Sémantique des jeux : de la programmation à la logique

GDR IM

Olivier LAURENT

Olivier.Laurent@pps.jussieu.fr

Preuves Programmes Systèmes
CNRS – Université Paris VII

Groupes GEOCAL et LAC

Sémantique des jeux

Un mélange de

- interprétation interactive de la logique
- sémantique dénotationnelle

qui permet de capturer (précisément)

- langage fonctionnel
- opérateurs de contrôle
- traits impératifs
- non-déterminisme
- aspects probabilistes
- concurrence
- ...

Modèles de jeux

Étude des traces d'exécution d'un programme

- deux joueurs O et J
- trace = partie = suite de coups
- programme = stratégie pour J
- modularité de l'interprétation = composition des stratégies

conditions sur les stratégies :

- déterminisme, filarité, visibilité, finitude, totalité,
- innocence, bon parenthésage, rigidité,
- ...

correspondant à des primitives de programmation précises

Fonction

$$\text{bool} \quad \longrightarrow \quad \text{bool}$$

q

q

t

f

`fun` b `->` `if` b `then` false
`else` true

Fonction constante

$$\text{bool} \longrightarrow \text{bool}$$

q

t

`fun` b `->` true

Fonction à plusieurs arguments

$$\text{bool} \quad \longrightarrow \quad \text{bool} \quad \longrightarrow \quad \text{bool}$$

		q
q		
t		
	q	
	t	
		t

$\texttt{fun}\ b_1\ b_2\ \texttt{->}\ \texttt{if}\ b_1\ \texttt{then}\ \texttt{if}\ b_2\ \texttt{then}\ \text{true}$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\texttt{else}\ \text{false}$
$\qquad\qquad\qquad\qquad\texttt{else}\ \texttt{if}\ b_2\ \texttt{then}\ \text{true}$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\texttt{else}\ \text{false}$

Fonction à plusieurs accès à l'argument

bool $\longrightarrow$ bool

q

q

t

q

t

t

`fun` b `->` `if` b `then` `if` b `then` true
`else` false
`else` `if` b `then` true
`else` false

Fonctionnelle

$$(\quad \text{bool} \quad \longrightarrow \quad \text{bool} \quad) \quad \longrightarrow \quad \text{bool}$$

		q
	q	
q		
t		
	f	
		f

fun f -> f true

Fonctionnelle (bis)

(bool	→ bool)	→ bool	→ bool
			q
	q		
q			
		q	
		t	
t			
	f		
			f

`fun` $f\ b$ `->` $f\ b$

Fonctionnelle non fonctionnelle (1)

$$(\ \text{bool} \ \rightarrow \ \text{bool} \ ) \ \rightarrow \ \text{bool}$$

q

q

q

t

`fun` f `->` `try` `if` f (`raise` α) `then` false
`else` false

`with` $\alpha \mapsto$ true

Fonctionnelle non fonctionnelle (2)

$$(\quad \text{bool} \quad \rightarrow \quad \text{bool} \quad) \quad \rightarrow \quad \text{bool}$$

q

q

t

f

`fun` f `-> try if` f (`raise` α) `then` false
`else` false
`with` $\alpha \mapsto$ true

Fonctionnelle non fonctionnelle (3)

$$(\text{ bool } \rightarrow \text{ bool }) \rightarrow \text{ bool}$$

q

q

f

f

`fun` f `-> try if` f (`raise` α) `then` false
`else` false
`with` $\alpha \mapsto$ true

Composition (1)

(	bool	$\rightarrow$	bool	)		(	bool	$\rightarrow$	bool	)	$\rightarrow$	bool
												q
			q						q			
			t						t			
												f

try if (fun b -> true) (raise α)
then false
else false
with $\alpha \mapsto$ true

Composition (1)

$$(\;\text{bool} \;\to\; \text{bool}\;) \qquad (\;\text{bool} \;\to\; \text{bool}\;) \;\to\; \text{bool}$$

q

q q

t t

f

$\texttt{try if}\ (\texttt{fun}\ b\ \texttt{->}\ \text{true})\ (\texttt{raise}\ \alpha)$
$\quad\texttt{then}\ \text{false}$
$\quad\texttt{else}\ \text{false}$
$\texttt{with}\ \alpha \mapsto \text{true}$

Composition (2)

(	bool	→	bool	)		(	bool	→	bool	)	→	bool
												q
			q						q			
	q						q					
												t

```
try if (fun b -> if b then false else true) (raise α)
       then false
       else false
with α ↦ true
```

Composition (2)

$$(\quad \text{bool} \quad \rightarrow \quad \text{bool} \quad) \qquad\qquad (\quad \text{bool} \quad \rightarrow \quad \text{bool} \quad) \quad \rightarrow \quad \text{bool}$$

q

q q

q q

t

`try if` (`fun` b `->` `if` b `then` false `else` true) (`raise` α)
`then` false
`else` false
`with` $\alpha \mapsto$ true

Fonctionnelle impérative

$$
\begin{array}{ccccccccc}
(& \mathsf{bool} & \rightarrow & \mathsf{bool} &) & \rightarrow & \mathsf{bool} \\
 & & & & & & \color{red}{\mathsf{q}} \\
 & & & \color{blue}{\mathsf{q}} & & & \\
 & \color{red}{\mathsf{q}} & & & & & \\
 & \color{blue}{\mathsf{t}} & & & & & \\
 & & & \color{red}{\mathsf{f}} & & & \\
 & & & & & & \color{blue}{\mathsf{t}} \\
\hline
 & & & & & & \color{red}{\mathsf{q}} \\
 & & & \color{blue}{\mathsf{q}} & & & \\
 & & & \color{red}{\mathsf{f}} & & & \\
 & & & & & & \color{blue}{\mathsf{f}}
\end{array}
$$

Fonctionnelle impérative

$$(\ \text{bool} \ \rightarrow \ \text{bool}\ ) \ \rightarrow \ \text{bool}$$

		q
	q	
q		
t		
	f	
		t

```
fun f -> let r = ref false in
         if f (r := true ; true) then !r
                                 else !r
```

Compteur

(bool	→ bool)	→ nat
		q
	q	
q		
t		
q		
t		
q		
t		
	f	
		3

Compteur

$$(\quad \text{bool} \quad \longrightarrow \quad \text{bool} \quad) \quad \longrightarrow \quad \text{nat}$$

q

q

q

t

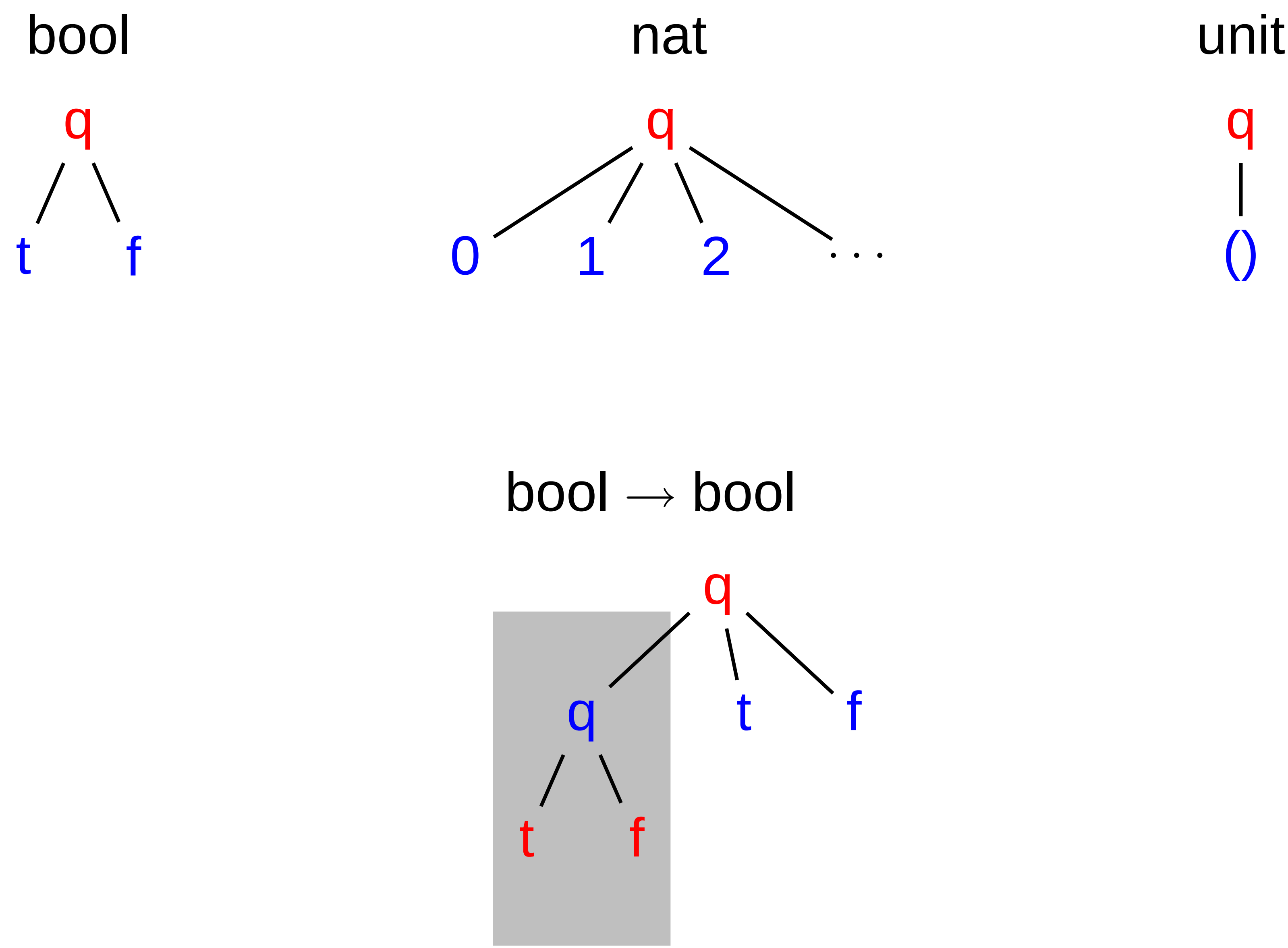
$\vdots$

f

n

`fun` f `->` `let` c `=` `ref` 0 `in`
`if` $f\ (\ c := !c + 1\ ;\ \text{true}\ )$ `then` $!c$
`else` $!c$

Type = arène



Constructeurs de type de base

bool	nat	unit
q	q	q
t	0	()
q	q	
f	1	
	⋮	
true	0	()
false	1	
	⋮	

Destructeurs de type de base

bool	$\rightarrow$	T	$\rightarrow$	T	$\rightarrow$	T
						q
q						
f						
				q		
				•		
						•
					$\vdots$	

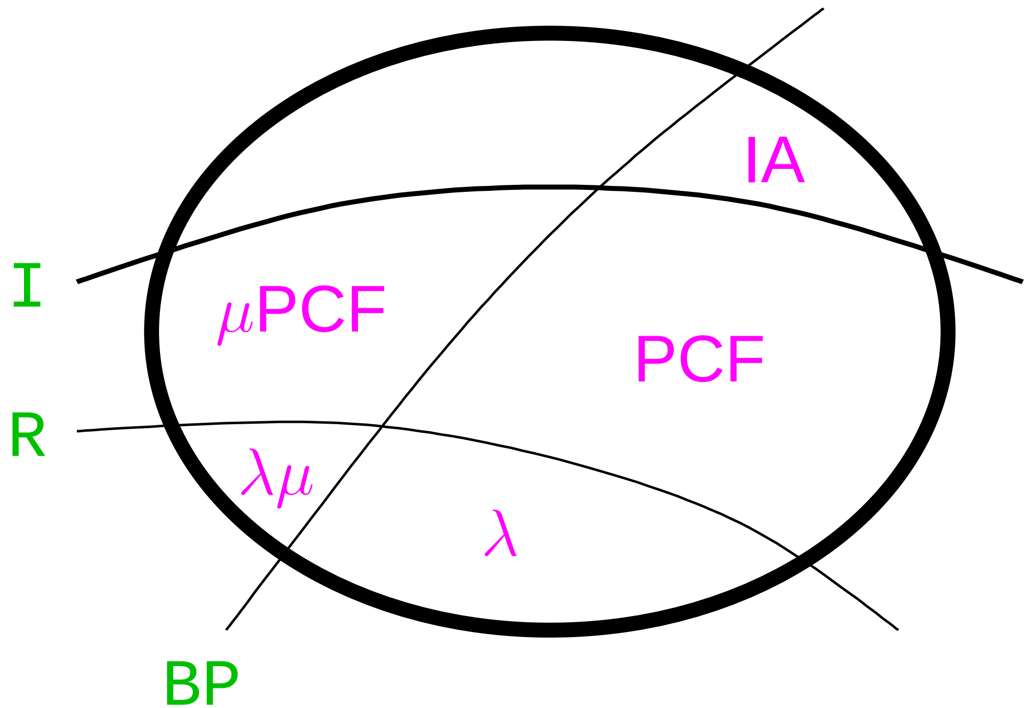
unit	$\rightarrow$	T	$\rightarrow$	T
				q
q				
()				
		q		
		•		
				•
			$\vdots$	

`fun` $b\ t_1\ t_2$ `->` `if` b `then` t_1 `else` t_2

`fun` $u\ t$ `->` u `;` t

Contraintes sur les stratégies

- Innocence : tout coup d'O est fils du coup précédent
- Bon Parenthésage : J répond à la dernière question d'O
- Rigidité Avant : si O répond alors J répond
- Rigidité Arrière : J ne répond que si O vient de répondre



Logique propositionnelle

Formules

$$A \quad ::= \quad X \quad | \quad \bot \quad | \quad A \to A$$

Interprétation

$X \simeq$ unit

q

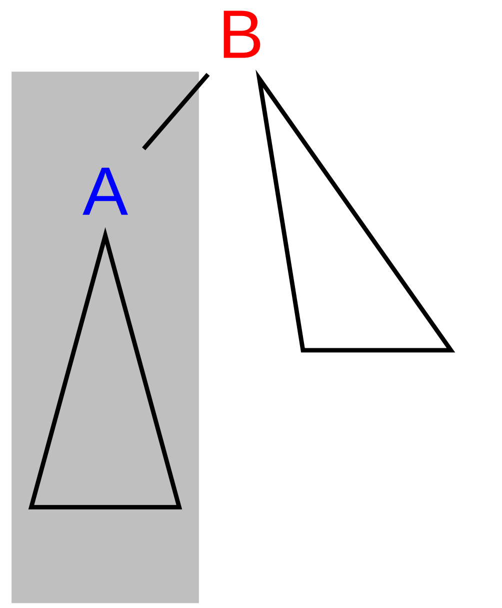
|

X

$\bot =$ void

q

$A \to B$



Système de Hilbert (1)

$$X \longrightarrow Y \longrightarrow X$$

q

q

X

X

Système de Hilbert (2)

$$
\begin{array}{ccccccccccccccccc}
(& X & \to & Y & \to & Z &) & \to & (& X & \to & Y &) & \to & X & \to & Z \\
 & & & & & & & & & & & & & & & & q \\
 & & & & & q & & & & & & & & & & & \\
 & & & q & & & & & & & & & & & & & \\
 & & & & & & & & & & & q & & & & & \\
 & & & & & & & & & q & & & & & & & \\
 & & & & & & & & & & & & & & q & & \\
 & & & & & & & & & & & & & & X & & \\
 & & & & & & & & & X & & & & & & &
\end{array}
$$

Système de Hilbert (3)

$$
\begin{array}{ccccccccccc}
(& (& \mathsf{X} & \rightarrow & \mathsf{Y} &) & \rightarrow & \mathsf{X} &) & \rightarrow & \mathsf{X} \\
 & & & & & & & & & & \mathsf{q} \\
 & & & & & & & \mathsf{q} & & & \\
 & & & & \mathsf{q} & & & & & & \\
 & & \mathsf{q} & & & & & & & & \\
 & & \mathsf{X} & & & & & & & & \\
 & & & & & & & & & & \mathsf{X}
\end{array}
$$

Système de Hilbert (4)

$$\perp \quad \rightarrow \quad \mathsf{X}$$

q

q

Logique du premier ordre

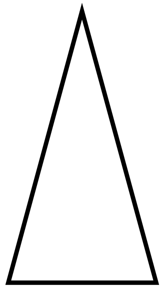
Formules

$$A \quad ::= \quad Xt_1 \ldots t_k \quad | \quad \bot \quad | \quad A \rightarrow A \quad | \quad \forall x A$$

Interprétation

$$\forall x \mathsf{A}$$

$$x.\ \ \mathsf{A}$$



Système de Hilbert (5)

$$\forall x \mathsf{X} x \quad \longrightarrow \quad \mathsf{X} t$$

$$\mathsf{q}\,[\mathsf{X}t]$$

$$t.\ \mathsf{q}\,[\mathsf{X}t]$$

$$\mathsf{X}$$

$$\mathsf{X}$$

Dernier exemple !

$$\forall x \quad (\quad \forall y \quad (\quad \mathsf{X}x \quad \to \quad \mathsf{X}y \quad) \quad \to \quad \bot \quad) \quad \to \quad \bot$$

$$\mathsf{q}$$

$$x_0 . \quad \mathsf{q}$$

$$y_0 . \quad \mathsf{q} \; [\mathsf{X}y_0]$$

$$y_0 . \quad \mathsf{q}$$

$$y_1 . \quad \mathsf{q} \; [\mathsf{X}y_1]$$

$$\mathsf{q} \; [\mathsf{X}y_0]$$

$$\mathsf{X}$$

$$\mathsf{X}$$

Correction et Complétude

Théorème

- si π est une preuve de A
  alors $[\![\pi]\!]$ est une stratégie gagnante sur l'arène A
- si σ est une stratégie gagnante sur l'arène A
  alors il existe une preuve π de A telle que $[\![\pi]\!] = \sigma$

stratégie gagnante = stratégie innocente rigide totale et finie