Réalisabilité concurrente

Emmanuel Beffara

PPS, Université Paris 7

LIPN, 12 décembre 2005

Logique et calcul

- Un peu d'histoire :

  1900 – Brouwer, Heyting, Kolmogorov : constructivisme.
  1930 – Kleene : réalisabilité, Church : λ-calcul.
  1950 – Kreisel : no-counterexample interpretation.
  1960 – isomorphisme de Curry-Howard.

- La correspondance :

logique intuitionniste	$\longleftrightarrow$	calcul fonctionnel	C-H
logique classique	$\longleftrightarrow$	calcul avec contrôle	Griffin'90
?	$\longleftrightarrow$	calcul concurrent	
logique linéaire	$\longleftrightarrow$	?	

Logique linéaire et concurrence

Interprétations interactives de la logique linéaire :

- interprétations dans CCS (Abramsky)
- réseaux de preuve en π-calcul (Bellin-Scott)
- modèles de jeux, ludique

Logiques pour la concurrence :

- logiques modales (Hennessy-Milner, pour CCS puis π)
- logiques spatiales (Caires-Cardelli)
- relevant logic, bunched implication, etc.

Réalisabilité à la Kleene

Une interprétation constructive de la logique intuitionniste :

- On se donne un ensemble M de témoins, muni d'une opération d'application.
- Une formule A est interprétée par l'ensemble $[A] \subseteq M$ des objets qui la réalisent :

$$[A \to B] = \{ x \mid \forall y \in [A], x\,y \in [B] \}$$

Deux interprétations :

- les objets interprètent les formules
- les formules décrivent les objets

Mais comment définir $x \Vdash A$ en général ?

Dualité en programmation

- Il y a toujours une dualité :

$$\mathsf{P} = \left\{ \begin{array}{rcl} \text{programme} & / & \text{données} \\ \text{joueur} & / & \text{adversaire} \\ \text{preuve} & / & \text{réfutation} \\ \text{processus} & / & \text{processus} \end{array} \right\} = \mathsf{N}$$

- On définit une observation : $\perp\!\!\!\perp \subseteq \mathsf{P} \times \mathsf{N}$.

$$A^{\perp\!\!\!\perp} = \{ x \mid \forall y \in A, y \perp\!\!\!\perp x \}$$

- Le bi-orthogonal $A^{\perp\!\!\!\perp\perp\!\!\!\perp}$ est une clôture, les ensembles clos sont des valeurs de vérité, ou comportements.
- $\{P\}^{\perp\!\!\!\perp} = \{Q\}^{\perp\!\!\!\perp}$ est une équivalence observationnelle.

Calcul concurrent

Par *concurrence* on désigne plusieurs aspects d'un système :

- un ensemble de processus qui s'exécutent en *parallèle*,
- une synchronisation par *communication*,
- un accès *concurrent* à des *ressources*.

On veut :

- un *langage* pour représenter ces systèmes,
  $\longrightarrow$ une variante du π-calcul ; *Milner'89*
- une *logique* pour les décrire,
  $\longrightarrow$ une variante de logique linéaire. *Girard'86*

Un π-calcul polarisé

- On construit des processus qui échangent des messages :

$S, T ::=$	$\bar{u}(\vec{x}).P \;/\; u(\vec{x}).P$	émission/réception de $\vec{x}$ sur u
	$S + T$	choix (gardé)
	$!S$	réplication (gardée)
$P, Q ::=$	S	processus séquentiel
	$P \mid Q$	composition parallèle
	$(\nu x)P$	restriction/création d'un nom x
	$u \to v$	routeur de u vers v

- La réduction correspond à la communication :

$$\bar{u}(x).P \mid u \to v \mid v(y).Q \longrightarrow (\nu x)(P \mid Q[x/y]) \mid u \to v$$

Tests et observations

Une **observation** est une propriété des systèmes clos.

- Soit $*$ une constante qui représente l'*acceptation*. *Girard'01*
- Soit $Acc = \{ P \mid \exists Q, P \equiv (* \mid Q) \}$: les états acceptants.
- On en déduit une forme d'observation :

may testing : $P \in \perp\!\!\!\perp$ si P peut atteindre un état de Acc

fair testing : $P \in \perp\!\!\!\perp$ si $P \rightarrow^* Q \Rightarrow Q \rightarrow^* Acc$

must testing : $P \in \perp\!\!\!\perp$ si toute exécution de P atteint Acc

L'orthogonalité est définie entre objets de même espèce :

$$P \perp\!\!\!\perp Q \qquad \text{si} \qquad P \mid Q \in \perp\!\!\!\perp$$

Interfaces

Des types de canaux imposent les arités et polarités :

Types de canaux : $s := \varepsilon(s_1 \dots s_n) \mid \cdots$

Interfaces : $I := x_1 : s_1, \dots, x_n : s_n$

Les règles de typage importantes :

$$\frac{P :: I, u : \varepsilon(s_1 \dots s_n), \{x_i : s_i\}}{u^{\varepsilon}(x_1 \dots x_n).P :: I, u : \varepsilon(s_1 \dots s_n)} \qquad \frac{P :: I, x_i : s, x_o : \bar{s}}{(\nu x)P[x/x_i,x_o] :: I}$$

Pour la composition de processus :

$$I \to J := \bar{I} \cup J$$

Logique des comportements – composition

La composition est une mise en parallèle et une restriction :

$$P :: I \to J, Q :: J \to K \quad \Rightarrow \quad P \cdot Q := (\nu J)(P \mid Q) :: I \to K$$

Cette composition définit l'implication :

$$A \cdot B := \{ P \cdot Q \mid P \in A, Q \in B \}^{\perp\!\!\perp}$$
$$A \multimap B := \{ P \mid \forall Q \in A, P \cdot Q \in B \}$$

Connecteurs multiplicatifs, pour A et B d'interfaces disjointes :

$A \otimes B = A \cdot B$	composition parallèle de A et B
$A \parr B = (A^{\perp\!\!\perp} \otimes B^{\perp\!\!\perp})^{\perp\!\!\perp}$	corrélation entre A et B
$A \multimap B = A^{\perp\!\!\perp} \parr B$	implication

Logique des comportements – modalités

Pour décrire les actions des processus, on définit des modalités :
Hennessy-Milner'85, MPW'93

$$P \in [u(\vec{x})]A \qquad \text{si} \qquad \begin{cases} P \text{ émet une action } u(\vec{x}) \\ P \xrightarrow{u(\vec{x})} Q \Rightarrow Q \in A \end{cases}$$

Les modalités sont duales :

$$\big([u(\vec{x})]A\big)^{\perp\!\!\perp} = [\bar{u}(\vec{x})]\big(A^{\perp\!\!\perp}\big)$$

Elles distribuent sur $\wedge$ et $\vee$.

Le système de typage

Axiome et coupure :

$$\frac{}{u{\to}v \vdash u : A^{\perp}, v : A} \qquad \frac{P \vdash \Gamma, \vec{x} : A \qquad Q \vdash \vec{x} : A^{\perp}, \Delta}{(\nu \vec{x})(P \mid Q) \vdash \Gamma, \Delta}$$

Multiplicatifs :

$$\frac{P \vdash \Gamma, \vec{x} : A \qquad Q \vdash \vec{y} : B, \Delta}{P \mid Q \vdash \Gamma, \vec{x}\vec{y} : A \otimes B, \Delta} \qquad \frac{P \vdash \Gamma, \vec{x} : A, \vec{y} : B}{P \vdash \Gamma, \vec{x}\vec{y} : A \parr B}$$

Actions :

$$\frac{P \vdash \Gamma, u\vec{x} : A}{u^{\varepsilon}(\vec{x}).P \vdash \Gamma, u : [\varepsilon]A} \qquad \frac{P \vdash \Gamma, \vec{x} : A}{u^{\varepsilon}(\vec{x}).P \vdash \Gamma, u : \varepsilon A}$$

Plus quantificateurs, points fixes, etc.

Le système de typage

Axiome et coupure :

$$\frac{}{u{\to}v \vdash u : A^{\perp}, v : A} \qquad \frac{P \vdash \Gamma, \vec{x} : A \qquad Q \vdash \vec{x} : A^{\perp}, \Delta}{(\nu \vec{x})(P \mid Q) \vdash \Gamma, \Delta}$$

Multiplicatifs :

$$\frac{P \vdash \Gamma, \vec{x} : A \qquad Q \vdash \vec{y} : B, \Delta}{P \mid Q \vdash \Gamma, \vec{x}\vec{y} : A \otimes B, \Delta} \qquad \frac{P \vdash \Gamma, \vec{x} : A, \vec{y} : B}{P \vdash \Gamma, \vec{x}\vec{y} : A \parr B}$$

Actions :

$$\frac{P \vdash \Gamma, u\vec{x} : A}{u^{\varepsilon}(\vec{x}).P \vdash \Gamma, u : [\varepsilon]A} \qquad \frac{P \vdash \Gamma, \vec{x} : A}{u^{\varepsilon}(\vec{x}).P \vdash \Gamma, u : \varepsilon A}$$

Plus quantificateurs, points fixes, etc.

Un petit exemple

Écrivons 2 en logique linéaire :

$$\dfrac{\dfrac{\dfrac{x{\to}x' \vdash x' : {\uparrow}A, x : {\downarrow}A^{\perp} \qquad y'{\to}y \vdash y' : {\downarrow}A^{\perp}, y : {\uparrow}A}{x{\to}x' \mid y'{\to}y \vdash x'y' : {\uparrow}A \otimes {\downarrow}A^{\perp}, x : {\downarrow}A^{\perp}, y : {\uparrow}A}}{\bar{u}\langle xy\rangle = u(x'y')(x{\to}x' \mid y'{\to}y) \vdash u : ?({\uparrow}A \otimes {\downarrow}A^{\perp}), x : {\downarrow}A^{\perp}, y : {\uparrow}A \quad \bar{u}\langle yz\rangle = u(y'z')(y{\to}y' \mid z'{\to}z) \vdash u : ?({\uparrow}A \otimes {\downarrow}A^{\perp}), y : {\downarrow}A^{\perp}, z : {\uparrow}A}}{(\nu y)(\bar{u}\langle xy\rangle \mid \bar{u}\langle yz\rangle) \vdash u : ?({\uparrow}A \otimes {\downarrow}A^{\perp}), x : {\downarrow}A^{\perp}, z : {\uparrow}A}}{(\nu y)(\bar{u}\langle xy\rangle \mid \bar{u}\langle yz\rangle) \vdash uxz : !({\uparrow}A \multimap {\uparrow}A) \multimap {\uparrow}A \multimap {\uparrow}A}$$

Propriétés du typage

Théorème

$P \vdash \Gamma \Longrightarrow P \Vdash \Gamma$

Corollaire

Les processus typés ne divergent pas.

Corollaire

Les processus typés ne se bloquent pas.

Conjecture

Le typage est préservé par réduction, à une congruence structurelle près.

Connecteurs construits

- La structure de treillis des comportements donne intersection, union, quantification, points fixes,

On déduit la construction d'autres connecteurs logiques :

- Modalités et intersection/union donnent les connecteurs additifs qui parlent de choix,
- Modalités et points fixes donnent les modalités exponentielles qui parlent de réplication.

Les exponentielles

La formule :

$$[?\alpha]A := \mu X.\Big([\alpha]A \vee \bot \vee (X \parr X) \cdot \delta\Big)$$

avec $\delta\langle uvw\rangle := u{\to}w \mid v{\to}w$.

Clients :

$$\frac{P \vdash \Gamma, \vec{x} : A}{\bar{u}(\vec{x}).P \vdash \Gamma, u : ?A} \qquad \frac{P \vdash \Gamma, u : ?A, v : ?A}{P[w/u,v] \vdash \Gamma, w : ?A} \qquad \frac{P \vdash \Gamma}{P \vdash \Gamma, u : ?A}$$

Serveur :

$$\frac{P \vdash ?\Gamma, \vec{x} : A}{!u(\vec{x}).P \vdash ?\Gamma, u : !A}$$

Observations vs logiques

Ajouter des règles logiques revient à contraindre l'observation :

- La règle mix signifie que l'observation est stable par composition :

$$\frac{P \vdash \Gamma \qquad Q \vdash \Delta}{P \mid Q \vdash \Gamma, \Delta} \quad \text{ssi} \quad \bot\!\!\!\bot \mid \bot\!\!\!\bot \subseteq \bot\!\!\!\bot$$

- La logique affine correspond aux observations insensibles à la divergence :

$$\frac{P \vdash \Gamma}{P \vdash \Gamma, \Delta} \quad \text{ssi} \quad \forall P \in \bot\!\!\!\bot, \forall Q, (P \mid Q) \in \bot\!\!\!\bot$$

- Beaucoup d'observations supportent du non-déterminisme :

$$\frac{P \vdash \Gamma \qquad Q \vdash \Gamma}{P \oplus Q \vdash \Gamma} \qquad \frac{P \vdash \Gamma, u : !A \qquad Q \vdash \Delta, u : !A}{P \mid Q \vdash \Gamma, \Delta, u : !A}$$

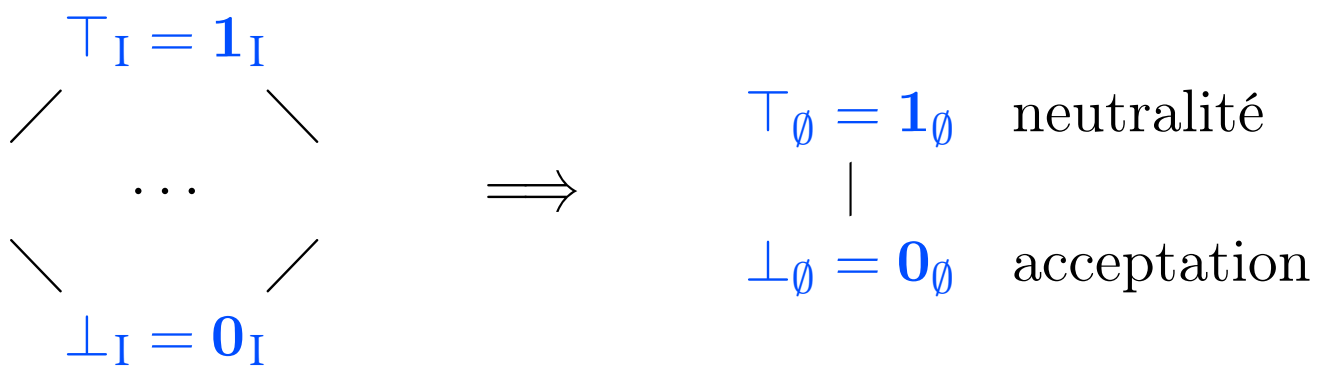
Constantes

Pour toute interface I, on pose

$$\mathbf{0}_{\mathrm{I}} = (\emptyset : \mathrm{I})^{\perp\!\!\perp\perp\!\!\perp} \qquad \top_{\mathrm{I}} = \text{tous les processus d'interface I}$$

$$\mathbf{1}_{\mathrm{I}} = (\{1\} : \mathrm{I})^{\perp\!\!\perp\perp\!\!\perp} \qquad \bot_{\mathrm{I}} = \{\, \mathrm{P} \mid \mathrm{P} \in \perp\!\!\!\perp, \mathrm{P} :: \mathrm{I} \,\}$$

May/fair-testing :



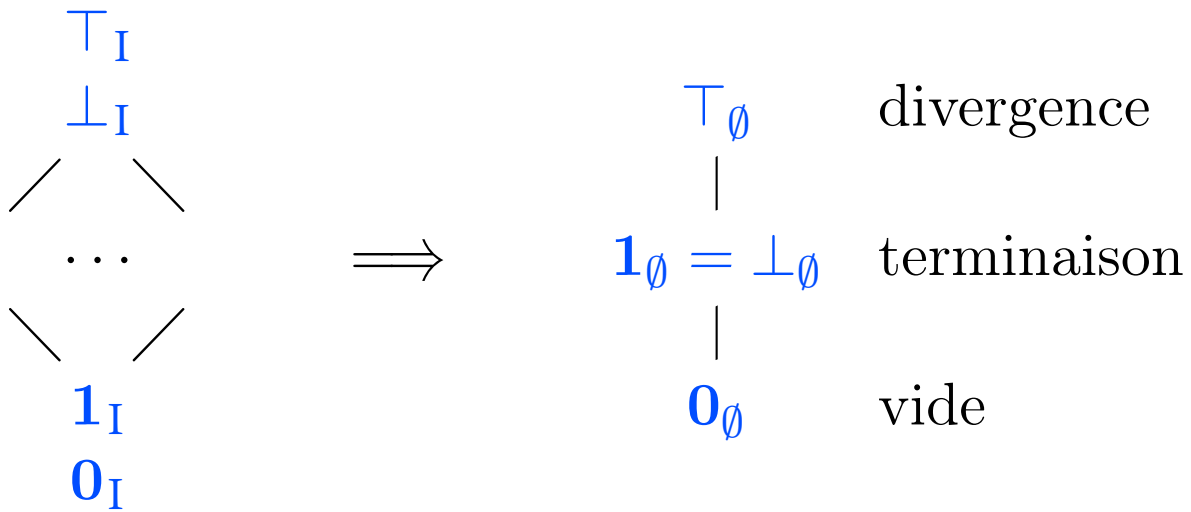
Constantes

Pour toute interface I, on pose

$$\mathbf{0}_{\mathrm{I}} = (\emptyset : \mathrm{I})^{\perp\!\!\perp\perp\!\!\perp} \qquad \top_{\mathrm{I}} = \text{tous les processus d'interface } \mathrm{I}$$

$$\mathbf{1}_{\mathrm{I}} = (\{1\} : \mathrm{I})^{\perp\!\!\perp\perp\!\!\perp} \qquad \bot_{\mathrm{I}} = \{\, \mathsf{P} \mid \mathsf{P} \in \perp\!\!\!\perp, \mathsf{P} :: \mathrm{I} \,\}$$

Test par terminaison :



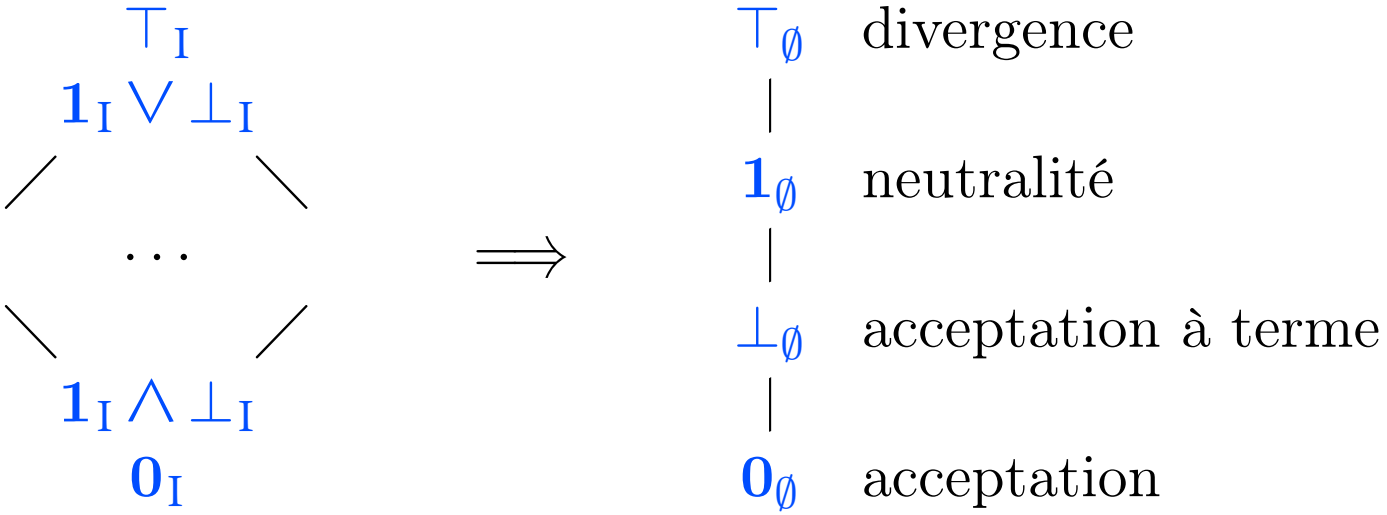
Constantes

Pour toute interface I, on pose

$$\mathbf{0}_{\mathrm{I}} = (\emptyset : \mathrm{I})^{\perp\!\!\perp\perp\!\!\perp} \qquad \top_{\mathrm{I}} = \text{tous les processus d'interface } \mathrm{I}$$

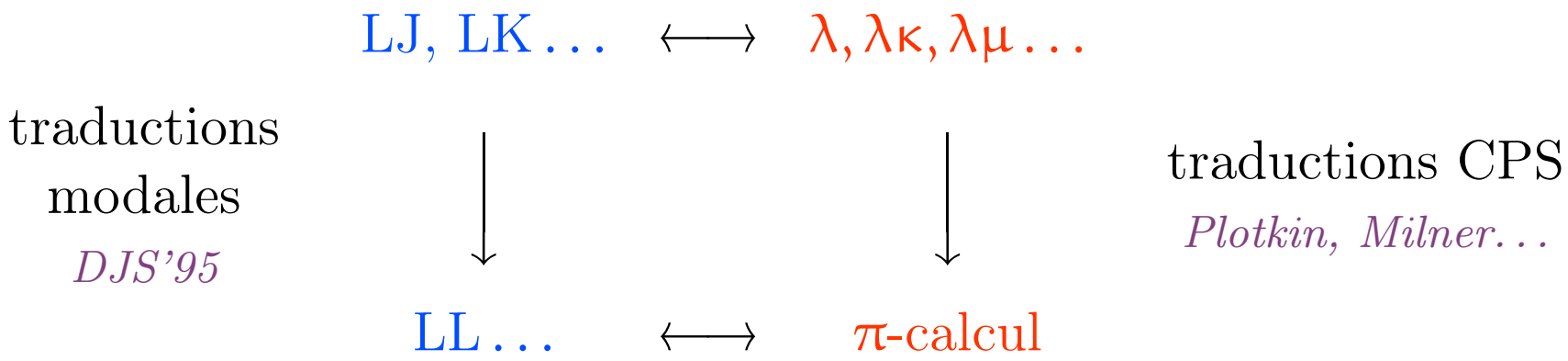
$$\mathbf{1}_{\mathrm{I}} = (\{1\} : \mathrm{I})^{\perp\!\!\perp\perp\!\!\perp} \qquad \bot_{\mathrm{I}} = \{\, \mathrm{P} \mid \mathrm{P} \in \perp\!\!\!\perp, \mathrm{P} :: \mathrm{I} \,\}$$

Must-testing :



Du fonctionnel au concurrent

Il existe des traductions entre systèmes classiques et concurrents



- traductions modales : introduire des exponentielles pour autoriser les contractions/affaiblissements
- traductions CPS : choisir une stratégie d'évaluation et la formuler comme une manipulation de continuations

Décomposition du calcul fonctionnel

$A \to B$	système	stratégie	référence
$!A \multimap B$		linéaire de tête	Hyland/Ong ?
$!{\downarrow}A \multimap {\downarrow}B$	LJT	appel par nom	Milner
$!?A \multimap ?B$	LKT	+ contrôle $\to \lambda\kappa$	?
$!A \multimap !B$	LJQ	appel par valeur	?
$!A \multimap ?!B$	LKQ	+ contrôle	Honda/Yoshida/Berger

Conclusions

Ce qu'on a :

- Exploration du sens interactif de la disjonction classique,
- Connection à la Curry-Howard entre LL et π-calcul :
  $\longrightarrow$ des modèles concurrents de LL,
  $\longrightarrow$ typage et spécification de processus.
- Reconstruction logique de traductions du λ-calcul.

Il reste beaucoup à faire :

- Étudier LL par des méthodes concurrentes
  (réseaux de preuve, élimination des coupures, λ différentiel, etc)
- Étudier la concurrence par des méthodes logiciennes
  (plus de spécification, extensions de la logique, etc)

Plus de logique

On aurait de quoi comprendre la logique linéaire en termes concurrents :

- normalisation et élimination des coupures,
- quel bon formalisme pour les réseaux de preuve,
- quel sens donner aux diverses équivalences observationnelles

Et pourquoi pas

- une reconstruction du λ-calcul différentiel ?
- des études de complexité (LLL, ELL...)
- quel statut donner au non-déterminisme

Plus de spécification

En jouant sur les observations, on montre que

- les isomorphismes de MLL spécifient des schémas de routage
- les formules prouvables spécifient des protocoles sûrs

Il faut chercher quelles extensions apporter pour

- garder la cohérence sans imposer terminaison et non-blocage
- parler d'ordonnancement (nécessite du codage ?)
- parler d'état (s'inspirer de la logique de Hoare ?)