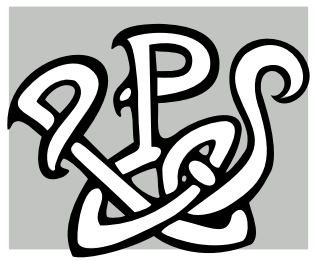
Jeux et Sémantique

Samuel Mimram

Laboratoire PPS, CNRS – Univ. Paris Diderot

Journée Théorie des Jeux et Informatique
18 février 2009



Origines de la sémantique des jeux

Précurseurs :

- Lorenzen [Lor61],

prouvabilité

$\Leftrightarrow$

existence d'une stratégie sur le jeu associé

$\forall x.\exists y.P$

Origines de la sémantique des jeux

Précurseurs :

- Lorenzen [Lor61],
- Conway, Joyal [Joy77].

Des modèles complets de (fragments de) logique linéaire :

- Abramsky, Jagadeesan [AJ94],
- Hyland, Ong [HO93].

Des modèles pleinement adéquats de PCF :

- Abramsky, Jagadeesan, Malacaria [AJM00],
- Hyland, Ong [HO00, Nic94].

Comment donner une description abstraite de
ce que calculent les programmes ?

Sémantique

sémantique	$=$	interprétation mathématique abstraite des programmes / des preuves

Sémantique dénotationnelle

sémantique	=	interprétation mathématique abstraite des programmes / des preuves

sémantique dénotationnelle	=	invariants du calcul

Modèle

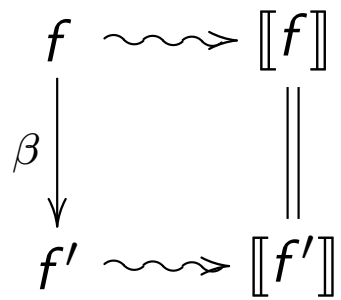
Modèle d'un langage de programmation :

- un type A est interprété par un « espace de calcul » $[\![A]\!]$,
- un programme $f : A \longrightarrow B$ est interprété par une « fonction » $[\![f]\!] : [\![A]\!] \longrightarrow [\![B]\!]$.

Modèle dénotationnel

Modèle d'un langage de programmation :

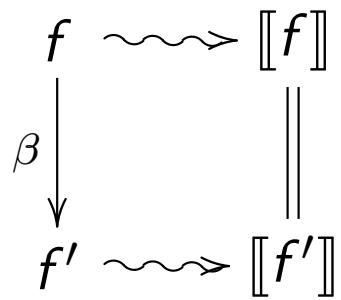
- un type A est interprété par un « espace de calcul » $[\![A]\!]$,
- un programme $f : A \longrightarrow B$ est interprété par une « fonction » $[\![f]\!] : [\![A]\!] \longrightarrow [\![B]\!]$.



Modèle dénotationnel

Modèle d'un langage de programmation :

- un type A est interprété par un « espace de calcul » $[\![A]\!]$,
- un programme $f : A \longrightarrow B$ est interprété par une « fonction » $[\![f]\!] : [\![A]\!] \longrightarrow [\![B]\!]$.



Sémantiques de jeux :

- les types sont interprétés par des **jeux**,
- les programmes sont interprétés par des **stratégies**.

Sémantiques de jeux

Les sémantiques de jeux modélisent *le comportement interactif* des programmes.

$$\texttt{fun}\ x \longrightarrow \texttt{not}\ x \quad : \quad \texttt{int} \Rightarrow \texttt{int}$$

$$\begin{aligned} (\texttt{fun}\ x \longrightarrow \texttt{not}\ x)\texttt{true} &\quad \rightsquigarrow \quad \texttt{false} \\ (\texttt{fun}\ x \longrightarrow \texttt{not}\ x)\texttt{false} &\quad \rightsquigarrow \quad \texttt{true} \end{aligned}$$

Le langage PCF

PCF est un langage

- fonctionnel,
- avec des valeurs de base : booléens et entiers,
- avec des constructions pour manipuler ces valeurs : `if then else`, `+`, etc.
- avec une construction de point fixe,
- simplement typé.

Un modèle pleinement adéquat de PCF

Deux termes M et N sont **extensionnellement équivalents** ($M \approx N$) lorsque pour tout contexte $C[-]$,

$$C[M] \Downarrow v \qquad \text{ssi} \qquad C[N] \Downarrow v$$

Un modèle pleinement adéquat de PCF

Deux termes M et N sont **extensionnellement équivalents** ($M \approx N$) lorsque pour tout contexte $C[-]$,

$$C[M] \Downarrow v \qquad \text{ssi} \qquad C[N] \Downarrow v$$

Modèle **pleinement adéquat** de PCF :

$$M \approx N \qquad \text{ssi} \qquad [\![M]\!] = [\![N]\!]$$

Définissabilité

Un élément $f : [\![A]\!]$ est **définissable** lorsqu'il existe $M : A$ tel que

$$f \quad = \quad [\![M]\!]$$

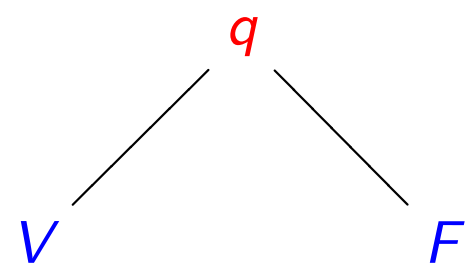
Une arène

$$(M, \vdash, \lambda, \lambda')$$

est constituée

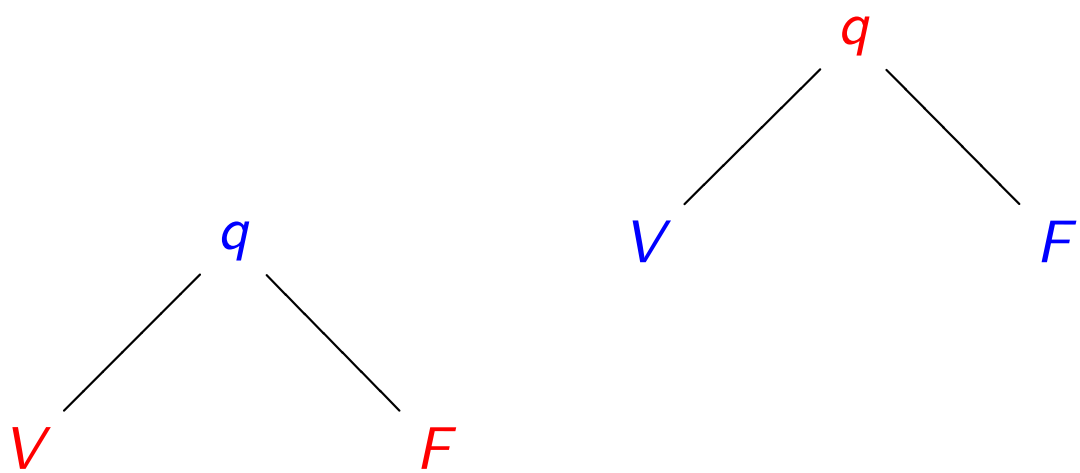
- d'un ensemble de **coups** M
- d'une relation binaire $\vdash$ exprimant les **dépendances causales**
- d'une **polarisation** des coups $\lambda : M \to \{O, P\}$
- d'une fonction $\lambda' : M \to \{Q, A\}$

$\mathbb{B}$ =

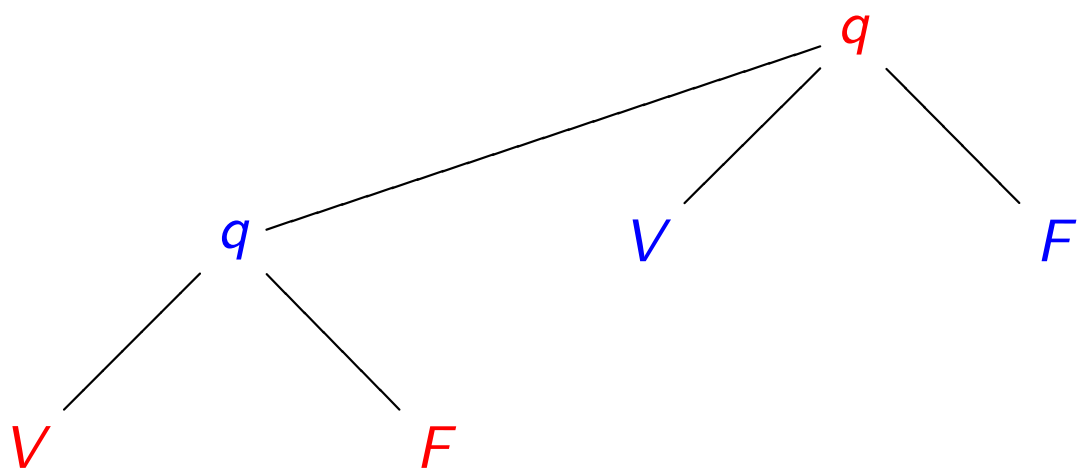


Le jeu $\mathbb{B} \times \mathbb{B}$:

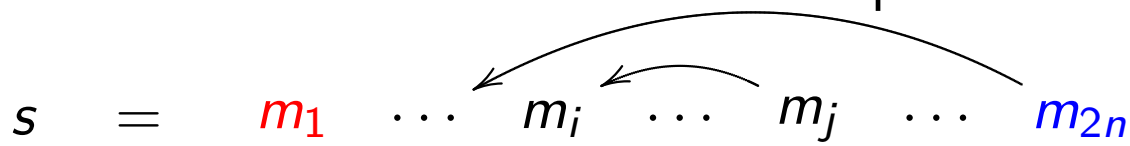
Le jeu $\mathbb{B} \Rightarrow \mathbb{B}$:



Le jeu $\mathbb{B} \Rightarrow \mathbb{B}$:

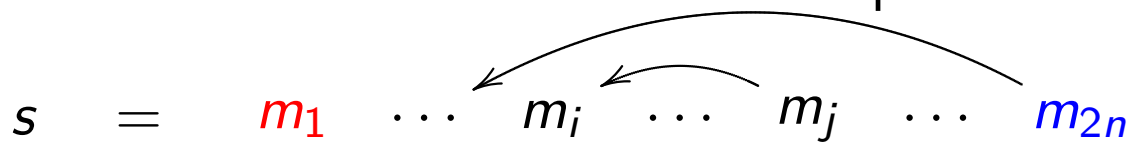


Une **partie** s dans une arène A est une suite pointée de coups



- alternée,
- de longueur paire,
- telle que si m_j pointe sur m_i alors $m_i \vdash_A m_j$.

Une **partie** s dans une arène A est une suite pointée de coups



- alternée,
- de longueur paire,
- telle que si m_j pointe sur m_i alors $m_i \vdash_A m_j$.

Une **stratégie** est un ensemble non vide de parties, clos par préfixe de longueur paire.

Sémantiques de jeux HO

La stratégie `not` :

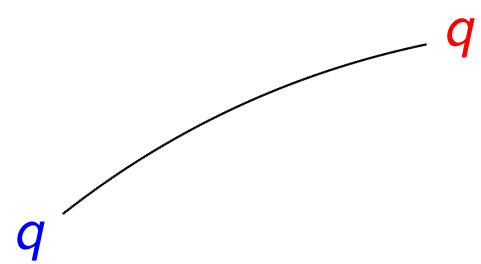
$$\mathbb{B} \qquad \Rightarrow \qquad \mathbb{B}$$

$$\{\ \varepsilon\ \}$$

Sémantiques de jeux HO

La stratégie `not` :

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

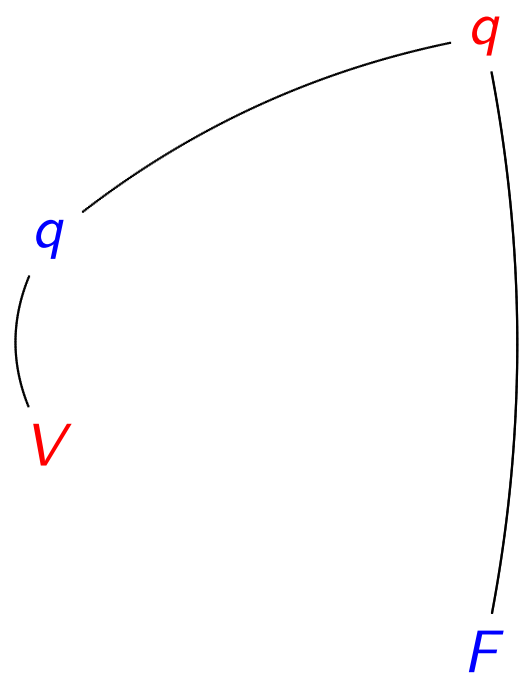


$$\{\, \varepsilon, \qquad q \cdot q \,\}$$

Sémantiques de jeux HO

La stratégie `not` :

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

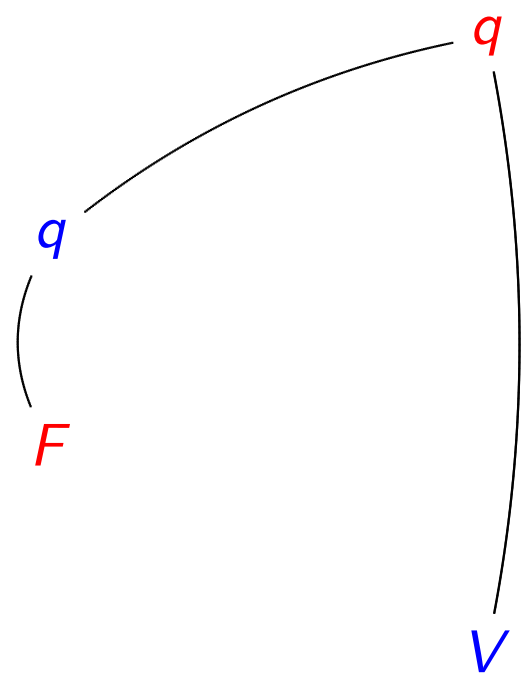


$$\{\ \varepsilon, \qquad q \cdot q, \qquad q \cdot q \cdot V \cdot F\ \}$$

Sémantiques de jeux HO

La stratégie `not` :

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

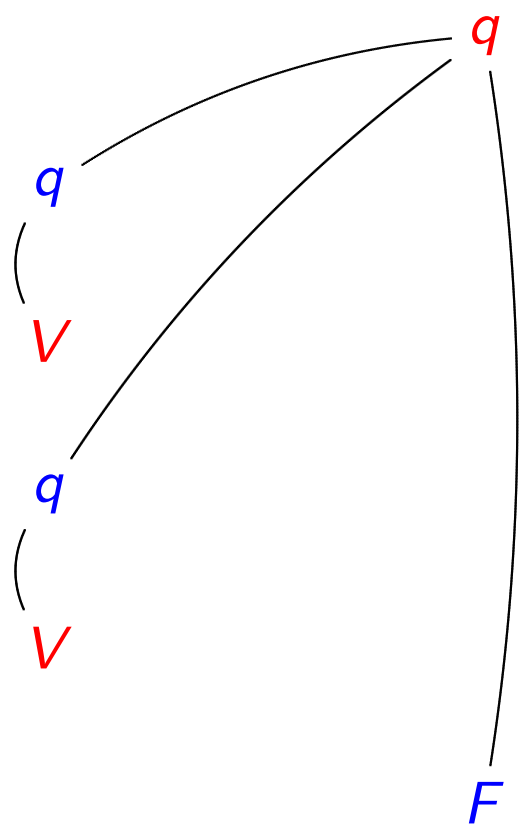


$$\{\, \varepsilon, \qquad q \cdot q, \qquad q \cdot q \cdot V \cdot F, \qquad q \cdot q \cdot F \cdot V \,\}$$

Sémantiques de jeux HO

$$\texttt{fun } x \longrightarrow \texttt{if } x \texttt{ then not } x \texttt{ else not } x$$

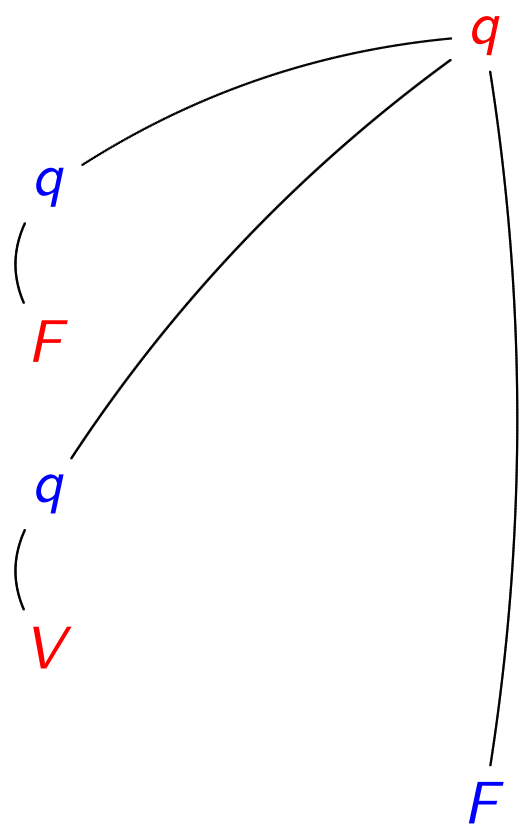
$$\mathbb{B} \qquad \Rightarrow \qquad \mathbb{B}$$



Sémantiques de jeux HO

$$\texttt{fun } x \longrightarrow \texttt{if } x \texttt{ then not } x \texttt{ else not } x$$

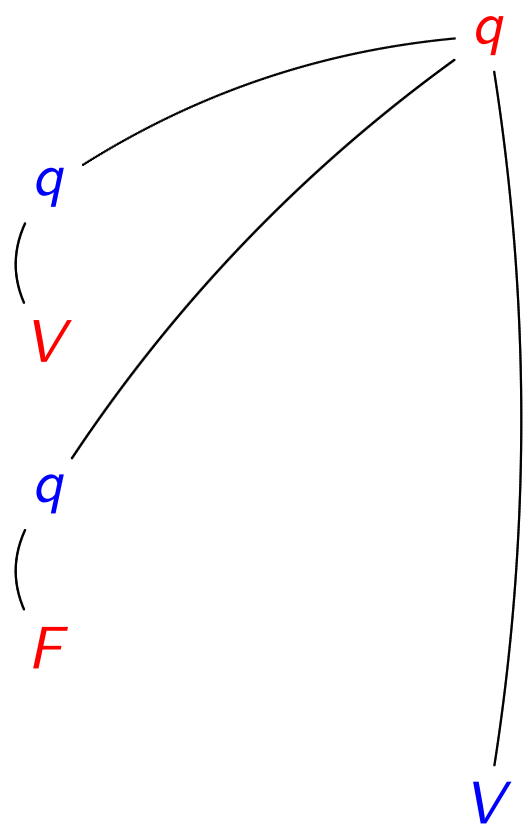
$$\mathbb{B} \qquad \Rightarrow \qquad \mathbb{B}$$



Sémantiques de jeux HO

$$\texttt{fun } x \longrightarrow \texttt{if } x \texttt{ then not } x \texttt{ else not } x$$

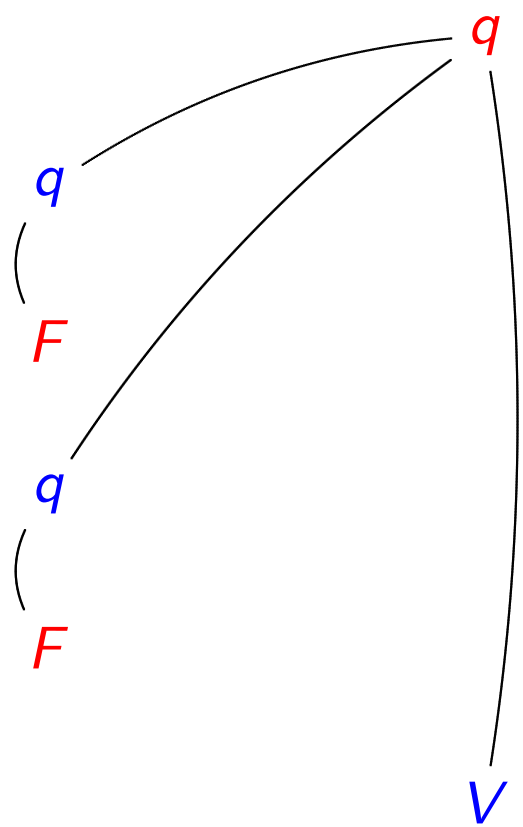
$$\mathbb{B} \qquad \Rightarrow \qquad \mathbb{B}$$



Sémantiques de jeux HO

`fun` $x \longrightarrow$ `if` x `then not` x `else not` x

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



Catégories de jeux

Les stratégies ne cherchent pas à gagner.

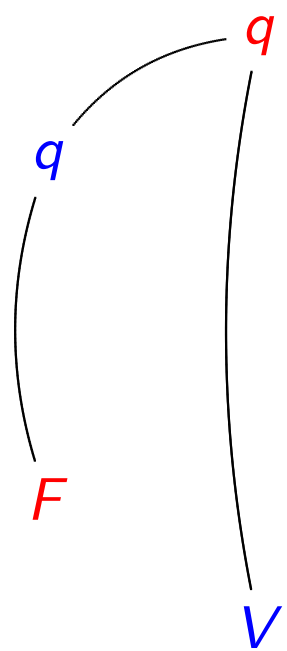
En revanche, on veut avoir une notion de *composition* qui reflète celle du langage modélisé.

Composition

Composition parallèle

Composition de `not` avec `not` :

$\mathbb{B} \longrightarrow \mathbb{B}$ $\qquad$ $\mathbb{B} \longrightarrow \mathbb{B}$

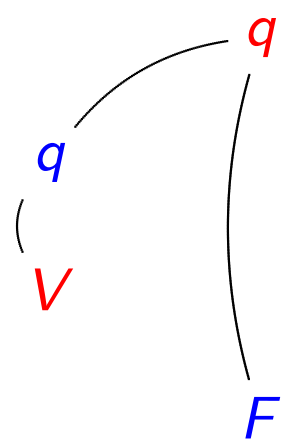


Composition

Composition parallèle

Composition de `not` avec `not` :

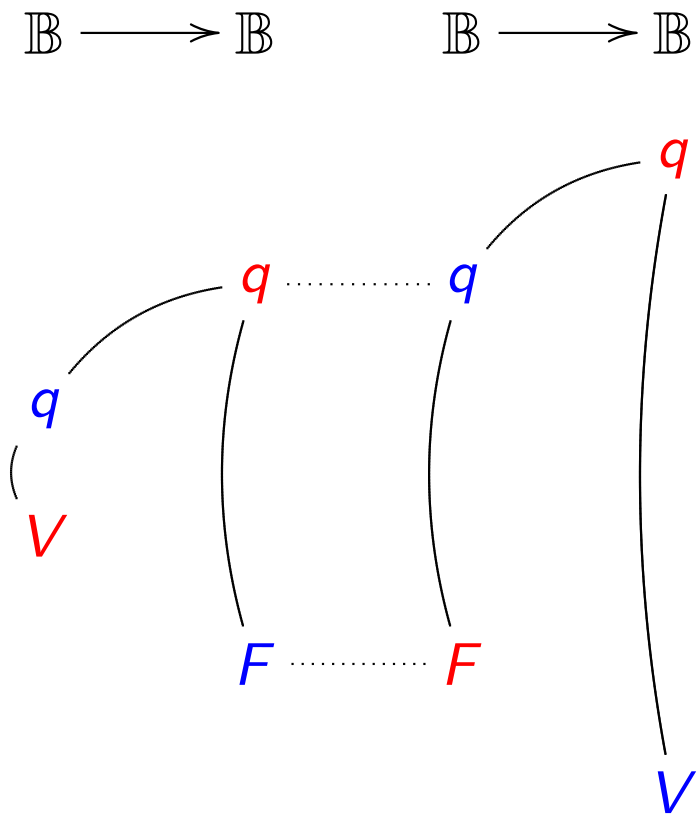
$$\mathbb{B} \longrightarrow \mathbb{B} \qquad \mathbb{B} \longrightarrow \mathbb{B}$$



Composition

Composition parallèle

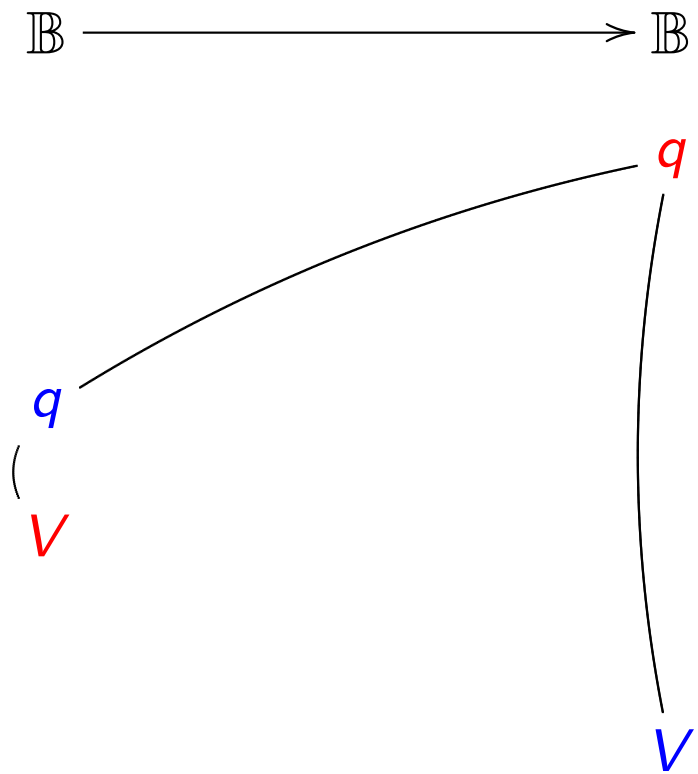
Composition de `not` avec `not` :



Composition

Composition parallèle + masquage.

Composition de `not` avec `not` :

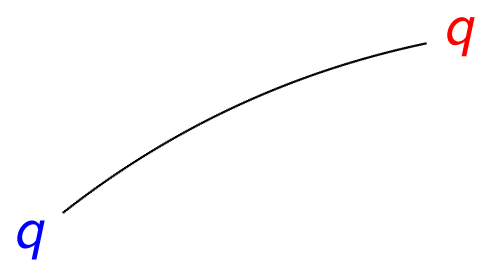


Stratégie identité (copycat)

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

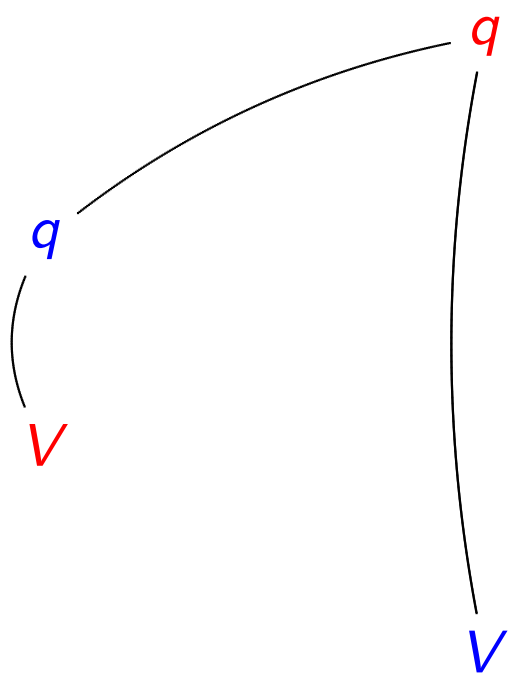
Stratégie identité (copycat)

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



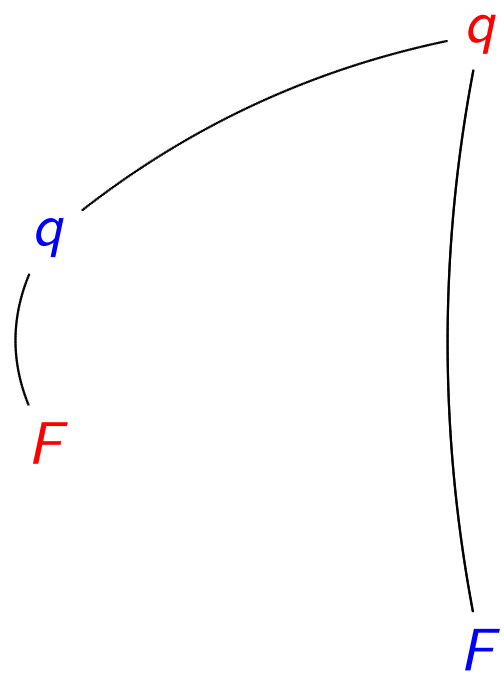
Stratégie identité (copycat)

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



Stratégie identité (copycat)

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



Comment caractériser les stratégies définissables par des termes de PCF ?

Déterminisme

Définition

Une stratégie $\sigma : A$ est **déterministe** lorsque

$$s \cdot m \in \sigma \quad \text{et} \quad s \cdot m' \in \sigma \quad \text{implique} \quad m = m'$$

Déterminisme

Définition

Une stratégie $\sigma : A$ est **déterministe** lorsque

$$s \cdot m \in \sigma \quad \text{et} \quad s \cdot m' \in \sigma \quad \text{implique} \quad m = m'$$

Contre-exemple : la stratégie $\sigma : \mathbb{B}$ définie par

$$\sigma \quad = \quad \{ \varepsilon, q \cdot F, q \cdot V \}$$

Innocence

Définition

Une stratégie $\sigma : A$ est **innocente** lorsqu'elle joue en n'ayant qu'une mémoire partielle de son passé : sa *vue*.

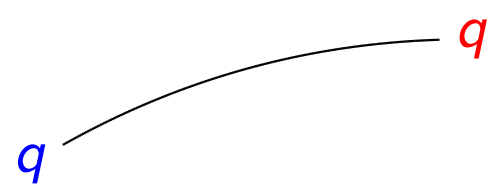
$$s \cdot m \in \sigma \quad \text{et} \quad \ulcorner s \urcorner = \ulcorner t \urcorner \quad \text{implique} \quad t \cdot m \in \sigma$$

Innocence

1er contre-exemple

$$\texttt{fun}\ f \longrightarrow \cdots f\ \cdots$$

$$(\mathbb{B} \Rightarrow \mathbb{B}) \Rightarrow \mathbb{B}$$

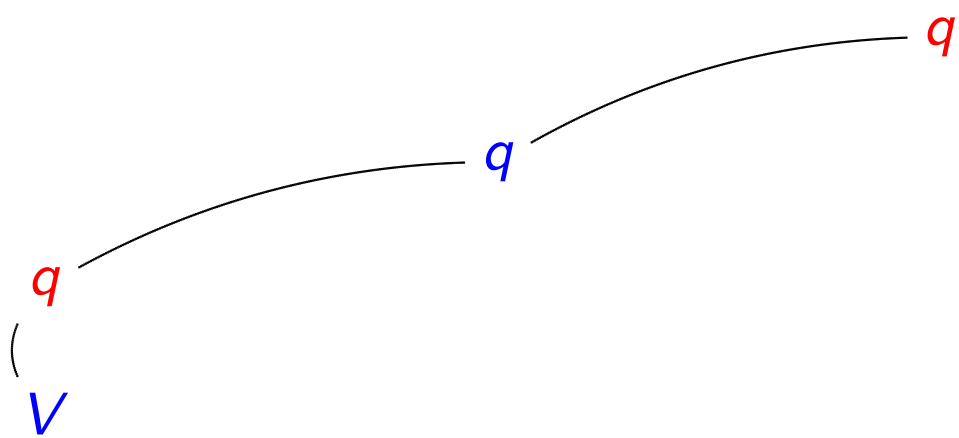


Innocence

1^{er} contre-exemple

$$\texttt{fun } f \to \cdots f\ \texttt{true} \cdots$$

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

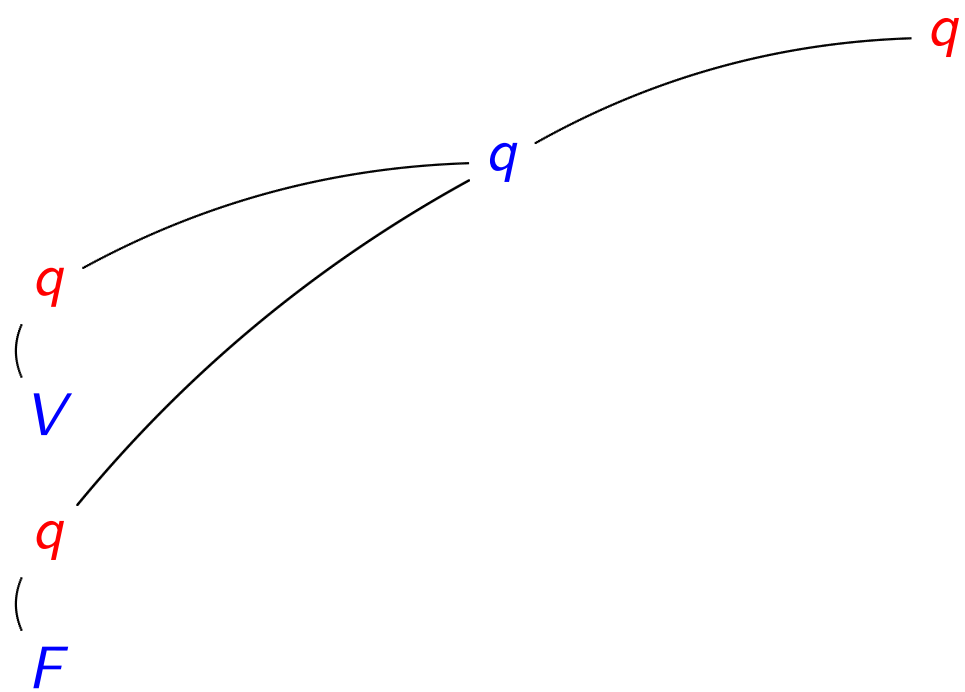


Innocence

1er contre-exemple

$$\texttt{fun}\ f \longrightarrow \cdots f\ \texttt{?} \cdots$$

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

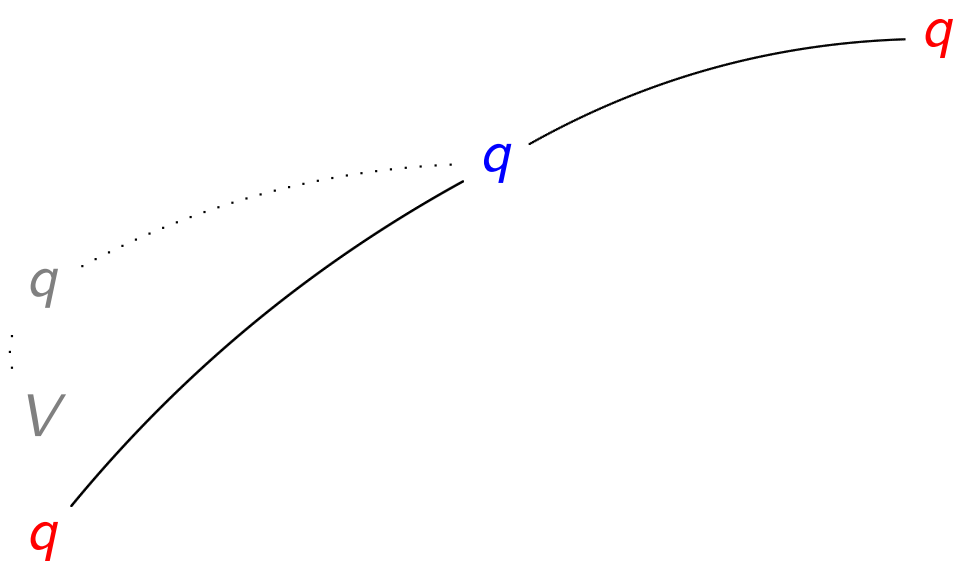


Innocence

1er contre-exemple

$$\texttt{fun}\ f \longrightarrow \cdots f\ ? \cdots$$

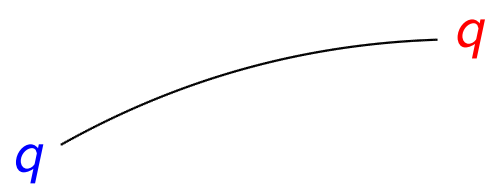
$$(\mathbb{B} \quad \Rightarrow \quad \mathbb{B}) \quad \Rightarrow \quad \mathbb{B}$$



Innocence

2e contre-exemple

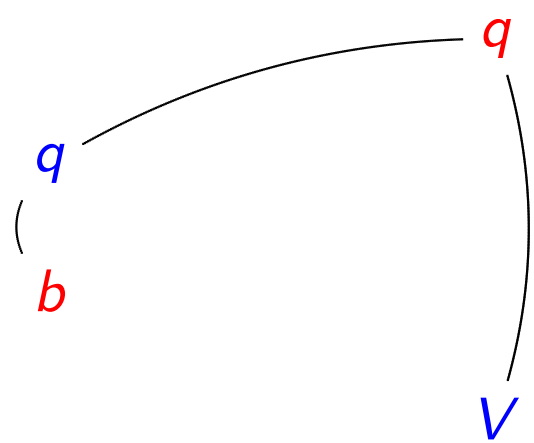
$$(\mathbb{B} \Rightarrow \mathbb{B}) \Rightarrow \mathbb{B}$$



Innocence

2e contre-exemple

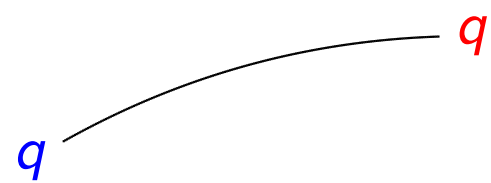
$$(\mathbb{B} \quad \Rightarrow \quad \mathbb{B}) \quad \Rightarrow \quad \mathbb{B}$$



Innocence

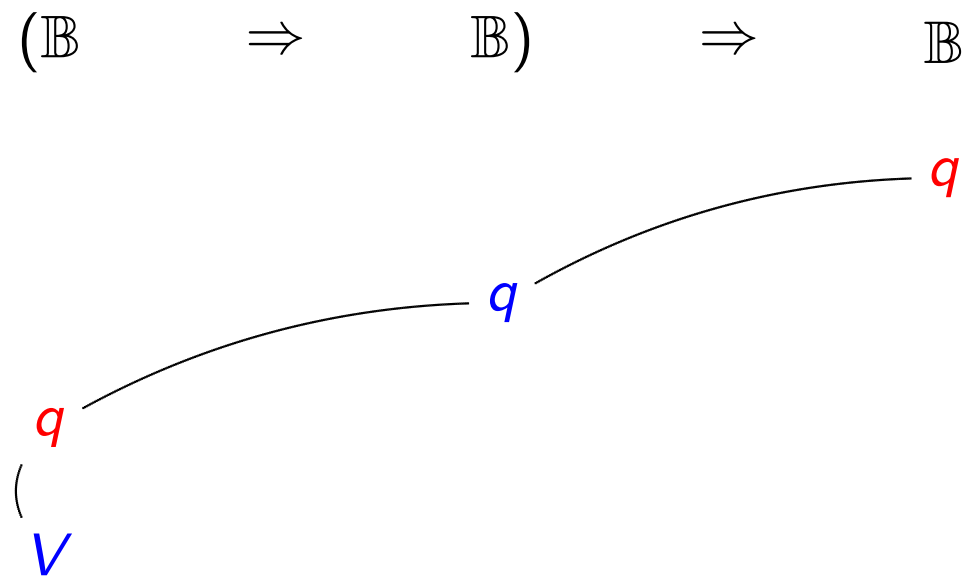
2e contre-exemple

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



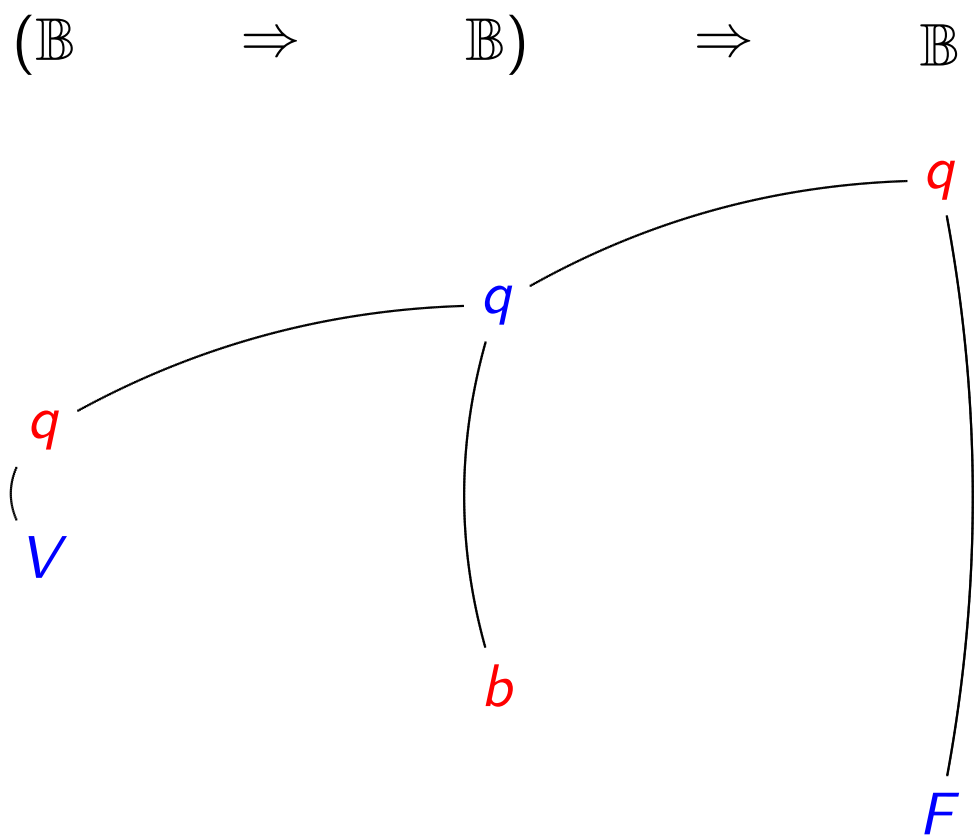
Innocence

2e contre-exemple



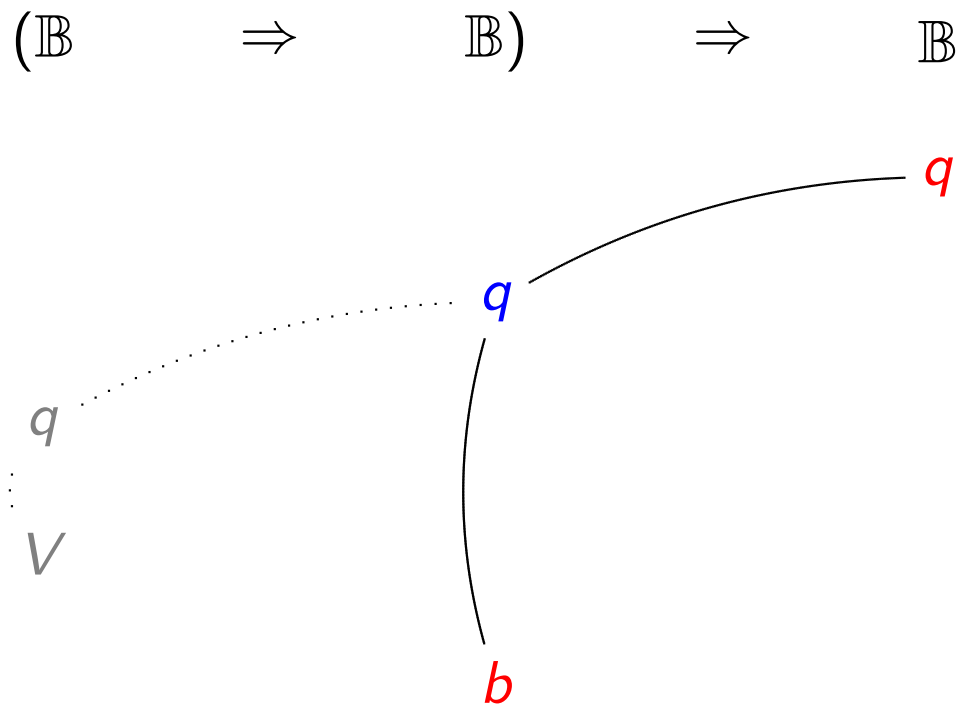
Innocence

2e contre-exemple



Innocence

2e contre-exemple



Innocence

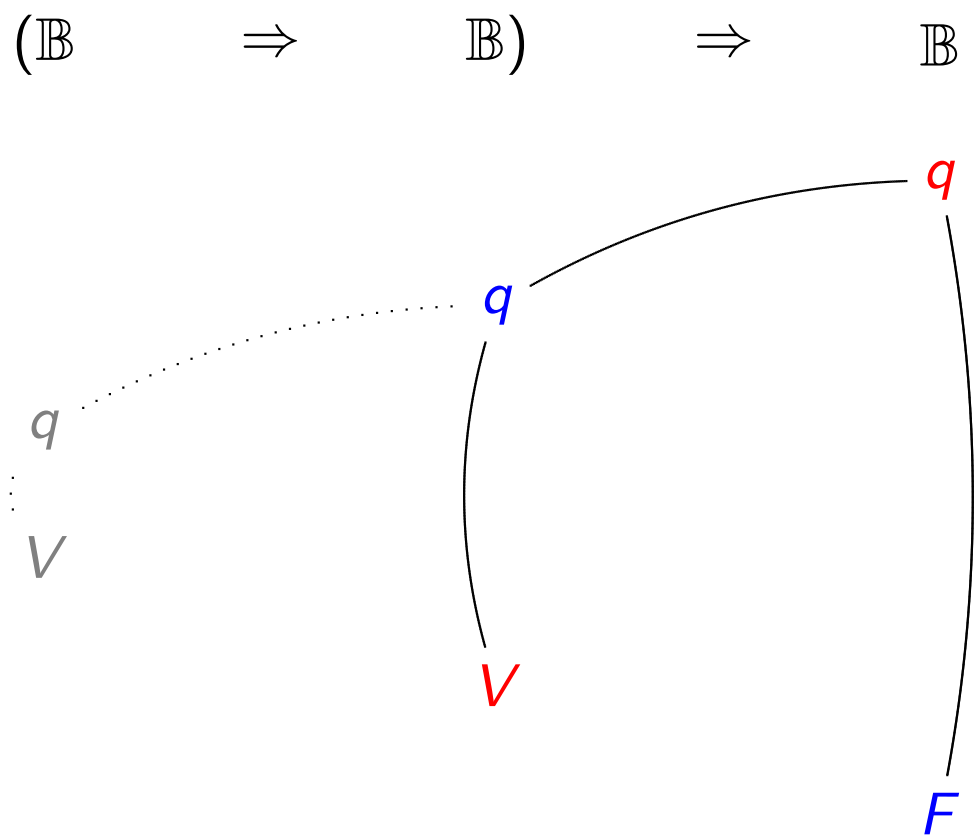
Définition

Pour toute partie s, la **vue** $\ulcorner s \urcorner$ de s est définie inductivement par

$$\begin{aligned}
\ulcorner s \cdot m \urcorner &= \ulcorner s \urcorner \cdot m && \text{si } m \text{ est un coup Joueur} \\
\ulcorner s \cdot m \cdot t \cdot n \urcorner &= \ulcorner s \cdot m \urcorner \cdot n && \text{si } n \text{ est un coup Opposant pointant sur } m \\
\ulcorner s \cdot m \urcorner &= m && \text{si } m \text{ est un coup Opposant initial}
\end{aligned}$$

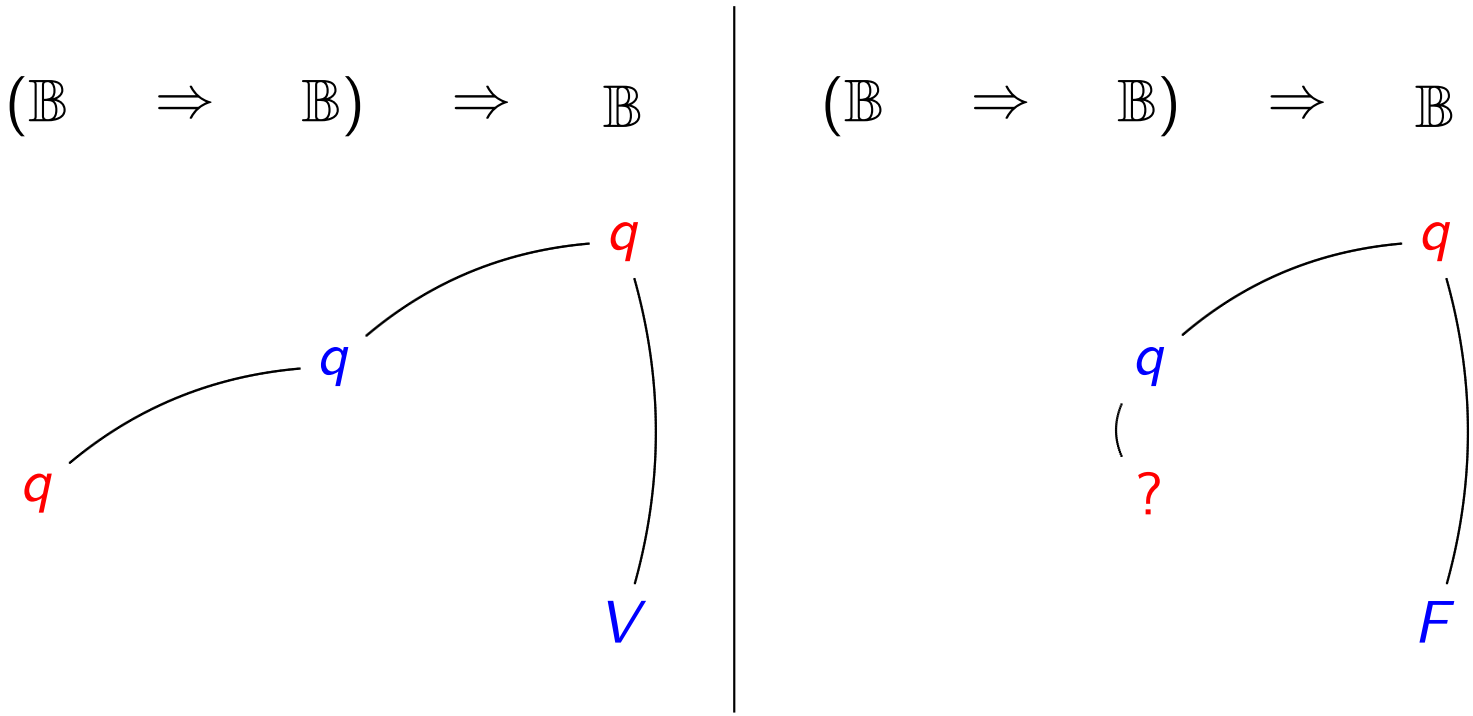
Innocence

2e contre-exemple



Catch

La stratégie `catch` :



Bon parenthésage

Définition

Une stratégie $\sigma : A$ est **bien parenthésée** si toute réponse m du Joueur dans une partie $s \cdot m$ pointe vers la question pendante de $\ulcorner s \urcorner$.

Théorème (HON 94)

Le modèle des stratégies déterministes, innocentes et bien parenthésées est pleinement adéquat pour PCF.

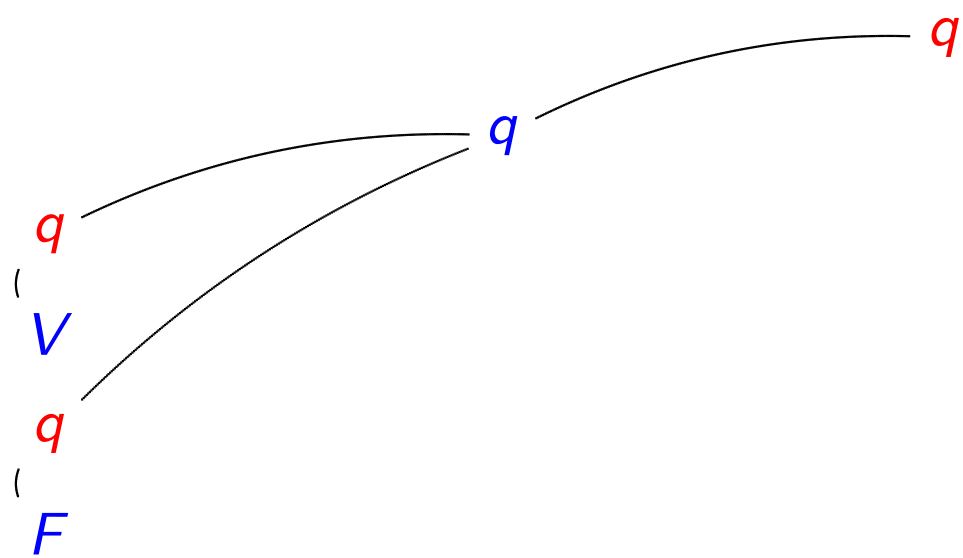
Parenthésage et innocence

Stratégies
visibles + déterministes + bien parenthésées + innocentes
$\approx$
programme PCF

Parenthésage et innocence

Stratégies
visibles + déterministes + bien parenthésées
$\approx$
programme PCF + références

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



$\texttt{fun}\ f \rightarrow x\ \texttt{:=}\ 0\,\texttt{;}\ \cdots\ f\ (\texttt{incr}\ x\,\texttt{;}\ x == 1)\ \cdots$

Parenthésage et innocence

Stratégies
visibles + déterministes + bien parenthésées + innocentes
$\approx$
programme PCF

Parenthésage et innocence

Stratégies

visibles + déterministes + bien parenthésées + innocentes

$\approx$

λ-termes simplement typés = preuve intuitionniste

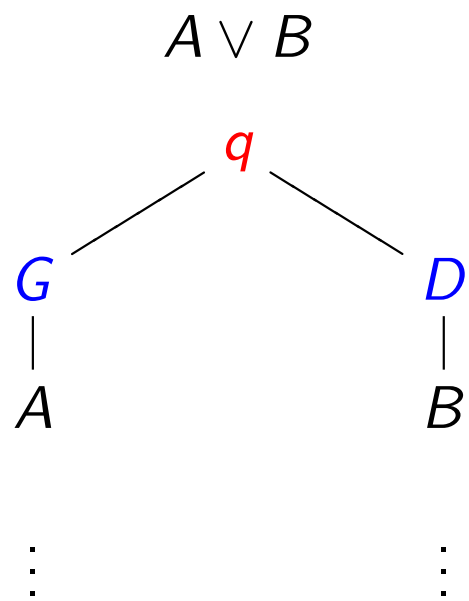
Parenthésage et innocence

Stratégies

visibles + déterministes + innocentes

$\approx$

$\lambda\mu$-termes simplement typés = preuve classique



Parenthésage et innocence

Stratégies

visibles + déterministes + innocentes

$\approx$

$\lambda\mu$-termes simplement typés = preuve classique

$\perp$

$q_\perp$

Parenthésage et innocence

Stratégies

visibles + déterministes + innocentes

$\approx$

$\lambda\mu$-termes simplement typés = preuve classique

$$\neg A \quad = \quad A \Rightarrow \bot$$

$q_\bot$

|

A^*

$\vdots$

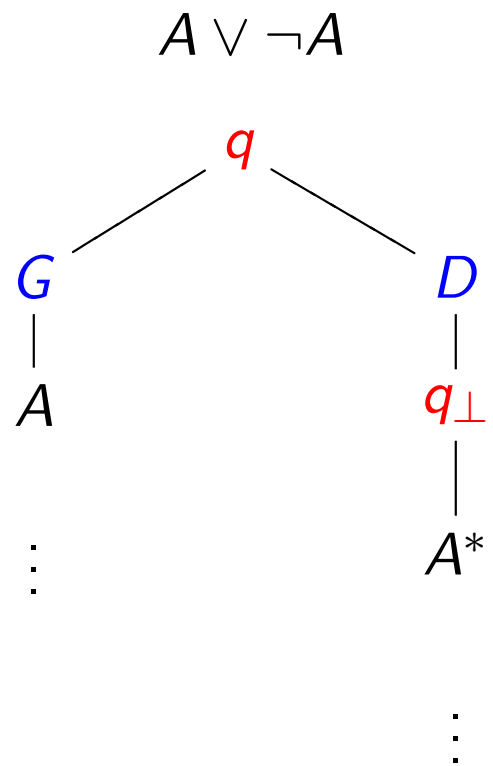
Parenthésage et innocence

Stratégies

visibles + déterministes + innocentes

$\approx$

$\lambda\mu$-termes simplement typés = preuve classique



Parenthésage et innocence

Stratégies

visibles + déterministes + innocentes

$\approx$

programme PCF + contrôle

Parenthésage et innocence

Stratégies

visibles + déterministes

$\approx$

programme PCF + contrôle + références

Applications

- permet de mieux comprendre le comportement interactif des programmes et des preuves,

Applications

- permet de mieux comprendre le comportement interactif des programmes et des preuves,
- model-checking et interprétation abstraite :
  [GM03] lorsque les types sont au plus du second ordre,
  - les pointeurs ne sont pas nécessaires,
  - les stratégies peuvent être décrites par des langages réguliers,
  - l'équivalence est décidable sur un alphabet fini...

Merci !

Bibliographie I


S. Abramsky and R. Jagadeesan.
Games and Full Completeness for Multiplicative Linear Logic.
The Journal of Symbolic Logic, 59(2) :543–574, 1994.


S. Abramsky, R. Jagadeesan, and P. Malacaria.
Full abstraction for PCF.
Information and Computation, 163(2) :409–470, 2000.


D.R. Ghica and G. McCusker.
The regular-language semantics of second-order idealized ALGOL.
Theoretical Computer Science, 309(1-3) :469–502, 2003.


J.M.E. Hyland and C.H.L. Ong.
Fair games and full completeness for multiplicative linear logic without the mix-rule.
preprint, 190, 1993.

Bibliographie II


M. Hyland and L. Ong.
On Full Abstraction for PCF : I, II and III.
Information and Computation, 163(2) :285–408, December 2000.


A. Joyal.
Remarques sur la théorie des jeux à deux personnes.
Gazette des Sciences Mathématiques du Quebec, 1(4) :46–52, 1977.


P. Lorenzen.
Ein dialogisches Konstruktivitatskriterium.
Infinitistic Methods, pages 193–200, 1961.

Bibliographie III


H. Nickau.
Hereditarily sequential functionals.
In A. Nerode and Yu. V. Matiyasevich, editors, *Proceedings of the Symposium on Logical Foundations of Computer Science : Logic at St. Petersburg*, volume 813 of *Lecture Notes in Computer Science*, pages 253–264. Springer Verlag, 1994.