B, C \parr D}(\parr)}{\vdash A \otimes B, C \parr D}(\otimes)}{\vdash (A \otimes B) \parr (C \parr D)}(\parr) \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash A} \qquad \dfrac{\vdots}{\vdash B, C, D}}{\vdash A \otimes B, C, D}(\otimes)}{\vdash A \otimes B, C, D}(\parr)}{\vdash (A \otimes B) \parr (C \parr D)}(\parr)$$

Every proof can be reorganized into a **focusing** proof:

- *negative phase*: if the sequent contains a negative formula then a negative formula should be decomposed,
- *positive phase*: otherwise a positive formula should be chosen and decomposed repeatedly until a (necessarily unique) formula is produced

Towards a functorial correspondence

The operation $(-)^\circ$ from the category of games
and courteous ingenuous strategies
to the category of relations is not functorial!



This mismatch is essentially due to **deadlock** situations occurring during the interaction.

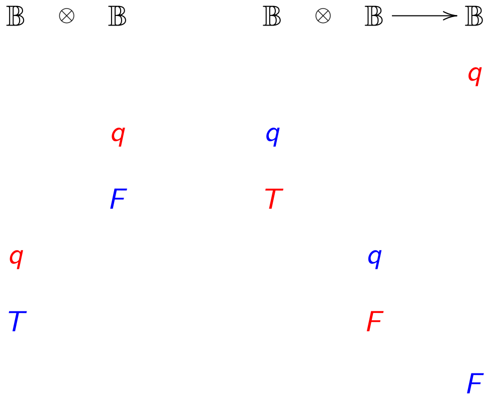
The scheduling criterion

the left conjunction:

$$
\begin{array}{ccccc}
\mathbb{B} & \otimes & \mathbb{B} & \longrightarrow & \mathbb{B} \\
 & & & & q \\
q & & & & \\
T & & & & \\
 & & q & & \\
 & & F & & \\
 & & & & F
\end{array}
$$

The scheduling criterion

The right boolean composed with the left conjunction:



The scheduling criterion

Two kinds of tensors: $\otimes$ and $\parr$.

The scheduling criterion

Two kinds of tensors: $\otimes$ and $\parr$.

$$\mathbb{B} \otimes \mathbb{B}$$

$$q$$

$$F$$

$$q$$

$$T$$

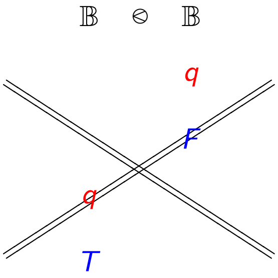
The scheduling criterion

Two kinds of tensors: $\otimes$ and $\parr$.

$\mathbb{B}$	$\ogreaterthan$	$\mathbb{B}$
		q
		F
q		
T		

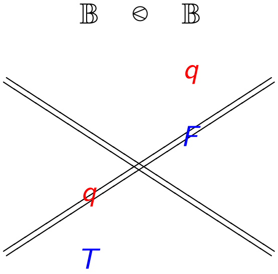
The scheduling criterion

Two kinds of tensors: $\otimes$ and $\parr$.



The scheduling criterion

Two kinds of tensors: $\otimes$ and $\parr$.



The role of the correctness criterion is to avoid deadlocks!

Functoriality

Theorem

Strategies which are

- *ingenuous*
- *courteous*
- *and satisfy the scheduling criterion*

compose and satisfy

$$(\sigma;\tau)^\circ \quad = \quad \sigma^\circ;\tau^\circ$$

Functoriality

Theorem

Strategies which are

- *ingenuous*
- *courteous*
- *and satisfy the scheduling criterion*

compose and satisfy

$$(\sigma;\tau)^\circ \quad = \quad \sigma^\circ;\tau^\circ$$

Theorem

The model we thus get is fully complete for MLL+MIX.

Conclusion

We have:

- a game semantics adapted to concurrency
- an unifying framework in which we recover
  - innocent strategies
  - game semantics
  - concurrent games
  - the relational model
  - event structure semantics

In the future:

- extend this model (exponentials in particular)
- typing of concurrent processes (CCS without deadlocks)
- links with geometrical models for concurrency