Innocence asynchrone et non alternée

Samuel Mimram

Équipe PPS, CNRS, Université Paris VII
(encadré par Paul-André Melliès)

15 mars 2006

Plan

1. La sémantique
2. La sémantique des jeux
3. La sémantique des jeux asynchrones
4. La sémantique des jeux asynchrones non alternés

Première partie I

La sémantique pour les enfants

Je suis un informaticien

$\boxed{\text{sémantique} = \text{donner un sens aux langages de programmation}}$

J'aime les lettres grecques

- Le λ-calcul :

$$T \quad ::= \quad x \quad | \quad \lambda x.T \quad | \quad T_1\,T_2$$

- L'α-conversion :

$$\lambda x.T \quad \equiv_\alpha \quad \lambda y.T[y/x]$$

- La β-reduction :

$$(\lambda x.T_1)\,T_2 \quad \longrightarrow_\beta \quad T_1[T_2/x]$$

J'aime les lettres grecques

- Le λ-calcul :

$$T \quad ::= \quad x \quad | \quad \lambda x.T \quad | \quad T_1\,T_2$$

  Ex : $\lambda f.\lambda x.\lambda y.fy$

- L'α-conversion :

$$\lambda x.T \quad \equiv_\alpha \quad \lambda y.T[y/x]$$

  Ex : $\lambda f.\lambda x.\lambda y.fy \quad \equiv_\alpha \quad \lambda g.\lambda z.\lambda y.gy$

- La β-reduction :

$$(\lambda x.T_1)\,T_2 \quad \longrightarrow_\beta \quad T_1[T_2/x]$$

La β-réduction par l'exemple

$$\begin{array}{rcl} (\lambda f.\lambda x.\lambda y.fy)(\lambda z.zz)tu & \longrightarrow_\beta & (\lambda x.\lambda y.(\lambda z.zz)y)tu \\ & \longrightarrow_\beta & (\lambda y.(\lambda z.zz)y)u \\ & \longrightarrow_\beta & (\lambda z.zz)u \\ & \longrightarrow_\beta & uu \end{array}$$

Une notion de calcul

$$\boxed{\begin{array}{rcl} \lambda\text{-terme} & = & \text{programme} \\ \beta\text{-reduction} & = & \text{calcul} \end{array}}$$

Les types c'est chic

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \Rightarrow B}(\Rightarrow\text{-I}) \qquad \frac{\Gamma \vdash M : A \Rightarrow B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B}(\Rightarrow\text{-E})$$

Les types c'est chic

$$\frac{\Gamma, \quad A \vdash \quad B}{\Gamma \vdash \quad A \Rightarrow B}(\Rightarrow\text{-I}) \qquad \frac{\Gamma \vdash \quad A \Rightarrow B \qquad \Gamma \vdash \quad A}{\Gamma \vdash \quad B}(\Rightarrow\text{-E})$$

Hurry coward

Isomorphisme de Curry-Howard :

$$\boxed{\text{programme typé} \qquad = \qquad \text{preuve de son type}}$$

β-réduction vs Curry-Howard

On a une dérivation :

$$\frac{\Gamma \vdash M : A \Rightarrow B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B}(\Rightarrow\text{-E})$$

et

$$MN \longrightarrow_\beta P$$

Que devient la preuve ?

β-réduction vs Curry-Howard

On a une dérivation :

$$\frac{\Gamma \vdash M : A \Rightarrow B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B}(\Rightarrow\text{-E})$$

et

$$MN \longrightarrow_\beta P$$

Que devient la preuve ?

$$\boxed{\text{la } \beta\text{-réduction préserve le typage}}$$

$$\boxed{\beta\text{-réduction (termes)} = \text{élimination des coupures (preuves)}}$$

Qu'est-ce qu'une sémantique ?

On interprète :

- $[\![A]\!]$: ensemble
- $[\![f]\!]$ avec $(f : A \Rightarrow B)$: fonction $[\![A]\!] \to [\![B]\!]$

Dénotationnel :

$$M \longrightarrow_\beta N \qquad \text{implique} \qquad [\![M]\!] = [\![N]\!]$$

Modulaire : ça doit être une catégorie[1]

[1] Ceci est la seule occurrence du mot *catégorie*.

Qu'est-ce qu'une bonne sémantique ?

On veut une sémantique qui soit la plus précise possible,
c'est-à-dire qui soit la plus proche possible de la syntaxe.

- Modèle correct :

$$[\![M]\!] = [\![N]\!] \qquad \text{implique} \qquad M \equiv N$$

- Modèle complet :

$$M \equiv N \qquad \text{implique} \qquad [\![M]\!] = [\![N]\!]$$

- Modèle pleinement complet :

$$\forall \mathcal{M}, \exists M, \qquad \mathcal{M} \quad = \quad [\![M]\!]$$

$$\boxed{\text{modèle pleinement complet} \qquad = \qquad \text{syntaxe abstraite}}$$

Deuxième partie II

La sémantique des jeux

La sémantique des jeux

C'est une sémantique

- extensionnelle
- interactive

`not` :

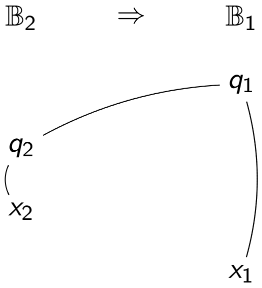
$$
\begin{array}{ccc}
\mathbb{B}_2 & \Rightarrow & \mathbb{B}_1 \\
 & & q_1 \\
q_2 & & \\
x_2 & & \\
 & & x_1
\end{array}
$$

La sémantique des jeux

C'est une sémantique

- extensionnelle
- interactive

`not` :



Formalisation

$$\boxed{\begin{array}{rcl} [\![\text{formule}]\!] & = & \text{jeu} \\ [\![\text{preuve}]\!] & = & \text{stratégie} \end{array}}$$

Jeux

Un **jeu** (ou *arène*) A est un triplet $A = \langle M_A, \lambda_A, \leq_A \rangle$ où

- M_A : *coups*,
- $\lambda_A : M_A \to \{J, O\} \times \{Q, R\}$: *fonction de polarisation*,
- $\leq_A$: *justification*.

$$\mathbb{B} : \begin{tikzcd} & q \arrow[dl] \arrow[dr] & \\ V & & F \end{tikzcd}$$

Parties

Une **partie** est une suite *alternée* de coups

$$s \quad = \quad m_1 \cdot m_1 \cdots m_n$$

"cohérente avec l'ordre".

$$\boxed{\text{partie} = \text{une interaction}}$$

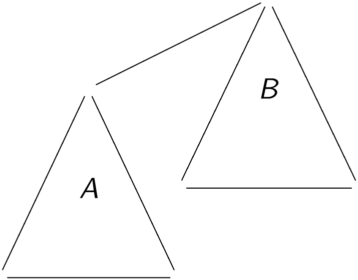
Stratégies

Une **stratégie** $\sigma : A$ est un ensemble de parties de longueur paire

- non-vide,
- clos par préfixe.

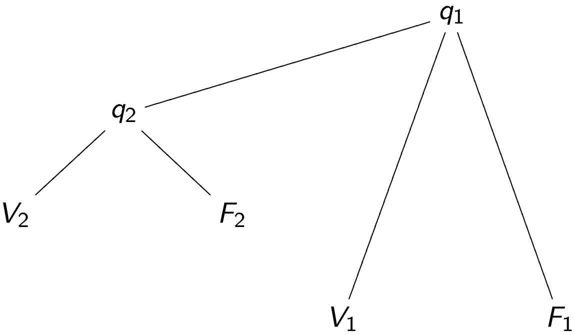
$$\boxed{\text{statégie} \quad = \quad \text{comment on répond à l'environement}}$$

L'arène $A \Rightarrow B$:



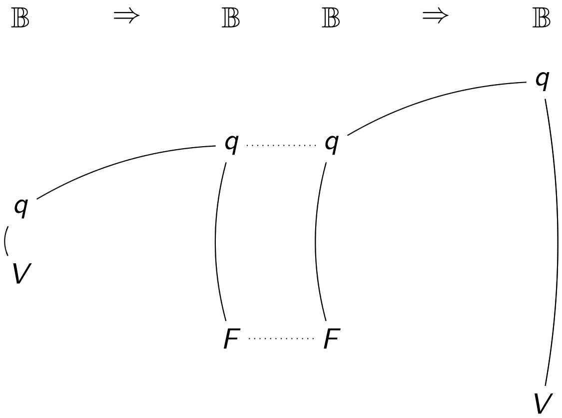
Stratégies

L'arène $\mathbb{B} \Rightarrow \mathbb{B}$:

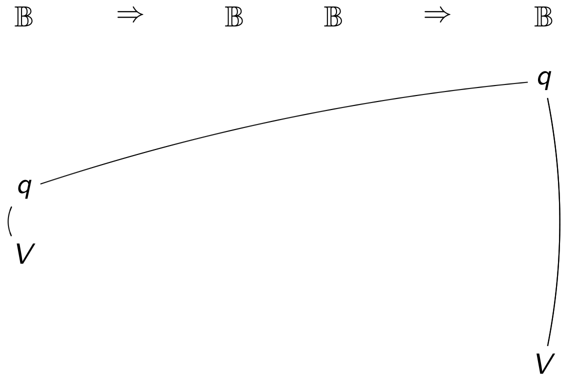


Interaction

Composition de `not` : $\mathbb{B} \Rightarrow \mathbb{B}$ et `not` : $\mathbb{B} \Rightarrow \mathbb{B}$:



Composition de `not` : $\mathbb{B} \Rightarrow \mathbb{B}$ et `not` : $\mathbb{B} \Rightarrow \mathbb{B}$:



Troisième partie III

L'innocence

Déterminisme

Une stratégie σ est **déterministe** lorsque :

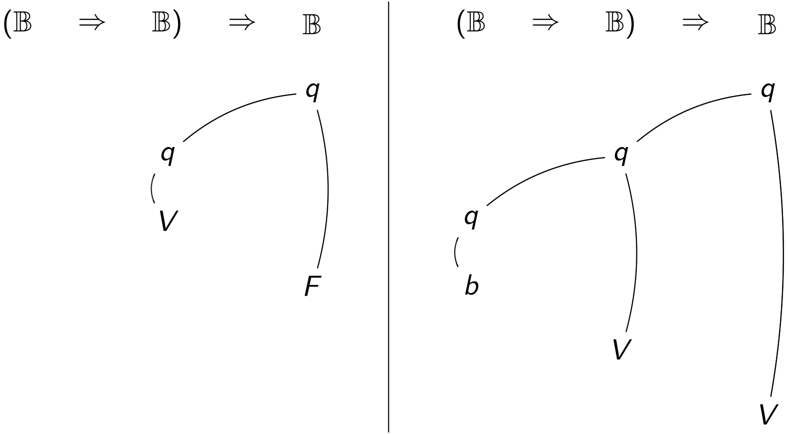
$$\forall s \cdot m \cdot n_1, s \cdot m \cdot n_2 \in \sigma, \quad n_1 = n_2$$

$\boxed{\text{on répond toujours la même chose à un même coup de l'opposant}}$

Innocence

Une stratégie σ est **innocente** lorsqu'

$\boxed{\text{elle ne peut pas observer les calculs intermédiaires de l'environnement}}$



Bon parenthésage

$\boxed{\text{on répond toujours à la dernière question}}$

Et avec ceci ?

jeu	=	formule	=	ordre partiel
stratégie	=	arbre de Böhm	=	ordre partiel

$\boxed{\text{Un arbre de Böhm raffine sa formule}}$

Quatrième partie IV

Un exemple

Considérons le λ-terme

$$\lambda f.\lambda g.g(\lambda y.(fy)) \quad : \quad (A \Rightarrow B) \Rightarrow ((A \Rightarrow B) \Rightarrow C) \Rightarrow C$$

Considérons le λ-terme

$$\lambda f.\lambda g.g(\lambda y.(fy)) \quad : \quad (A \Rightarrow B) \Rightarrow ((A \Rightarrow B) \Rightarrow C) \Rightarrow C$$

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{(Ax)}
\UnaryInfC{$\Gamma \vdash g : (A \Rightarrow B) \Rightarrow C$}
\AxiomC{}
\RightLabel{(Ax)}
\UnaryInfC{$\Gamma, y : A \vdash f : A \Rightarrow B$}
\AxiomC{}
\RightLabel{(Ax)}
\UnaryInfC{$\Gamma, y : A \vdash y : A$}
\RightLabel{($\Rightarrow$-E)}
\BinaryInfC{$\Gamma, y : A \vdash fy : B$}
\RightLabel{($\Rightarrow$-I)}
\UnaryInfC{$\Gamma \vdash \lambda y.fy : A \Rightarrow B$}
\RightLabel{($\Rightarrow$-E)}
\BinaryInfC{$f : A \Rightarrow B, g : (A \Rightarrow B) \Rightarrow C \vdash g(\lambda y.(fy)) : C$}
\UnaryInfC{$\vdash \lambda f.\lambda g.g(\lambda y.(fy)) : (A \Rightarrow B) \Rightarrow ((A \Rightarrow B) \Rightarrow C) \Rightarrow C$}
\end{prooftree}
$$

$$(A \quad \Rightarrow \quad B) \quad \Rightarrow \quad ((A \quad \Rightarrow \quad B) \quad \Rightarrow \quad C) \quad \Rightarrow \quad \underset{q}{C}$$

$$\lambda f.\lambda g. \qquad\qquad : \qquad\qquad\qquad\qquad \Rightarrow C$$

$$(A \quad \Rightarrow \quad B) \quad \Rightarrow \quad ((A \quad \Rightarrow \quad B) \quad \Rightarrow \quad C) \quad \Rightarrow \quad C$$

$$q \qquad q$$

$$\lambda f.\lambda g.g \qquad : \qquad (\qquad \Rightarrow C) \Rightarrow C$$

$$(A \quad \Rightarrow \quad B) \quad \Rightarrow \quad ((A \quad \Rightarrow \quad B) \quad \Rightarrow \quad C) \quad \Rightarrow \quad C$$

$$\begin{array}{ccccccccc} & & & & & & & & q \\ & & & & & & q & & \\ & & & & q & & & & \end{array}$$

$$\lambda f.\lambda g.g(\lambda y.\qquad) \quad : \qquad\qquad ((\quad \Rightarrow B) \Rightarrow C) \Rightarrow C$$

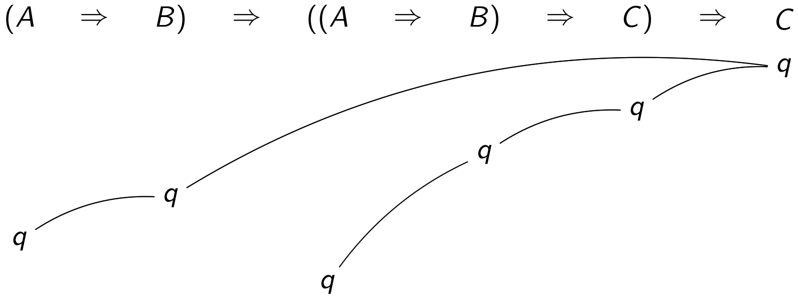
$$(A \quad \Rightarrow \quad B) \quad \Rightarrow \quad ((A \quad \Rightarrow \quad B) \quad \Rightarrow \quad C) \quad \Rightarrow \quad C$$

$$\begin{array}{ccccccccc} & & & & & & & & q \\ & & & & & & q & & \\ & & & & q & & & & \\ & q & & & & & & & \end{array}$$

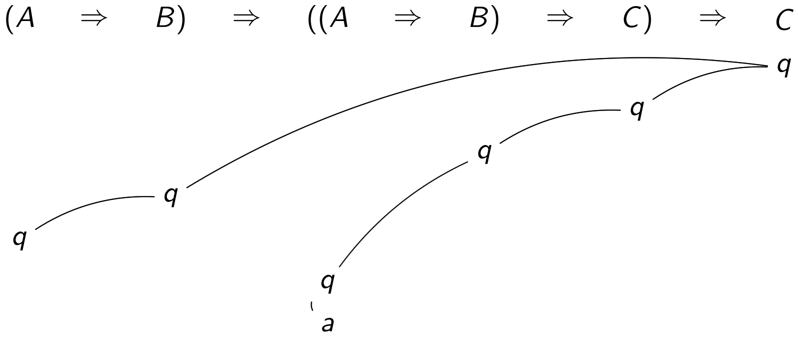
$$\lambda f.\lambda g.g(\lambda y.(f \quad)) \quad : \quad (\quad \Rightarrow B) \Rightarrow ((\quad \Rightarrow B) \Rightarrow C) \Rightarrow C$$

$$
\begin{array}{ccccccccccccc}
(A & \Rightarrow & B) & \Rightarrow & ((A & \Rightarrow & B) & \Rightarrow & C) & \Rightarrow & C \\
 & & & & & & & & & & q \\
 & & & & & & & & q & & \\
 & & & & & & q & & & & \\
 & & q & & & & & & & & \\
q & & & & & & & & & &
\end{array}
$$

$$\lambda f.\lambda g.g(\lambda y.(f \quad)) \quad : \quad (A \Rightarrow B) \Rightarrow ((\quad \Rightarrow B) \Rightarrow C) \Rightarrow C$$

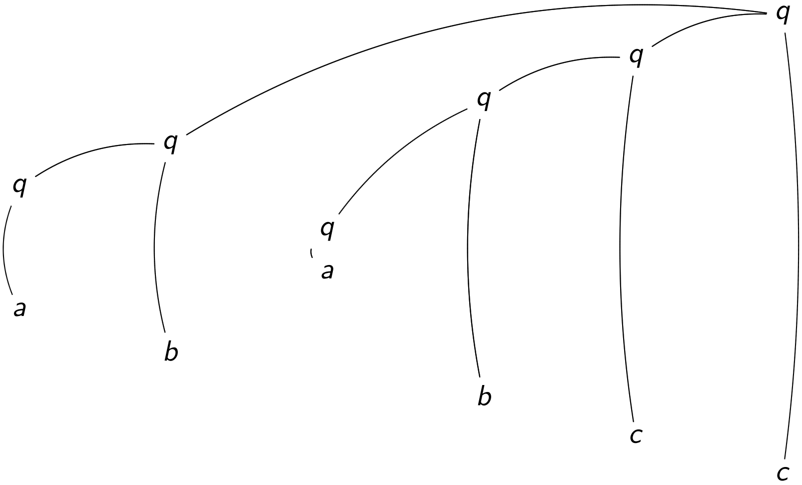


$$\lambda f.\lambda g.g(\lambda y.(fy)) \quad : \quad (A \Rightarrow B) \Rightarrow ((A \Rightarrow B) \Rightarrow C) \Rightarrow C$$



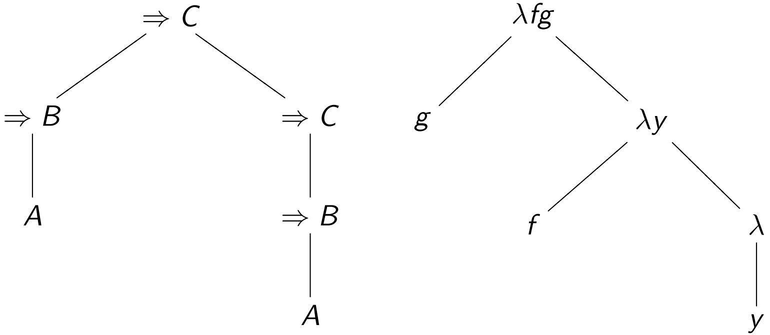
$$\lambda f.\lambda g.g(\lambda y.(fy)) \quad : \quad (A \Rightarrow B) \Rightarrow ((A \Rightarrow B) \Rightarrow C) \Rightarrow C$$

$$(A \quad \Rightarrow \quad B) \quad \Rightarrow \quad ((A \quad \Rightarrow \quad B) \quad \Rightarrow \quad C) \quad \Rightarrow \quad C$$



$$\lambda f.\lambda g.g(\lambda y.(fy)) \quad : \quad (A \Rightarrow B) \Rightarrow ((A \Rightarrow B) \Rightarrow C) \Rightarrow C$$

Exploration de $(A \Rightarrow B) \Rightarrow ((A \Rightarrow B) \Rightarrow C) \Rightarrow C$



jeu   stratégie

Exploration de $(A \Rightarrow B) \Rightarrow ((A \Rightarrow B) \Rightarrow C) \Rightarrow C$

$$
\begin{tikzcd}
 & \Rightarrow C \arrow[dl, dash] \arrow[dr, dash] & \\
\Rightarrow B \arrow[d, dash] & & \Rightarrow C \arrow[d, dash] \\
A & & \Rightarrow B \arrow[d, dash] \\
 & & A
\end{tikzcd}
$$

jeu

$$
\begin{tikzcd}
 & \Rightarrow C \arrow[dr] & \\
\Rightarrow B \arrow[d] & & \Rightarrow C \arrow[d] \\
A \arrow[drr] & & \Rightarrow B \arrow[ull] \\
 & & A
\end{tikzcd}
$$

stratégie

L'innocence

Une stratégie innocente est :

1. une exploration de sa formule
2. caractérisée par un ordre partiel sur les coups
3. avec des propriétés

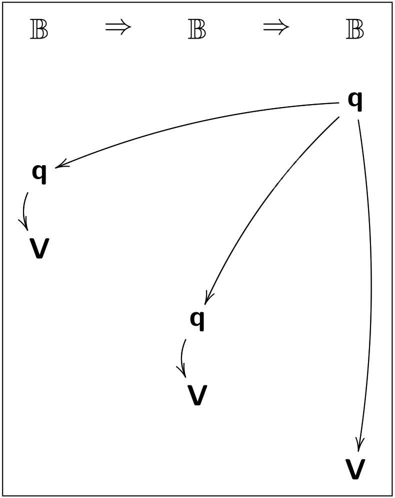
Ces ordres engendrent des *traces*, on aimerait caractériser l'innocence directement sur les traces par des *propriétés diagrammatiques* locales.

Cinquième partie V

Jeux asynchrones et non-alternés

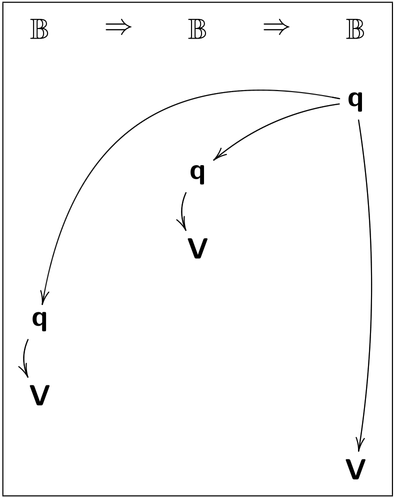
Sémantique des jeux alternée

et gauche



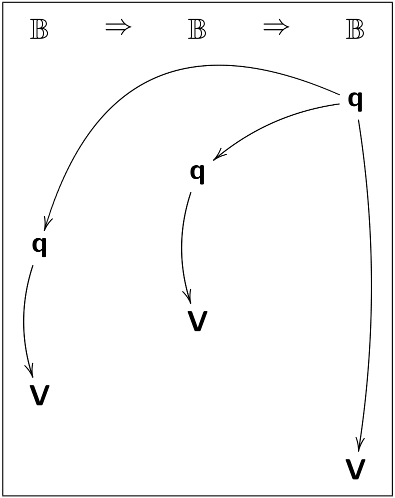
Sémantique des jeux alternée

et droit



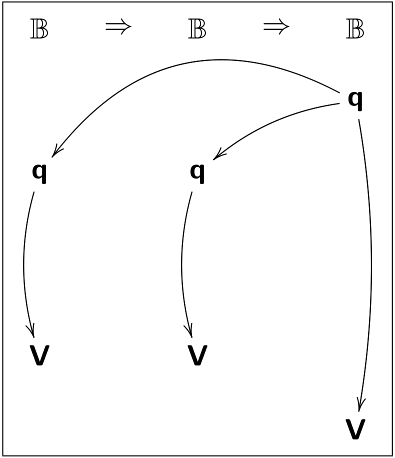
Sémantique des jeux non-alternée

et parallèle



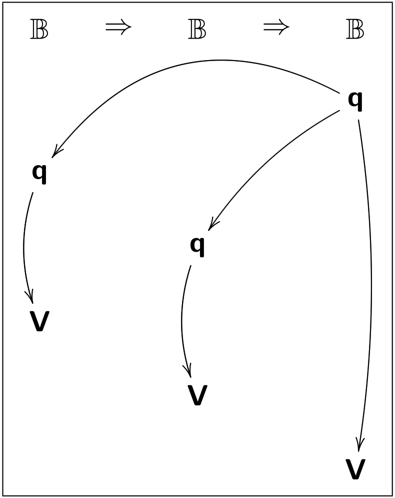
Sémantique des jeux non-alternée

et parallèle



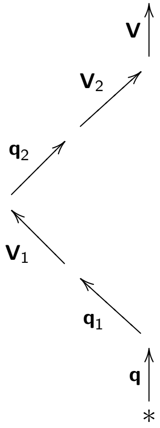
Sémantique des jeux non-alternée

et parallèle



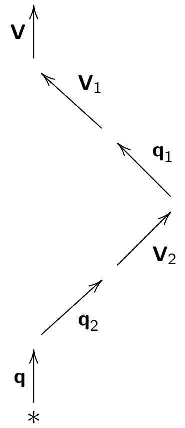
Stratégies non-alternées asynchrones

et gauche



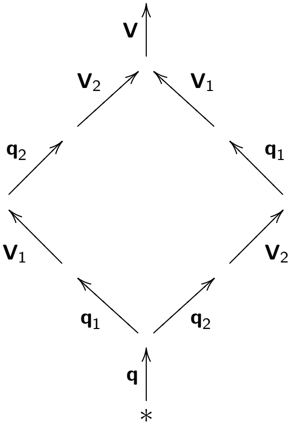
Stratégies non-alternées asynchrones

et droit



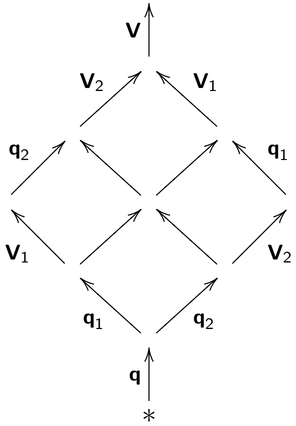
Stratégies non-alternées asynchrones

`et` parallèle



Stratégies non-alternées asynchrones

`et` parallèle



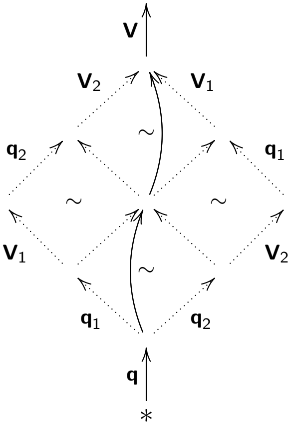
Stratégies non-alternées asynchrones

et parallèle

$$
\begin{tikzcd}
 & & {} & & \\
 & & {} \arrow[u, "\mathbf{V}"] & & \\
 & {} \arrow[ur, "\mathbf{V}_2"] & & {} \arrow[ul, "\mathbf{V}_1"'] & \\
{} \arrow[ur, "\mathbf{q}_2"] & & {} \arrow[ul] \arrow[ur] \arrow[u, phantom, "\sim"] & & {} \arrow[ul, "\mathbf{q}_1"'] \\
 & {} \arrow[ul, "\mathbf{V}_1"] \arrow[ur] \arrow[u, phantom, "\sim"] & & {} \arrow[ul] \arrow[ur, "\mathbf{V}_2"'] \arrow[u, phantom, "\sim"] & \\
 & & {} \arrow[ul, "\mathbf{q}_1"] \arrow[ur, "\mathbf{q}_2"'] \arrow[u, phantom, "\sim"] & & \\
 & & * \arrow[u, "\mathbf{q}"] & &
\end{tikzcd}
$$

Stratégies non-alternées asynchrones

`et` parallèle



L'homotopie

$$\begin{tikzcd}
 & {} & \\
{} \arrow[ur, "n"] \arrow[rr, phantom, "\sim"] & & {} \arrow[ul, "m"'] \\
 & {} \arrow[ul, "m"] \arrow[ur, "n"'] &
\end{tikzcd}$$

L'homotopie

$$\begin{tikzcd}
& {} & \\
{} \arrow[ur, "\texttt{write}(y)"] & \sim & {} \arrow[ul, "\texttt{read}(x)"'] \\
& {} \arrow[ul, "\texttt{read}(x)"] \arrow[ur, "\texttt{write}(y)"'] &
\end{tikzcd}$$

L'homotopie

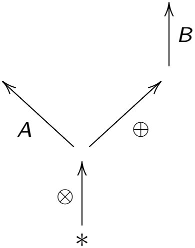
$$
\begin{tikzcd}
 & \cdot & \\
\cdot \arrow[ur, "\texttt{write}(x)"] & & \cdot \arrow[ul, "\texttt{read}(x)"'] \\
 & \cdot \arrow[ul, "\texttt{read}(x)"] \arrow[ur, "\texttt{write}(x)"'] &
\end{tikzcd}
$$

L'homotopie

$$
\begin{tikzcd}
 & {} & \\
{} \arrow[ur, dotted, "n"] & & {} \arrow[ul, dotted, "m"'] \\
 & {} \arrow[ul, dotted, "m"] \arrow[ur, dotted, "n"'] \arrow[uu, bend right, "\sim"] &
\end{tikzcd}
$$

La logique linéaire n'est pas alternée

$$\dfrac{\dfrac{\vdots}{A} \qquad \dfrac{\dfrac{\vdots}{B}}{B \oplus C}}{A \otimes (B \oplus C)}$$



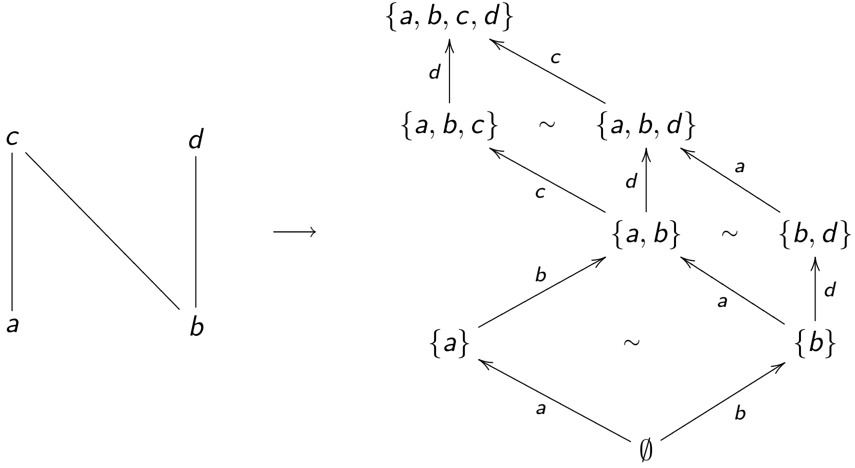
Notre cadre

- jeu : 2-graphe pointé par $*$
- stratégie : ensemble de chemins partant de $*$

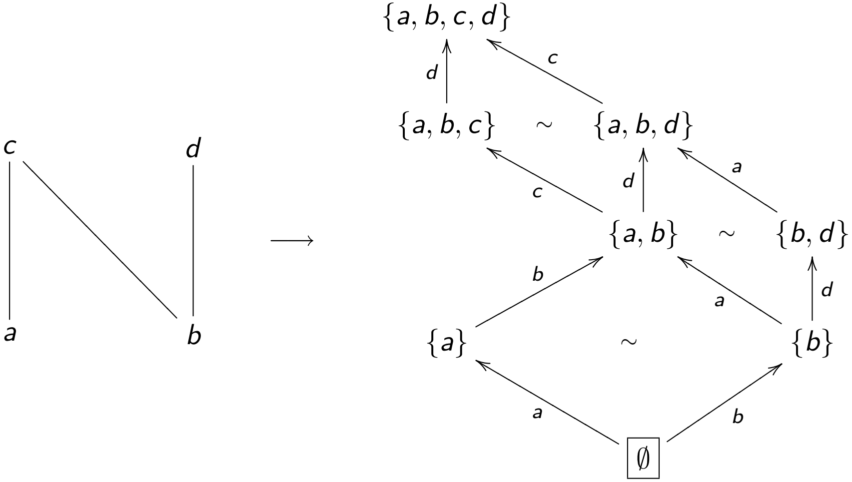
Sixième partie VI

Innocence asynchrone et non-alternée

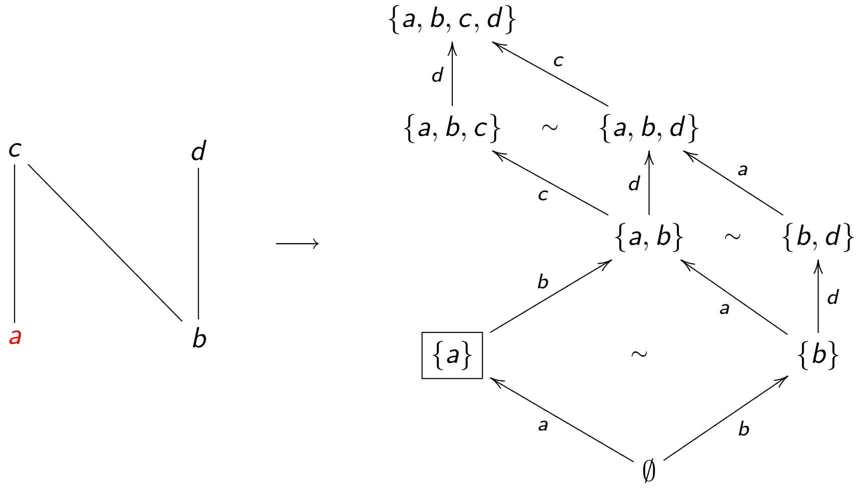
Ordre partiel et graphe positionnel



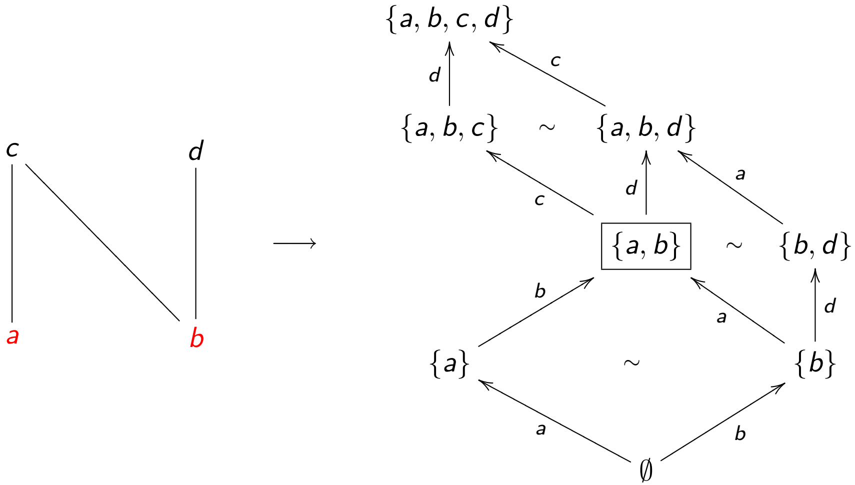
Ordre partiel et graphe positionnel



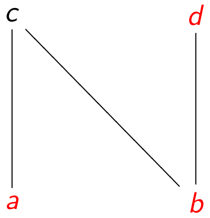
Ordre partiel et graphe positionnel



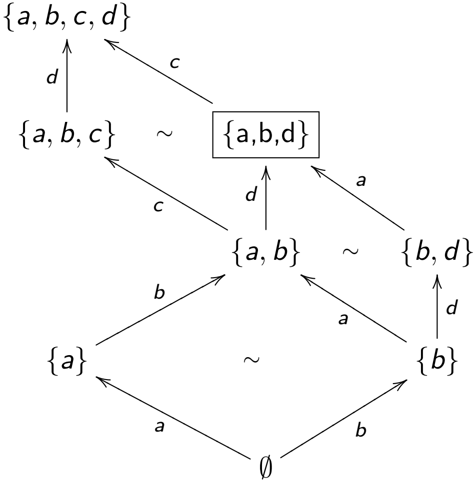
Ordre partiel et graphe positionnel



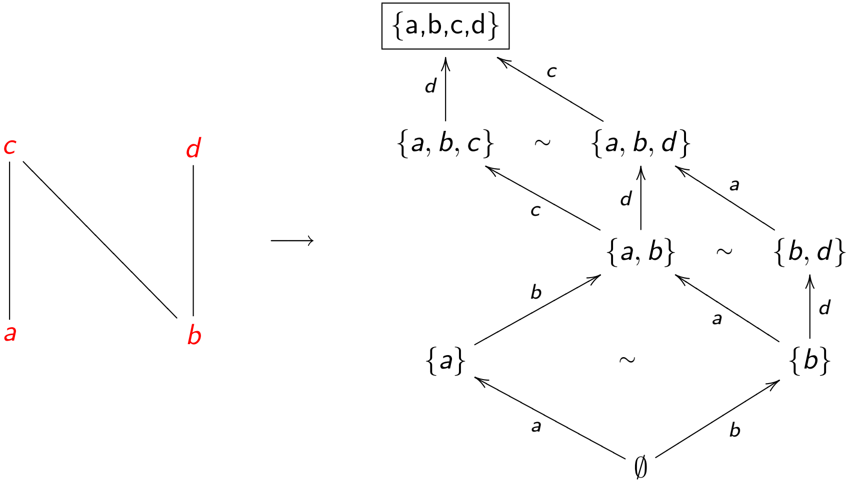
Ordre partiel et graphe positionnel



$\longrightarrow$



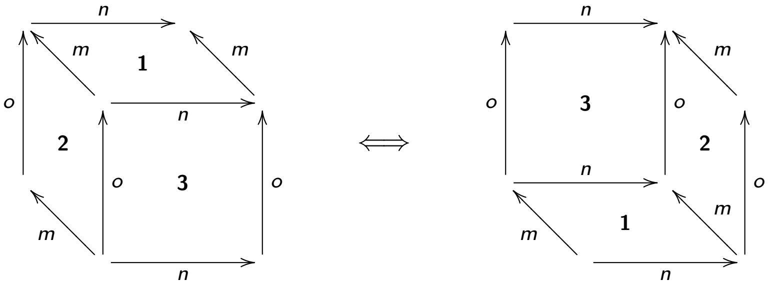
Ordre partiel et graphe positionnel



La Propriété du Cube

$$
\begin{tikzcd}
{} \arrow[rr, "n"] & {} \arrow[dr, phantom, "\sim"] & {} & {} \\
{} \arrow[dr, phantom, "\sim"] & {} \arrow[ul, "m"'] \arrow[rr, "n"] \arrow[ddrr, phantom, "\sim"] & {} & {} \arrow[ul, "m"'] \\
{} \arrow[uu, "o"] & {} & {} & {} \\
{} & {} \arrow[ul, "m"] \arrow[uu, "o"'] \arrow[rr, "n"'] & {} & {} \arrow[uu, "o"']
\end{tikzcd}
\quad \Longleftrightarrow \quad
\begin{tikzcd}
{} \arrow[rr, "n"] \arrow[ddrr, phantom, "\sim"] & {} & {} & {} \\
{} & {} & {} \arrow[dr, phantom, "\sim"] & {} \arrow[ul, "m"'] \\
{} \arrow[uu, "o"] \arrow[rr, "n"] & {} \arrow[dr, phantom, "\sim"] & {} \arrow[uu, "o"'] & {} \\
{} & {} \arrow[ul, "m"] \arrow[rr, "n"'] & {} & {} \arrow[ul, "m"'] \arrow[uu, "o"']
\end{tikzcd}
$$

La Propriété du Cube



$\mathbf{1} : m \| n$ $\qquad$ $\mathbf{2} : m \| o$ $\qquad$ $\mathbf{3} : n \| o$

Innocence 1 – La positionalité

Positionalité
On veut que les parties d'une stratégie soient les chemins d'un sous-graphe du jeu.

Innocence 1 – La positionalité

1. Propriété d'extension

$$
\begin{tikzcd}
 & {} & \\
 & {} \arrow[u, twoheadrightarrow, "u"] & \\
{} \arrow[ur, "n"] & \sim & {} \arrow[ul, "m"'] \\
 & {} \arrow[ul, "m"] \arrow[ur, "n"'] & \\
 & * \arrow[u, twoheadrightarrow, "s"] &
\end{tikzcd}
$$

$$
\frac{s \cdot m \cdot n \in \sigma \qquad s \cdot n \cdot m \in \sigma \qquad s \cdot m \cdot n \cdot u \in \sigma}{s \cdot n \cdot m \cdot u \in \sigma}
$$

2. Préservation de la compatibilité

$$
\begin{tikzcd}
 & & {} & & \\
\sigma \ni & {} \arrow[ur, dotted, "n"] & \sim & {} \arrow[ul, dotted, "m"'] & \in \sigma \\
 & & {} \arrow[ul, "m"] \arrow[ur, "n"'] & & \\
 & & * \arrow[u, twoheadrightarrow, "s"] & &
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
 & & {} & & \\
\sigma \ni & {} \arrow[ur, "n"] & \sim & {} \arrow[ul, "m"'] & \in \sigma \\
 & & {} \arrow[ul, "m"] \arrow[ur, "n"'] & & \\
 & & * \arrow[u, twoheadrightarrow, "s"] & &
\end{tikzcd}
$$

3. La Propriété du cube

$$
\begin{tikzcd}
{} \arrow[rr, "n"] \arrow[drr, phantom, "\sim"] & & {} & \\
& {} \arrow[ul, "m"'] \arrow[rr, "n"] \arrow[drr, phantom, "\sim"] & & {} \arrow[ul, "m"'] \\
{} \arrow[uu, "o"] \arrow[ur, phantom, "\sim"] & & & \\
& {} \arrow[ul, "m"] \arrow[uu, "o"] \arrow[rr, "n"'] & & {} \arrow[uu, "o"'] \\
& * \arrow[u, two heads, "s"] & &
\end{tikzcd}
\quad \Longleftrightarrow \quad
\begin{tikzcd}
{} \arrow[rr, "n"] \arrow[drr, phantom, "\sim"] & & {} & \\
& & {} \arrow[r, phantom, "\sim"] & {} \arrow[ul, "m"'] \\
{} \arrow[uu, "o"] \arrow[rr, "n"] \arrow[drr, phantom, "\sim"] & & {} \arrow[uu, "o"'] & \\
& {} \arrow[ul, "m"] \arrow[rr, "n"'] & & {} \arrow[ul, "m"'] \arrow[uu, "o"'] \\
& * \arrow[u, two heads, "s"] & &
\end{tikzcd}
$$

Innocence 3 – Gérer la composition

4. Déterminisme

$$\sigma \ni \begin{tikzcd} \cdot & & \cdot \\ & \cdot \arrow[ul, "m"] \arrow[ur, "n"'] & \\ & * \arrow[u, twoheadrightarrow, "s"] & \end{tikzcd} \in \sigma \quad \Longrightarrow \quad \sigma \ni \begin{tikzcd} & \cdot & \\ \cdot \arrow[ur, "n"] & & \cdot \arrow[ul, "m"'] \\ & \cdot \arrow[ul, "m"] \arrow[ur, "n"'] \arrow[u, phantom, "\sim"] & \\ & * \arrow[u, twoheadrightarrow, "s"] & \end{tikzcd} \in \sigma$$

Innocence 3 – Gérer la composition

5. Courtoisie

$$\sigma \ni \begin{tikzcd} & {} & \\ {} \arrow[ur, "n"] & \sim & {} \arrow[ul, dotted, "m"'] \\ & {} \arrow[ul, "m"] \arrow[ur, dotted, "n"'] & \\ & * \arrow[u, twoheadrightarrow, "s"] & \end{tikzcd} \Longrightarrow \sigma \ni \begin{tikzcd} & {} & \\ {} \arrow[ur, "n"] & \sim & {} \arrow[ul, "m"'] \\ & {} \arrow[ul, "m"] \arrow[ur, "n"'] & \\ & * \arrow[u, twoheadrightarrow, "s"] & \end{tikzcd} \in \sigma$$

Innocence 3 – Gérer la composition

6. Réceptivité

$$
\sigma \ni
\begin{tikzcd}
\cdot \\
\cdot \arrow[u, dotted, "m"] \\
* \arrow[u, two heads, "s"]
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
\cdot \\
\cdot \arrow[u, "m"] \\
* \arrow[u, two heads, "s"]
\end{tikzcd}
\in \sigma
$$

Conclusion

- On a une notion de jeux asynchrone non-alterné
- On a une notion d'innocence dans ce cadre
- Il reste du travail à faire
- (en particulier modéliser des π-calcul-like)