Une géométrie du calcul

Réseaux de preuve, Appel-Par-Pousse-Valeur et topologie du Consensus

Jules Chouquet

IRIF

Université de Paris

6 décembre 2019

De quoi ça parle

En général

De quoi ça parle

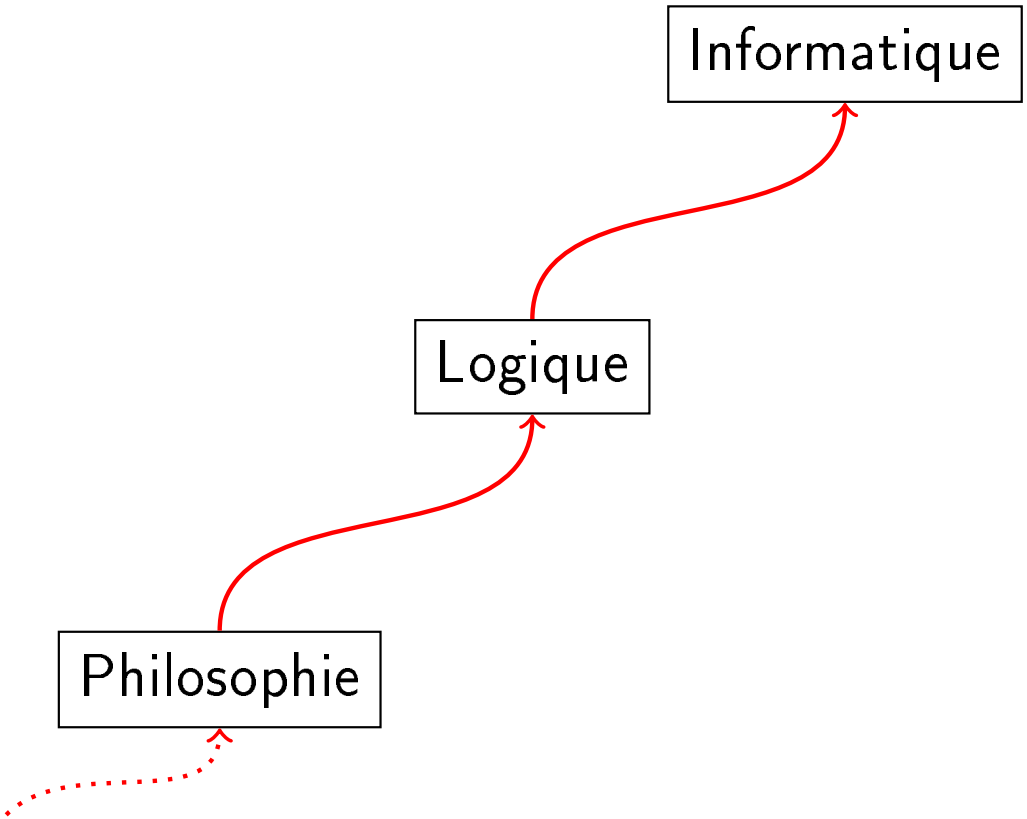
En général

Logique mathématique

Informatique théorique

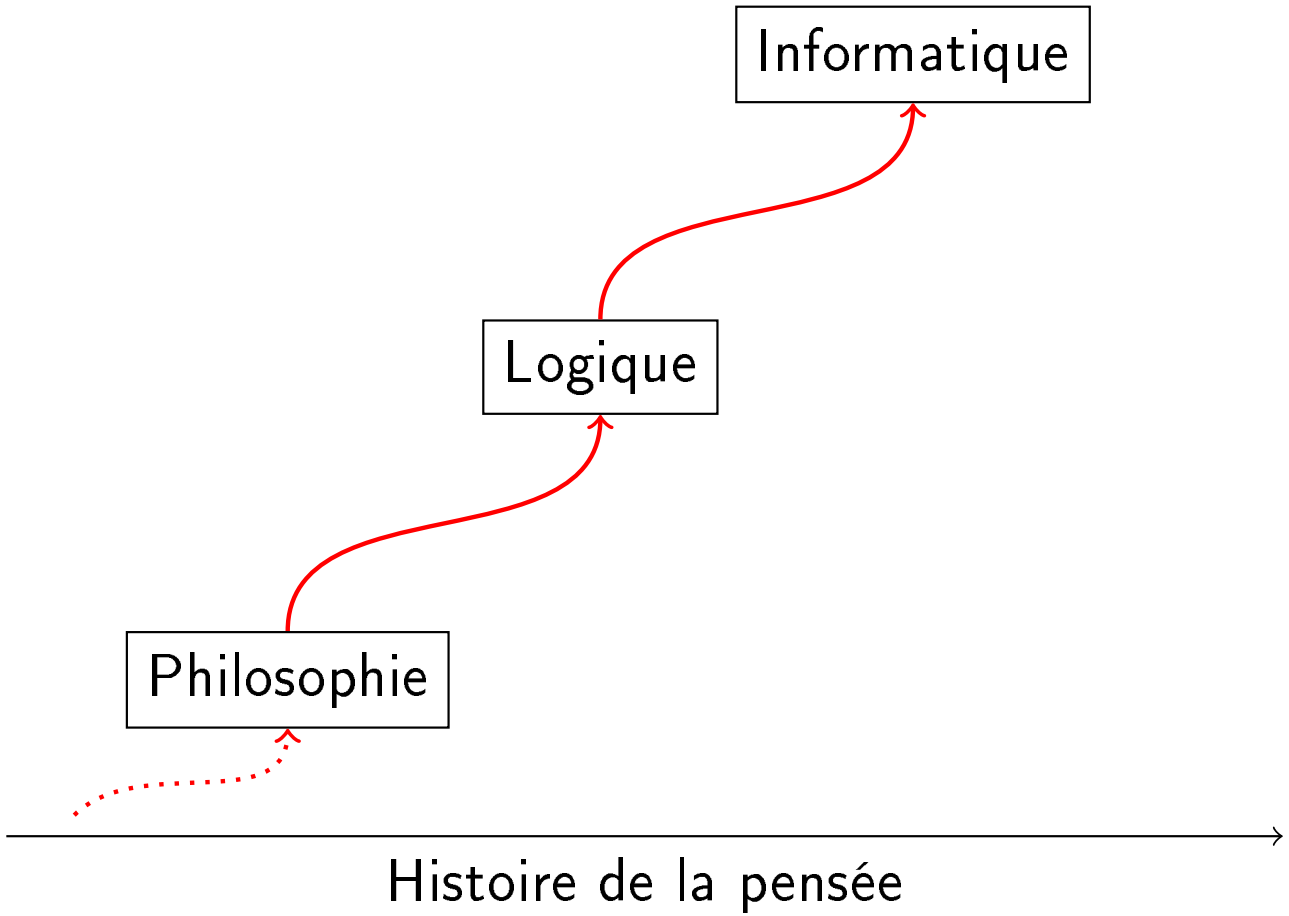
Un chemin

Le raisonnement et le calcul



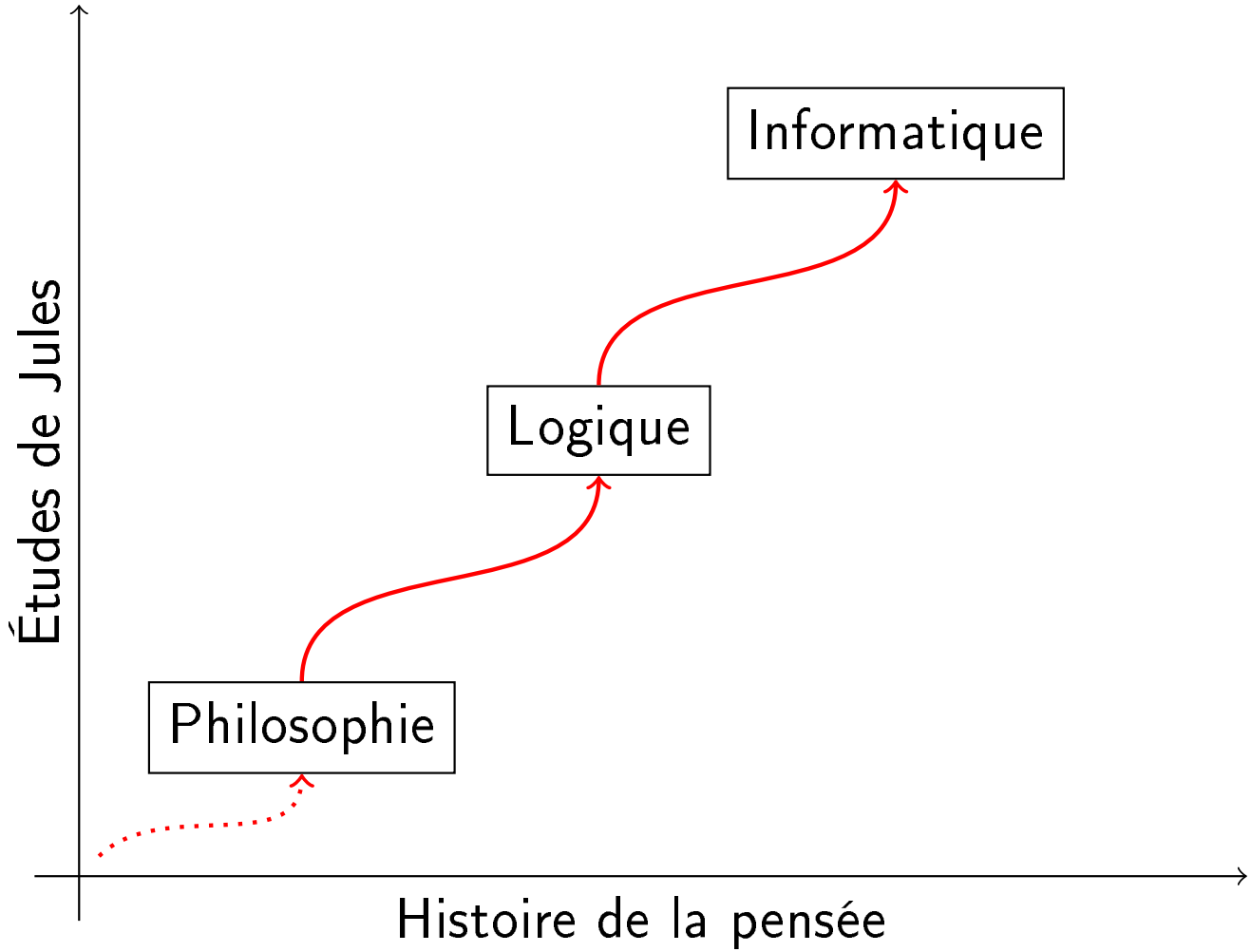
Un chemin

Le raisonnement et le calcul

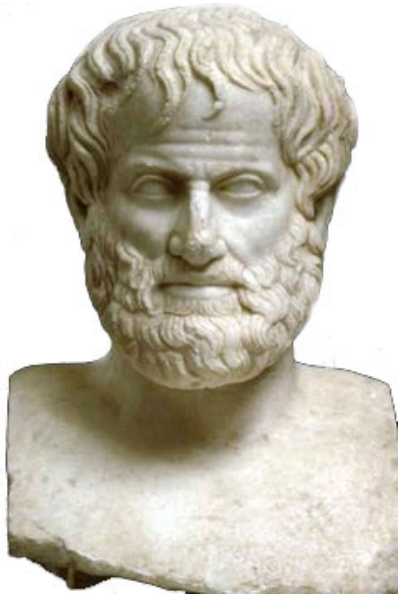


Un chemin

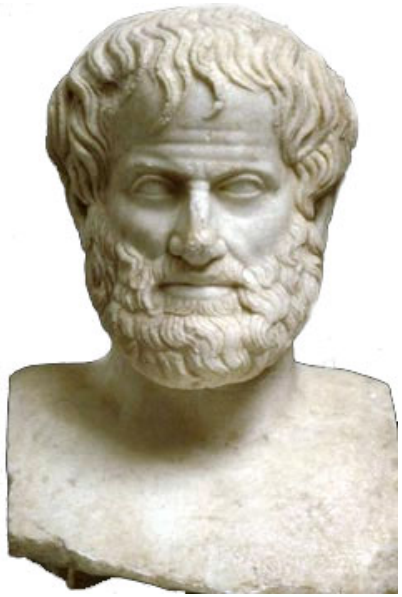
Le raisonnement et le calcul



Les formes du raisonnement

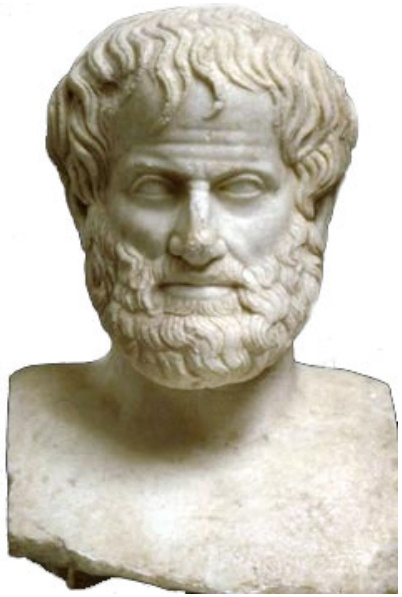


Les formes du raisonnement



$$\frac{\text{Tous les } A \text{ sont } B \qquad \text{Tous les } B \text{ sont } C}{\text{Tous les } A \text{ sont } C}$$

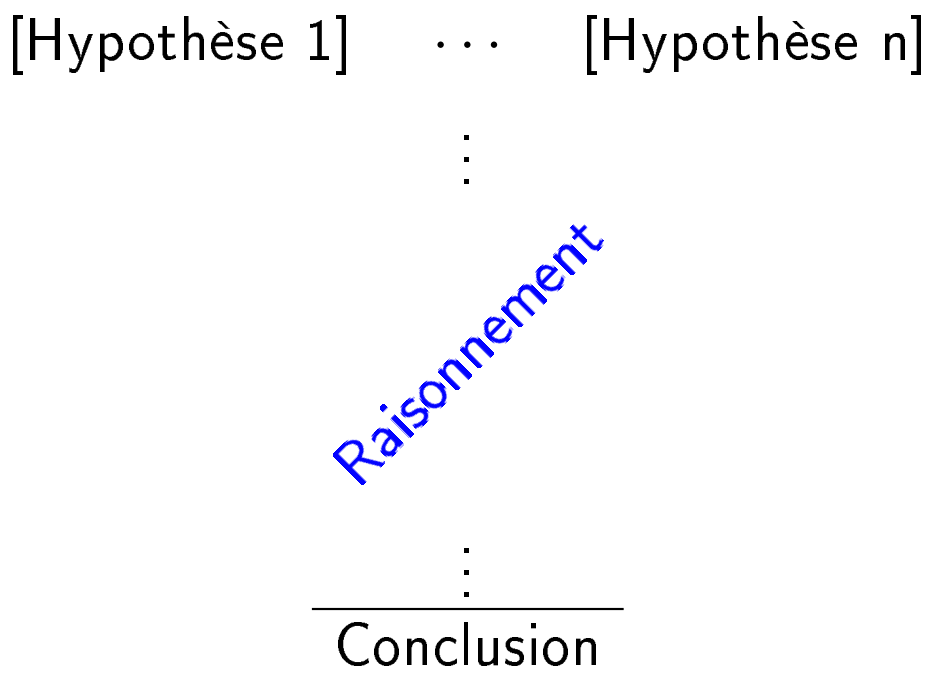
Les formes du raisonnement



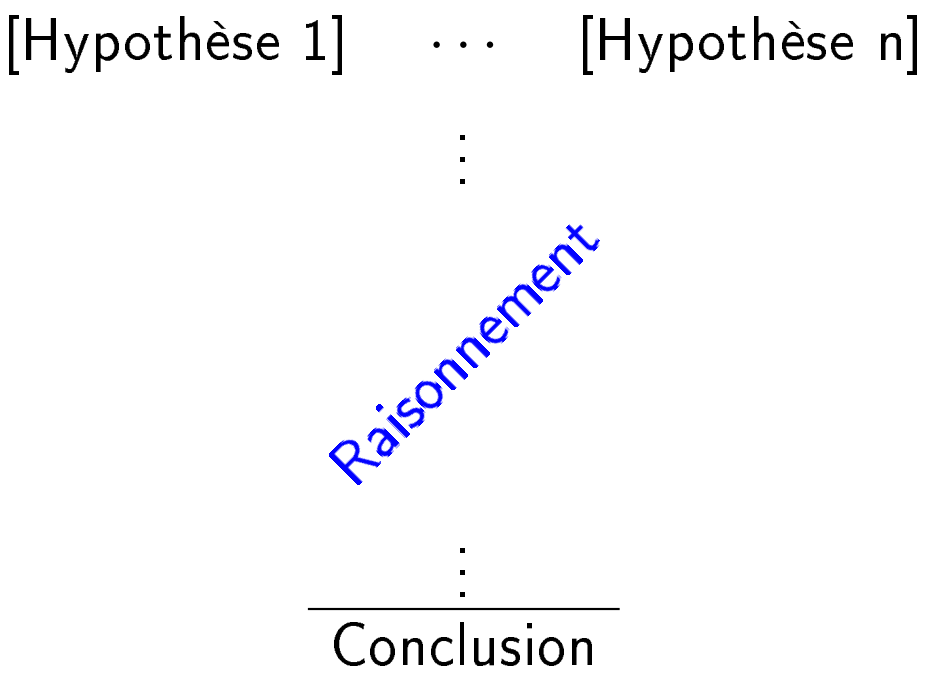
$$\frac{\text{Tous les } A \text{ sont } B \qquad \text{Tous les } B \text{ sont } C}{\text{Tous les } A \text{ sont } C}$$

$$\frac{\text{Tous les doctorants sont précaires} \qquad \text{Tous les précaires sont inquiets}}{\text{Tous les doctorants sont inquiets}}$$

Démonstrations



Démonstrations



$$\frac{A \vdash B}{\vdash A \to B}$$

$$\frac{\vdash A \to B \qquad \vdash A}{\vdash B}$$

Théorie de la démonstration

Théorie de la démonstration

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma, A \vdash B}}{\Gamma \vdash A \to B} \quad \dfrac{\pi_2}{\Delta \vdash A}}{\Gamma, \Delta \vdash B} \quad \to \quad \dfrac{\pi_1\,[\pi_2/(A \vdash)]}{\Gamma, \Delta \vdash B}$$

Cette réécriture est un algorithme.

Théorie de la démonstration

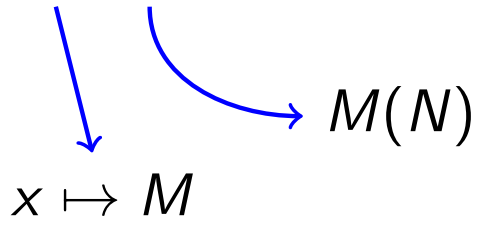
$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma, A \vdash B}}{\Gamma \vdash A \to B} \quad \dfrac{\pi_2}{\Delta \vdash A}}{\Gamma, \Delta \vdash B} \quad \to \quad \dfrac{\pi_1\,[\pi_2/(A \vdash)]}{\Gamma, \Delta \vdash B}$$

Cette réécriture est un algorithme.

(Gehrhard Gentzen)

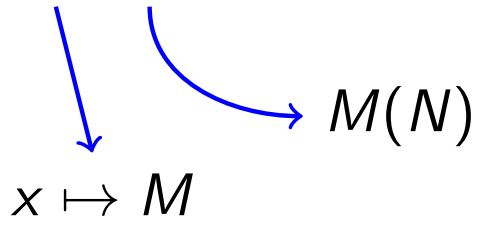
λ-calcul

- Une syntaxe simple : $M, N, \ldots := x \mid \lambda x M \mid MN$



λ-calcul

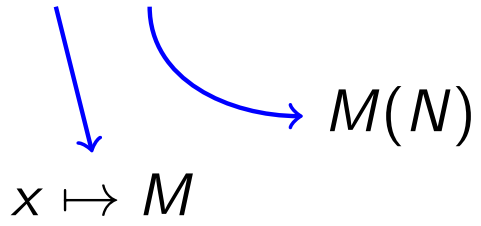
- Une syntaxe simple : $M, N, \ldots := x \mid \lambda x M \mid MN$



- Une unique règle de calcul : $(\lambda x M)N \to_\beta M[N/x]$

λ-calcul

- Une syntaxe simple : $M, N, \ldots := x \mid \lambda x M \mid MN$



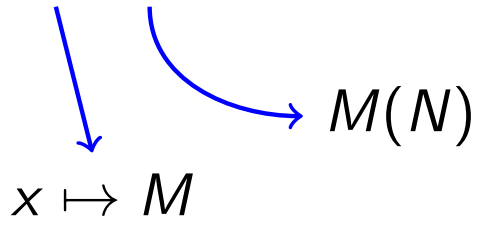
- Une unique règle de calcul : $(\lambda x M)N \to_\beta M[N/x]$

Exemples de λ-termes

$\lambda x(x+1)$ $\lambda y(y*y)$ $\lambda x 42$

λ-calcul

- Une syntaxe simple : $M, N, \ldots := x \mid \lambda x M \mid MN$



- Une unique règle de calcul : $(\lambda x M)N \to_\beta M[N/x]$

Exemples de λ-termes

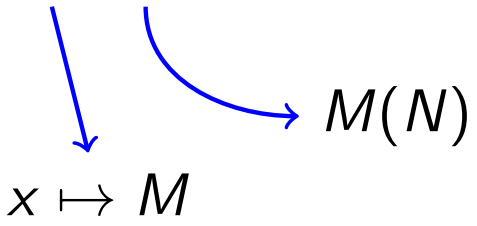
$\lambda x(x+1)$ $\qquad$ $\lambda y(y*y)$ $\qquad$ $\lambda x 42$

Exemples de β-réductions

$(\lambda x(x+1))5 \to_\beta 5+1$ $\qquad$ $(\lambda y(y*y))5 \to_\beta 5*5$ $\qquad$ $(\lambda x 42)5 \to_\beta 42$

λ-calcul

- Une syntaxe simple : $M, N, \ldots := x \mid \lambda x M \mid MN$



- Une unique règle de calcul : $(\lambda x M)N \to_\beta M[N/x]$

Exemples de λ-termes

$\lambda x(x+1)$ $\qquad$ $\lambda y(y*y)$ $\qquad$ $\lambda x 42$

Exemples de β-réductions

$(\lambda x(x+1))5 \to_\beta 5+1$ $\qquad$ $(\lambda y(y*y))5 \to_\beta 5*5$ $\qquad$ $(\lambda x 42)5 \to_\beta 42$

(Alonzo Church)

Correspondance de Curry-Howard

Correspondance de Curry-Howard

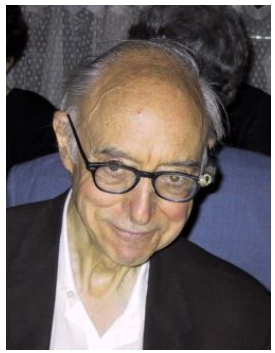
$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma, x : A \vdash M : B}}{\Gamma \vdash \lambda x M : A \to B} \quad \dfrac{\pi_2}{\Delta \vdash N : A}}{\Gamma, \Delta \vdash (\lambda x M) N : B} \quad \to \quad \dfrac{\pi_1 \, [\pi_2/(A \vdash)]}{\Gamma, \Delta \vdash M[N/x] : B}$$

Correspondance de Curry-Howard

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma, x : A \vdash M : B}}{\Gamma \vdash \lambda x M : A \to B} \qquad \dfrac{\pi_2}{\Delta \vdash N : A}}{\Gamma, \Delta \vdash (\lambda x M) N : B} \quad \to \quad \dfrac{\pi_1 \, [\pi_2/(A \vdash)]}{\Gamma, \Delta \vdash M[N/x] : B}$$

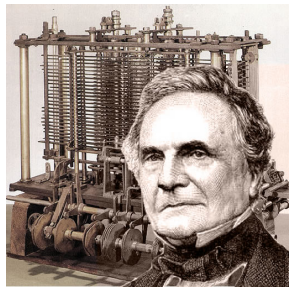
Haskell B. Curry

Robert Feys



William A. Howard

Une autre histoire



Charles Babbage

Alan Turing

IX^e — 1821 — 1843 — 1930 — 1997 — 2009

Algoritmi

Ada Lovelace

Kasparov VS Deep Blue

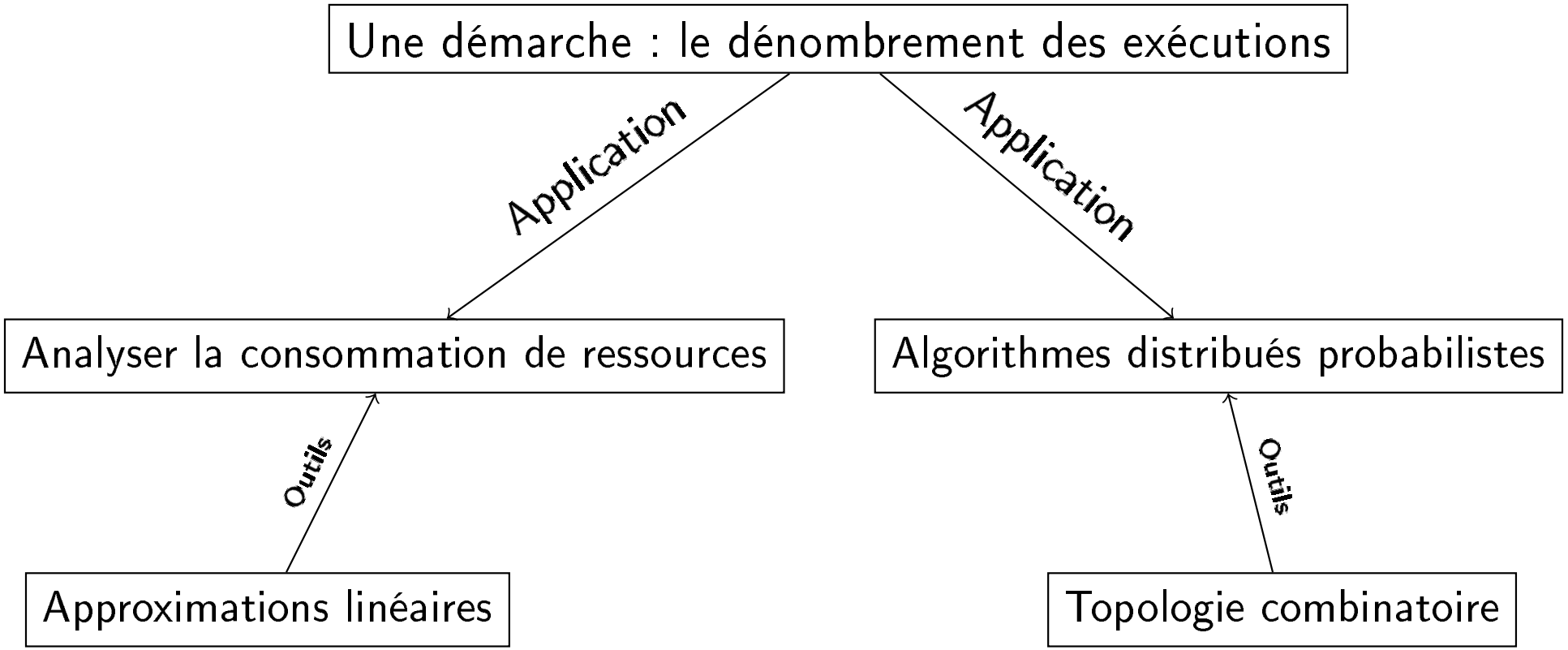
De quoi ça parle

Plus précisément

Une démarche : le dénombrement des exécutions

De quoi ça parle

Plus précisément



Sémantique quantitative et approximations

Développement de Taylor

$f : \mathbf{R} \to \mathbf{R}$
$x \in \mathbf{R}$

$$f(x) = \sum_{n \in \mathbf{N}} \frac{1}{n!} f^n(0) \cdot x^n$$

$P : X \to Y$
$x : X$

$$P\ x = \sum_{n \in \mathbf{N}} p_n \underbrace{(x, \ldots, x)}_{n}$$

- λ-calcul différentiel
- logique linéaire différentielle
  $\to$ Ehrhard-Regnier

Linéaire $\Leftrightarrow$ Pas de duplication ni d'effacement

Comment développer cette analogie dans des langages de preuves ou de programmes les plus généraux et puissants possibles ?

Contributions

Concrétiser l'analogie entre approximation de fonctions et approximation de programmes :

Contributions

Concrétiser l'analogie entre approximation de fonctions et approximation de programmes :

- Réseaux de preuve de la Logique Linéaire (Girard)
  - Système plus général que le λ-calcul
  - Bonnes propriétés logiques et algorithmiques

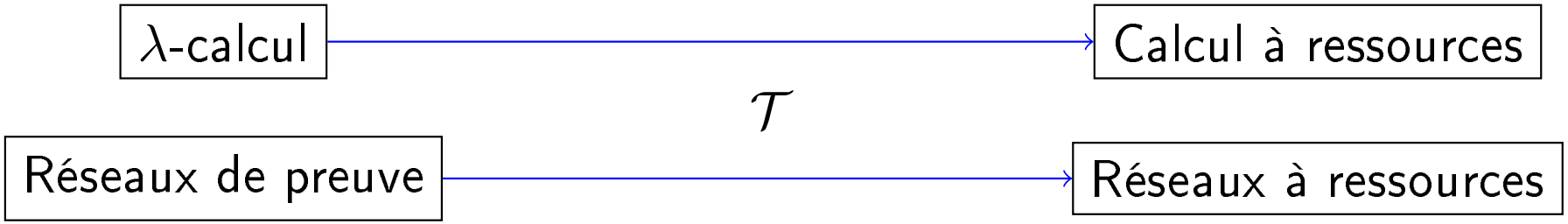
Contributions

Concrétiser l'analogie entre approximation de fonctions et approximation de programmes :

- Réseaux de preuve de la Logique Linéaire (Girard)
  - Système plus général que le λ-calcul
  - Bonnes propriétés logiques et algorithmiques
- Appel-Par-Pousse-Valeur (Levy)
  - Permet d'encoder des stratégies d'évaluation du λ-calcul
  - Plus proche d'un langage de programmation

Quels outils pour cette correspondance ?

Ingrédients du développement de Taylor syntaxique



Idée

Si P est un reseau de preuves ou un λ-terme, alors :

$$\mathcal{T}(P) = \sum_{i \in \mathcal{I}} a_i \cdot p_i$$

où pour tout $i \in \mathcal{I}$, p_i est une *approximation* de P.

Quel langage pour ces approximations ?

Calculs à ressources

Une syntaxe linéaire

Calculs à ressources

Une syntaxe linéaire

Réduction de ressources

Réduction entre termes qui ne duplique ni n'efface aucune composante du terme réduit.

Calculs à ressources

Une syntaxe linéaire

Réduction de ressources

Réduction entre termes qui ne duplique ni n'efface aucune composante du terme réduit.

Exemples

- $\langle \lambda x \langle x \rangle [x, x] \rangle [M', M'', M'''] \to \langle M' \rangle [M'', M''']$

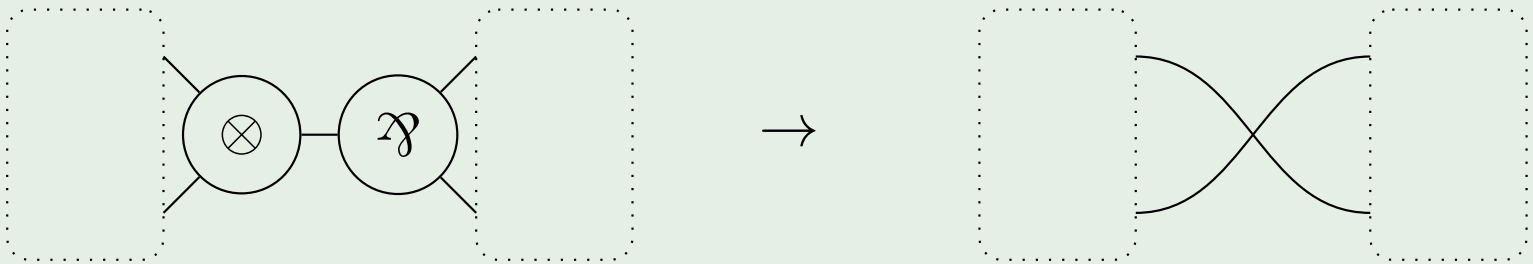
Calculs à ressources

Une syntaxe linéaire

Réduction de ressources

Réduction entre termes qui ne duplique ni n'efface aucune composante du terme réduit.

Exemples

- $\langle \lambda x \langle x \rangle [x, x] \rangle [M', M'', M'''] \to \langle M' \rangle [M'', M''']$
- 

Correction du développement de Taylor

Résultat souhaité

Le développement de Taylor permet de simuler la dynamique du calcul de départ. *In extenso :*

$$\text{Si } P \to Q, \text{ alors } \mathcal{T}(P) \Rightarrow \mathcal{T}(Q)$$

Quelles méthodes pour la correction du développement de Taylor ?

Si $P \to Q$, $\mathcal{T}(P) \Rightarrow \mathcal{T}(Q)$

1. Disposer d'un calcul à ressources approprié.
2. Définir le développement de Taylor avec des sommes infinies dans le susdit calcul.
3. Définir une réduction $\Rightarrow$ qui s'applique à des sommes infinies
4. Vérifier que $\Rightarrow$ permet bien de simuler la sémantique opérationnelle du calcul de départ.

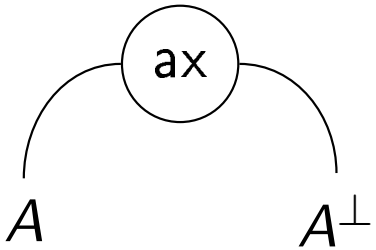
Une présentation graphique des démonstrations

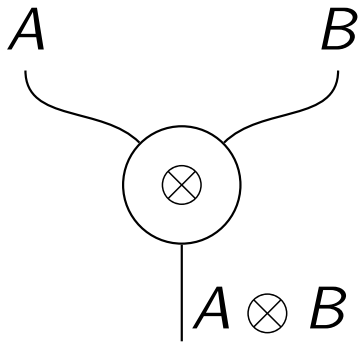
Une présentation graphique des démonstrations

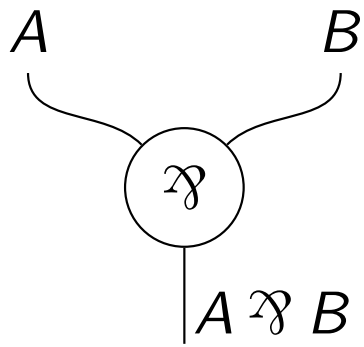
$$\textbf{MELL} : A, B ::= X \mid A \otimes B \mid A \parr B \mid !A \mid ?A$$

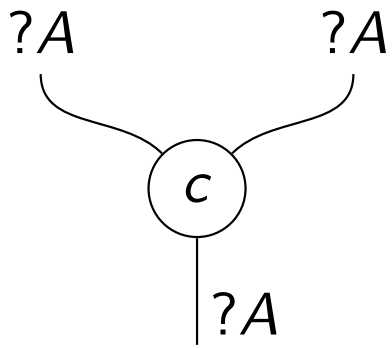
Une présentation graphique des démonstrations

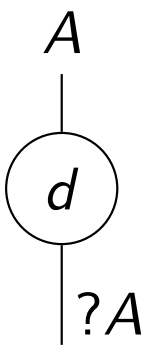
$$\textbf{MELL} : A, B ::= X \mid A \otimes B \mid A \parr B \mid !A \mid ?A$$

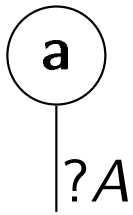


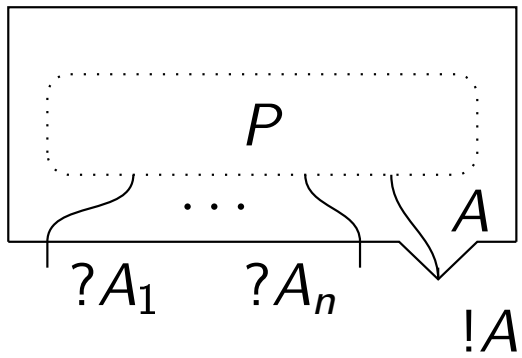






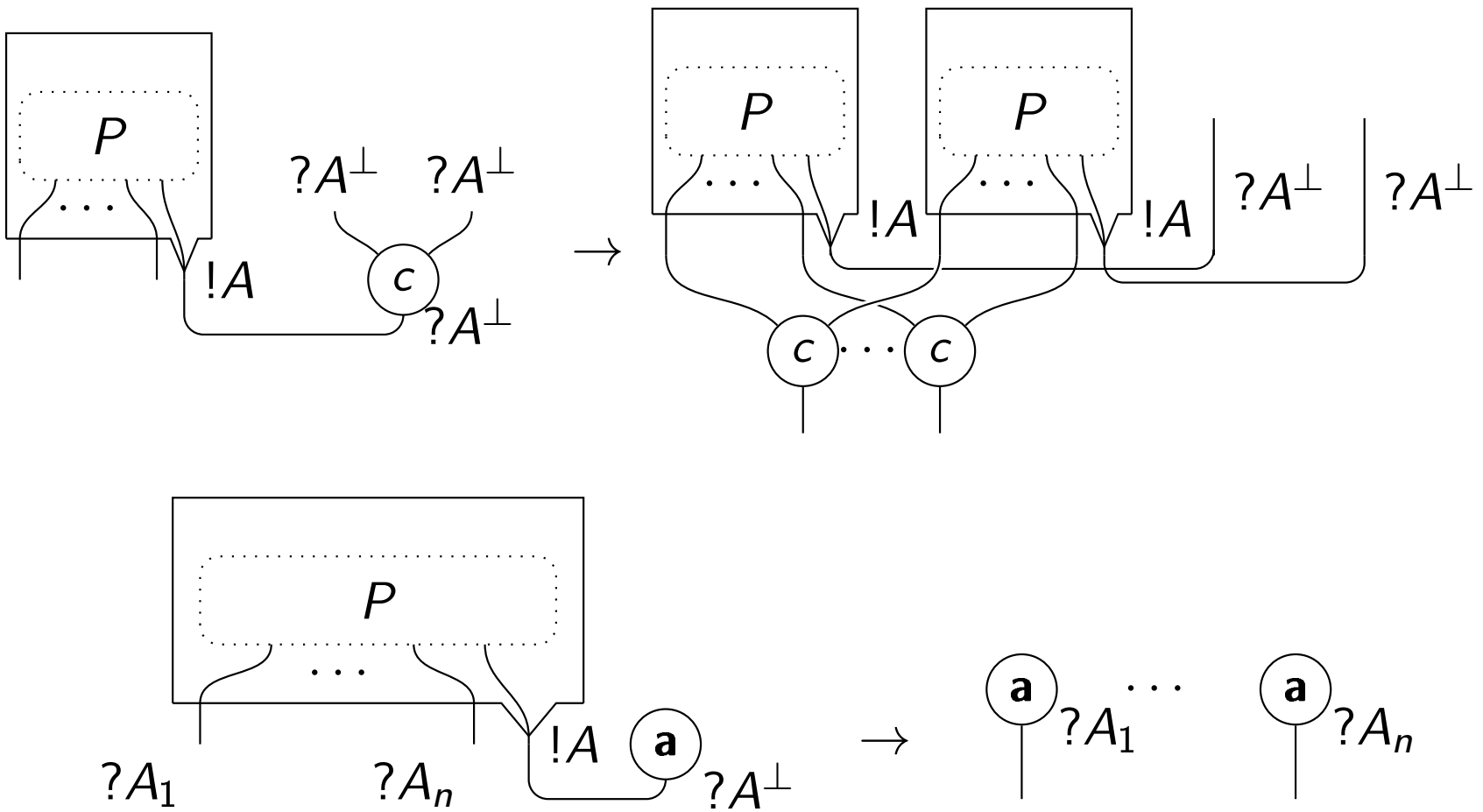






Boîtes, duplication et effacement

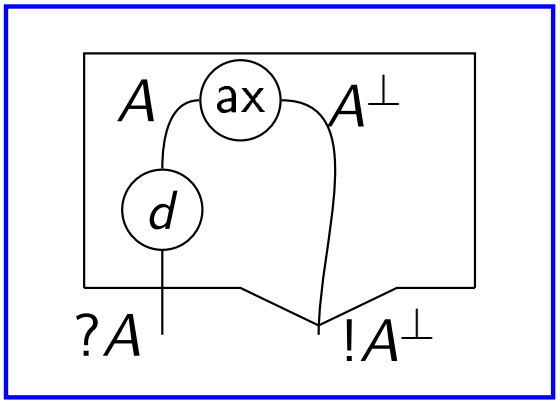
Boîtes, duplication et effacement



Approximations n-linéaires d'une boîte

Dans les réseaux à ressources

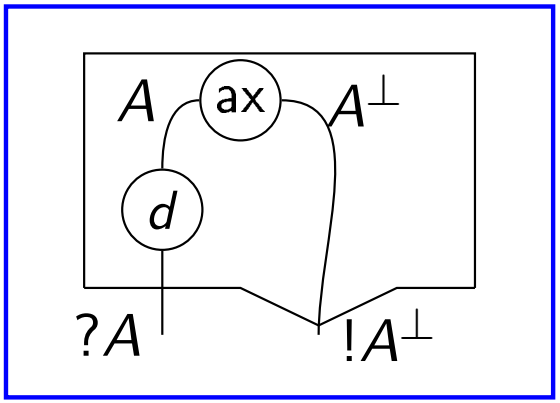
MELL

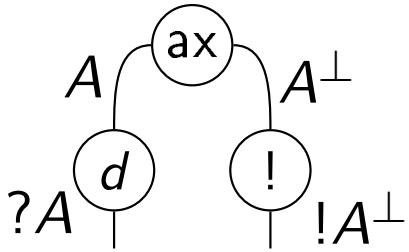


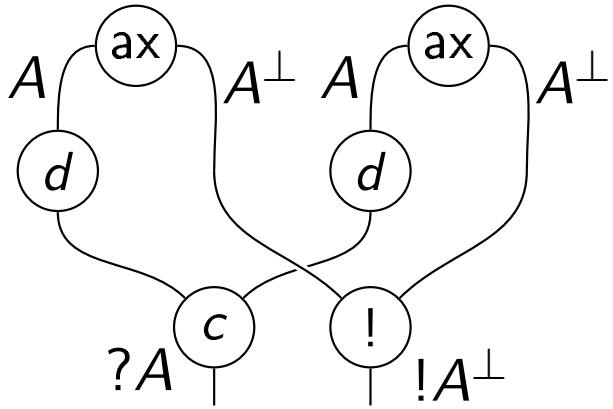
Approximations n-linéaires d'une boîte

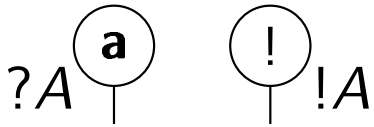
Dans les réseaux à ressources

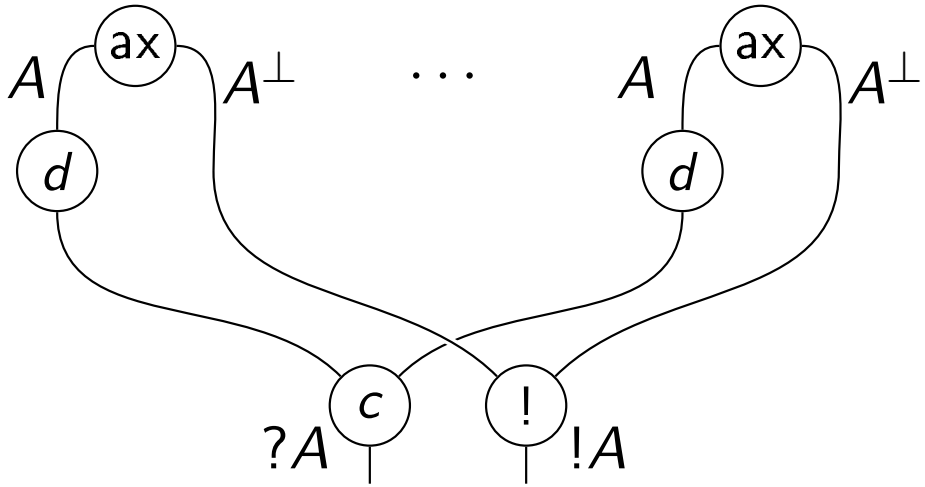
MELL





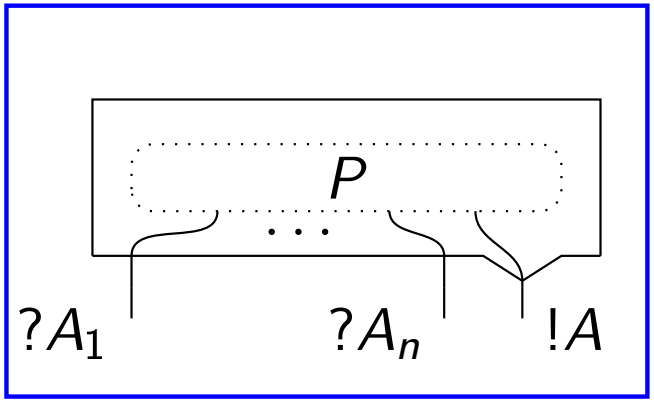






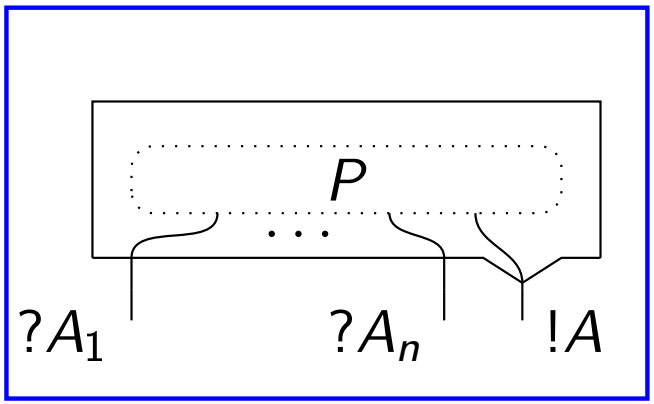
On simule la duplication et la suppression des boîtes

Développement de Taylor des réseaux de preuve



MELL

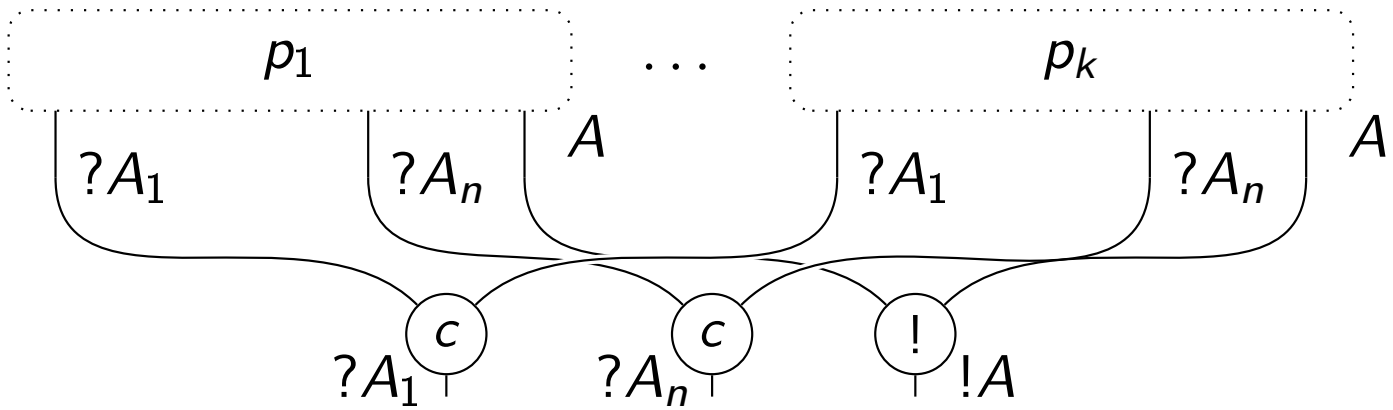
Développement de Taylor des réseaux de preuve



MELL

Réseaux à ressources

$$\sum_{k \in \mathbf{N}} \frac{1}{k!}$$



Quelles méthodes pour la correction du développement de Taylor ?

Si $P \to Q$, $\mathcal{T}(P) \Rightarrow \mathcal{T}(Q)$

1. Disposer d'un calcul à ressources approprié.
2. Définir le développement de Taylor avec des sommes infinies dans le susdit calcul.

Quelles méthodes pour la correction du développement de Taylor ?

Si $P \to Q$, $\mathcal{T}(P) \Rightarrow \mathcal{T}(Q)$

1. Disposer d'un calcul à ressources approprié.
2. Définir le développement de Taylor avec des sommes infinies dans le susdit calcul.

$\longrightarrow$ Ehrhard & Regnier (2005)

Quelles méthodes pour la correction du développement de Taylor ?

Si $P \to Q$, $\mathcal{T}(P) \Rightarrow \mathcal{T}(Q)$

1. Disposer d'un calcul à ressources approprié.
2. Définir le développement de Taylor avec des sommes infinies dans le susdit calcul.

$\longrightarrow$ Ehrhard & Regnier (2005)

3. **Définir $\Rightarrow$ dans des sommes infinies de réseaux à ressources**
4. **Vérifier que $\Rightarrow$ permet bien de simuler l'élimination des coupures de MELL.**

Définir $\Rightarrow$

(Dans des sommes infinies de réseaux à ressources)

Définir $\Rightarrow$

(Dans des sommes infinies de réseaux à ressources)

Réduction $\Rightarrow$

Élimination des coupures parallèles étendues aux sommes infinies.

Définir $\Rightarrow$

(Dans des sommes infinies de réseaux à ressources)

Réduction $\Rightarrow$

Élimination des coupures parallèles étendues aux sommes infinies.

Lemme clef

La longueur des chemins d'interrupteurs augmente à hauteur de $2n!$ sous réduction parallèle.

Propriété

« $\Rightarrow$ » préserve des propriétés géométriques et combinatoires, qui sont vérifiées dans $\mathcal{T}(P)$ pour tout réseau P de **MELL**.

Résultats

Théorème (Chouquet-Vaux, CSL 2018)

$\mathcal{T}(P) \Rightarrow \_$ est toujours bien définie

Théorème (n° 8, section 2.6.4)

Si $P \to Q$, $\mathcal{T}(P) \Rightarrow \mathcal{T}(Q)$

Quelles méthodes pour la correction du développement de Taylor ?

Si $P \to Q$, $\mathcal{T}(P) \Rightarrow \mathcal{T}(Q)$

1. Disposer d'un calcul à ressources approprié.
2. Définir le développement de Taylor avec des sommes infinies dans le susdit calcul.
3. Définir $\Rightarrow$ dans des sommes infinies de termes à ressources.
4. Vérifier que $\Rightarrow$ permet bien de simuler la sémantique opérationnelle du calcul de départ.

Quelles méthodes pour la correction du développement de Taylor ?

Si $P \to Q$, $\mathcal{T}(P) \Rightarrow \mathcal{T}(Q)$

1. **Disposer d'un calcul à ressources approprié.**
2. Définir le développement de Taylor avec des sommes infinies dans le susdit calcul.
3. Définir $\Rightarrow$ dans des sommes infinies de termes à ressources.
4. Vérifier que $\Rightarrow$ permet bien de simuler la sémantique opérationnelle du calcul de départ.

Résultats

Théorèmes (Chouquet, MFPS 2019)

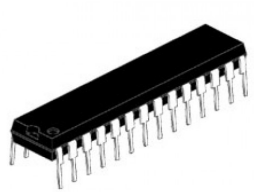
- Pour $M \in \mathbf{CBNeed}$, si $M \to N$, $\mathcal{T}(M) \Rightarrow \mathcal{T}(N)$.
- Pour $M \in \mathbf{PCF}$, si $M \to N$, $\mathcal{T}(M) \Rightarrow \mathcal{T}(N)$.

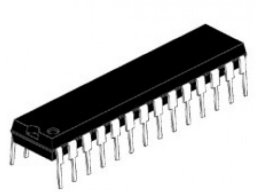
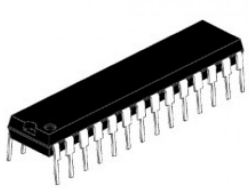
Théorème (Chouquet-Tasson, CSL 2020)

Pour $M \in \Lambda_{\mathsf{pv}}$, si $M \to N$, $\mathcal{T}(M) \Rightarrow \mathcal{T}(N)$.

Retour aux algorithmes

Travailler à plusieurs c'est plus sympa



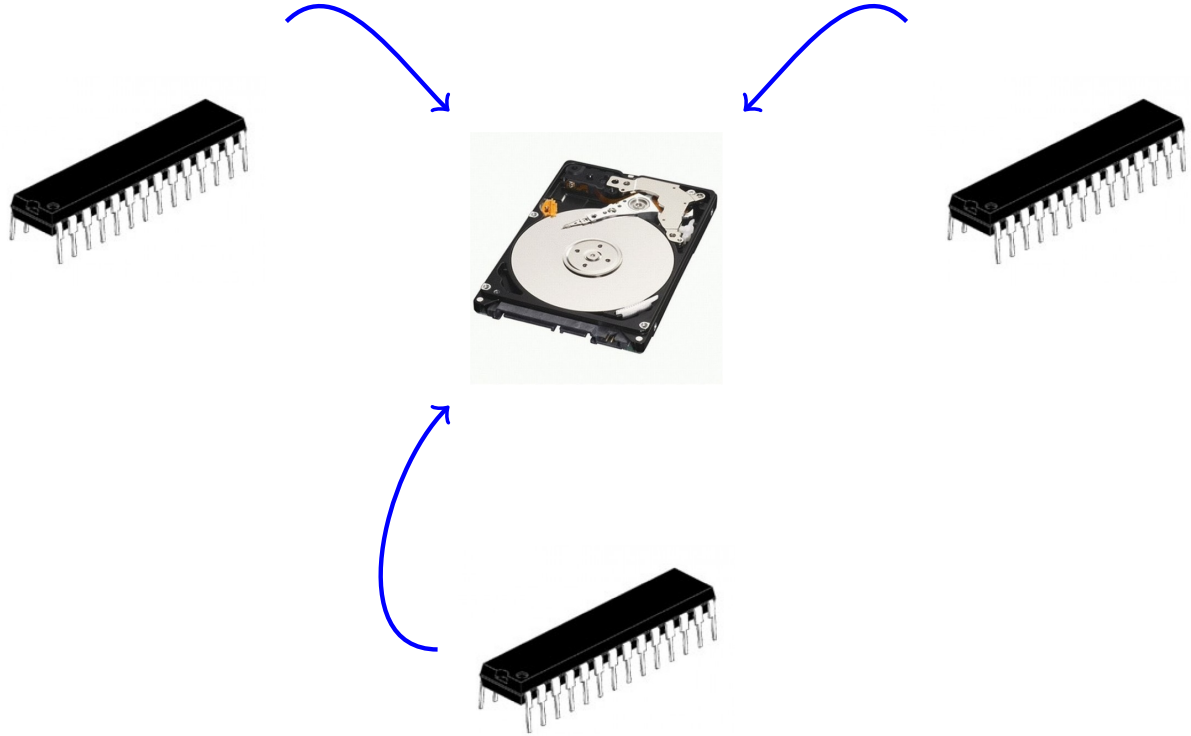



Modèle

Système à mémoire partagée et opérations d'écriture/lecture immédiate.

Retour aux algorithmes

Travailler à plusieurs c'est plus sympa

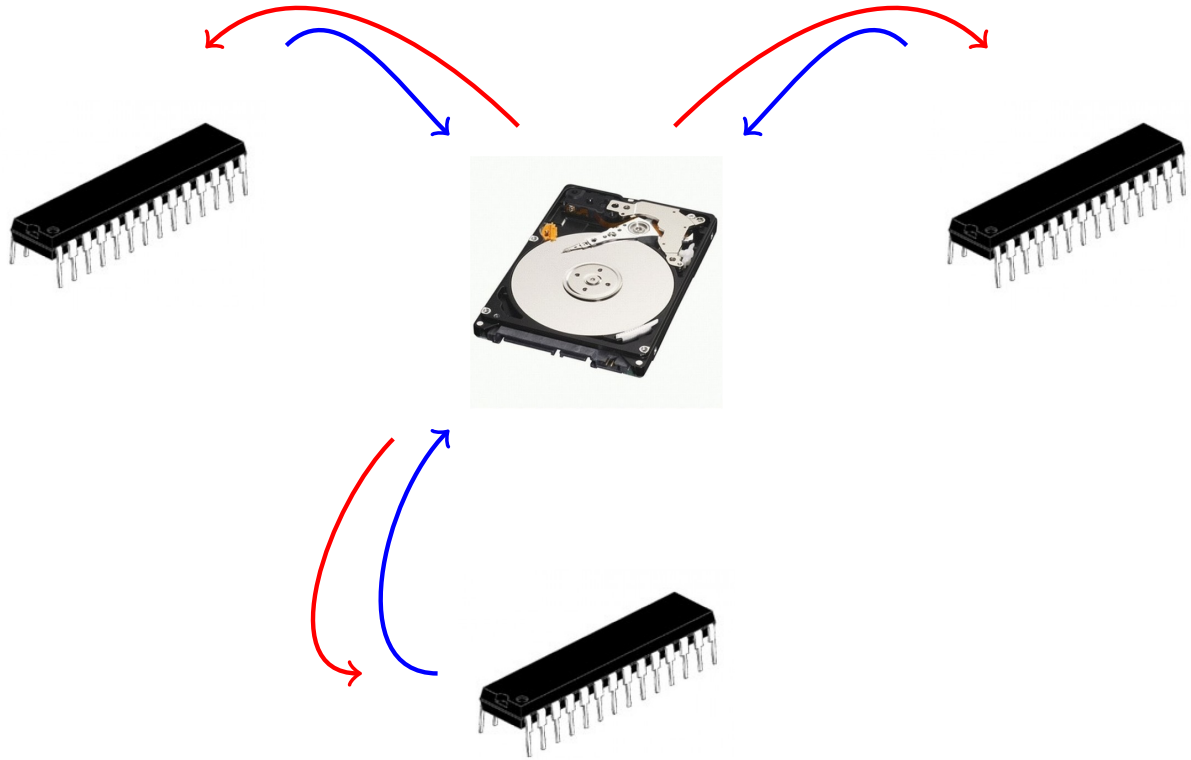


Modèle

Système à mémoire partagée et opérations d'écriture/lecture immédiate.

Retour aux algorithmes

Travailler à plusieurs c'est plus sympa

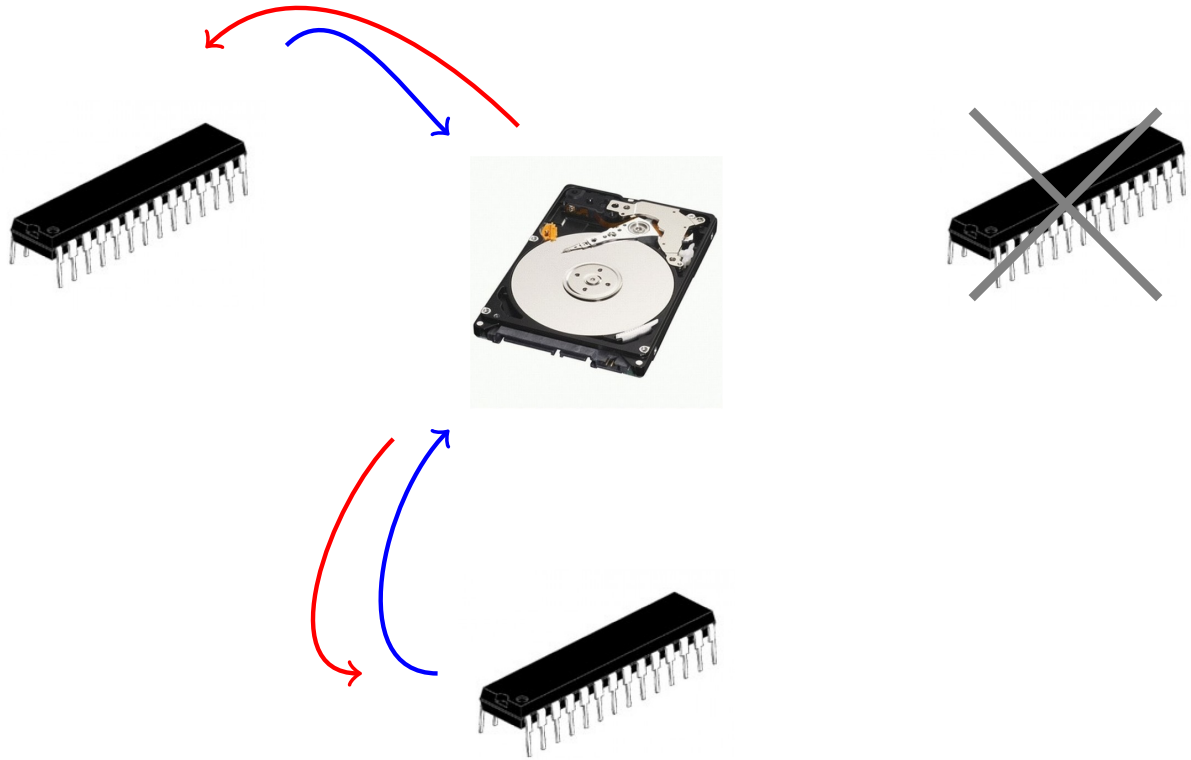


Modèle

Système à mémoire partagée et opérations d'écriture/lecture immédiate.

Retour aux algorithmes

Travailler à plusieurs c'est plus sympa

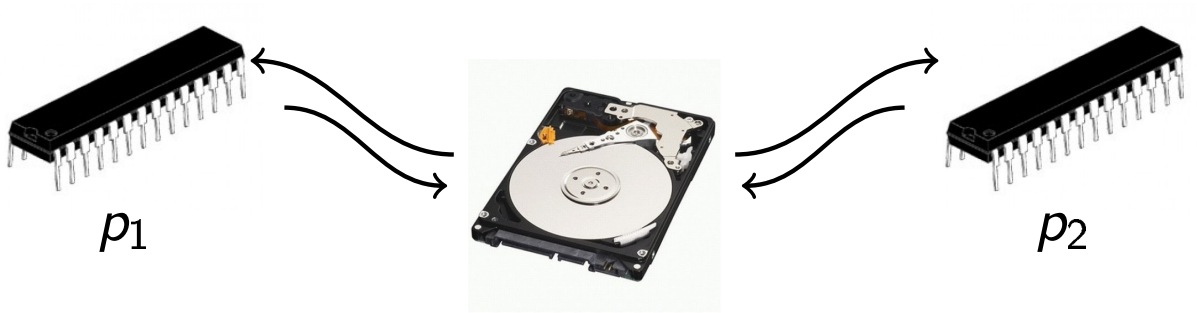


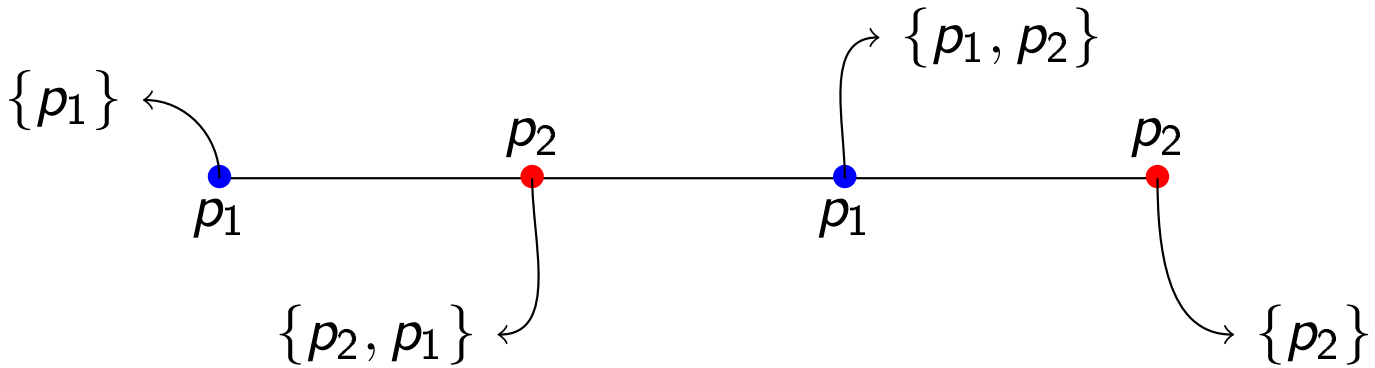
Modèle

Système à mémoire partagée et opérations d'écriture/lecture immédiate.

Communication et complexes simpliciaux

Une représentation topologique de la communication

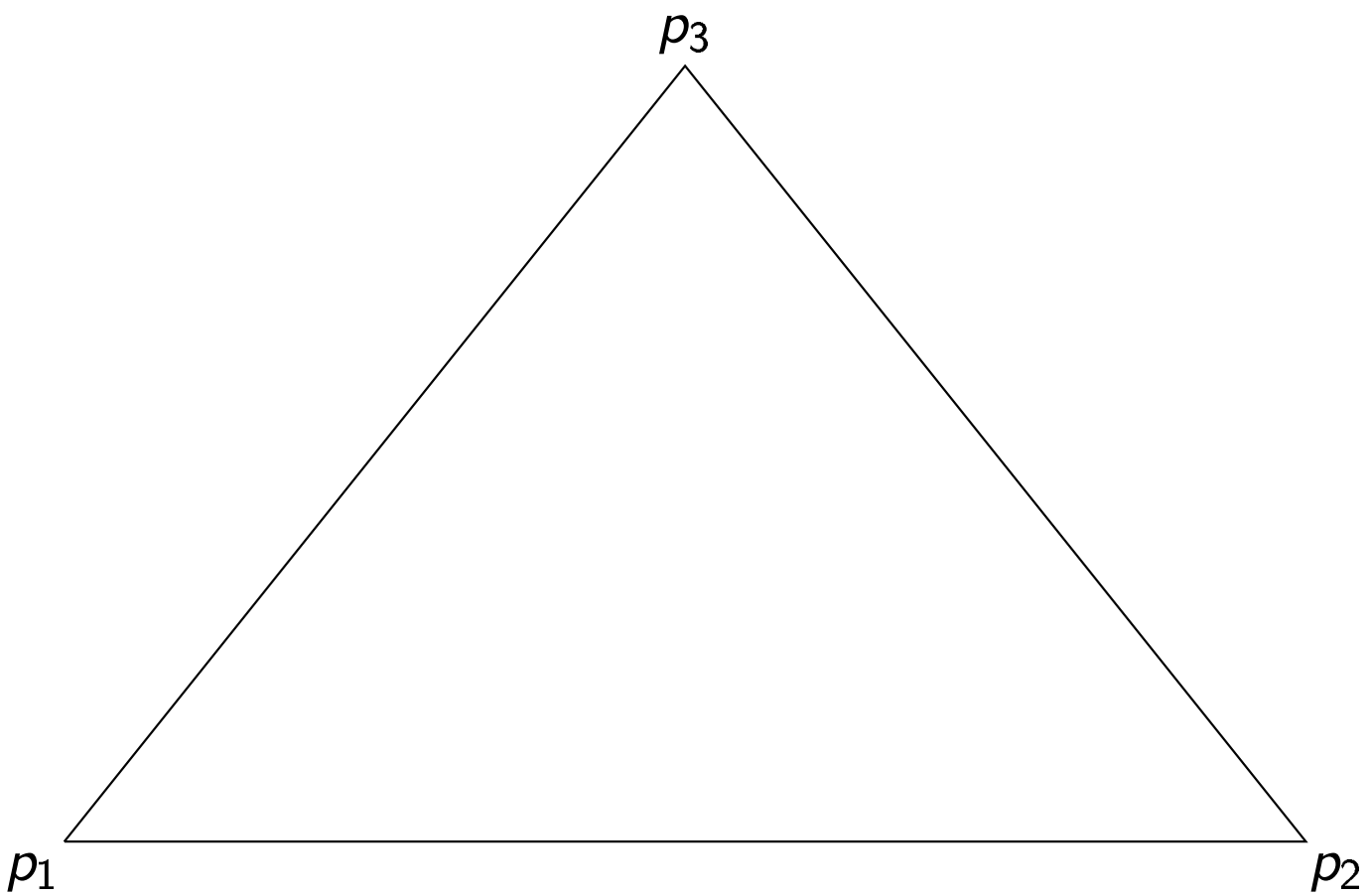




(Complexe de protocole de dimension 1)

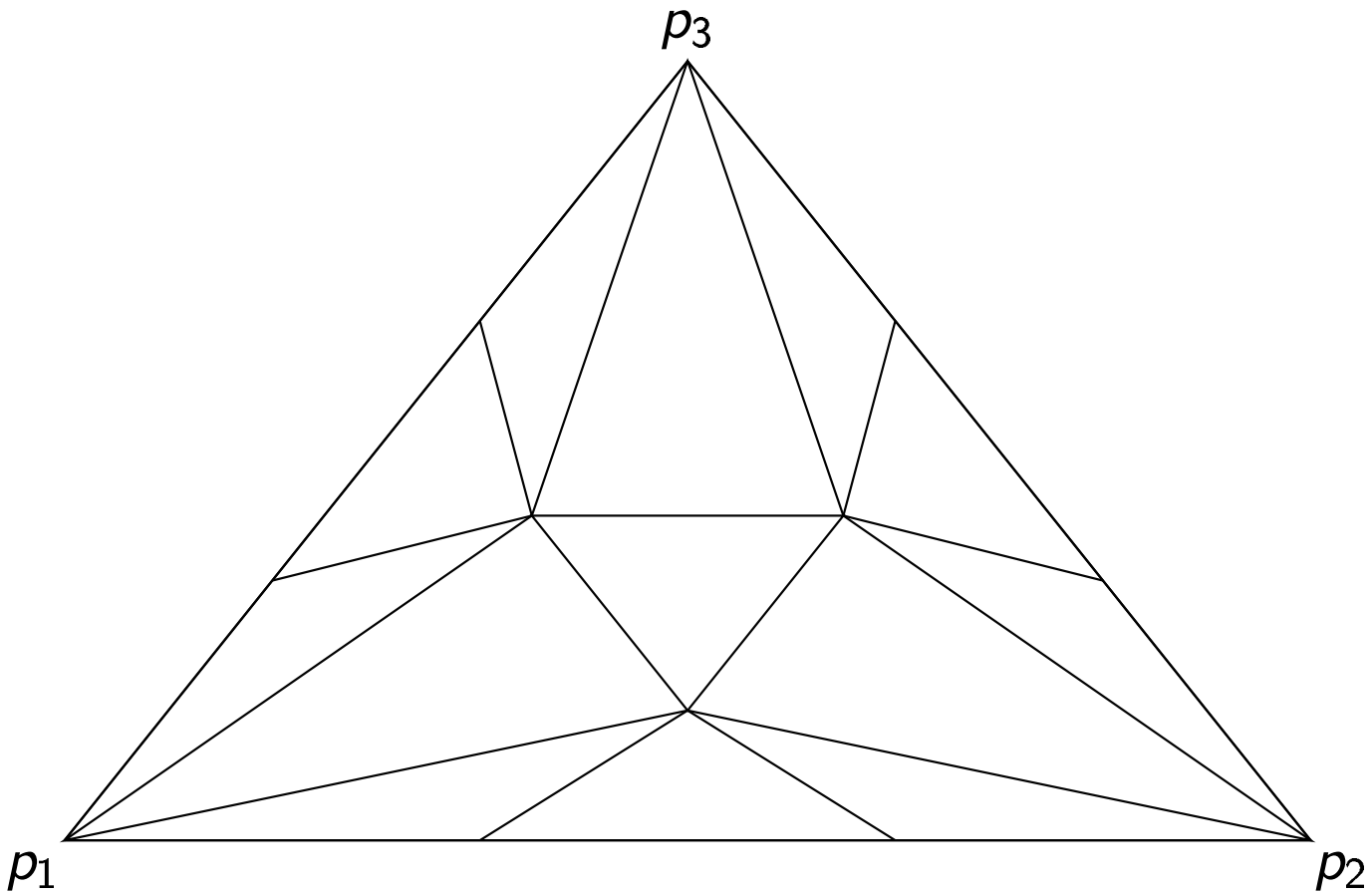
Complexe de protocole

Dimension 2



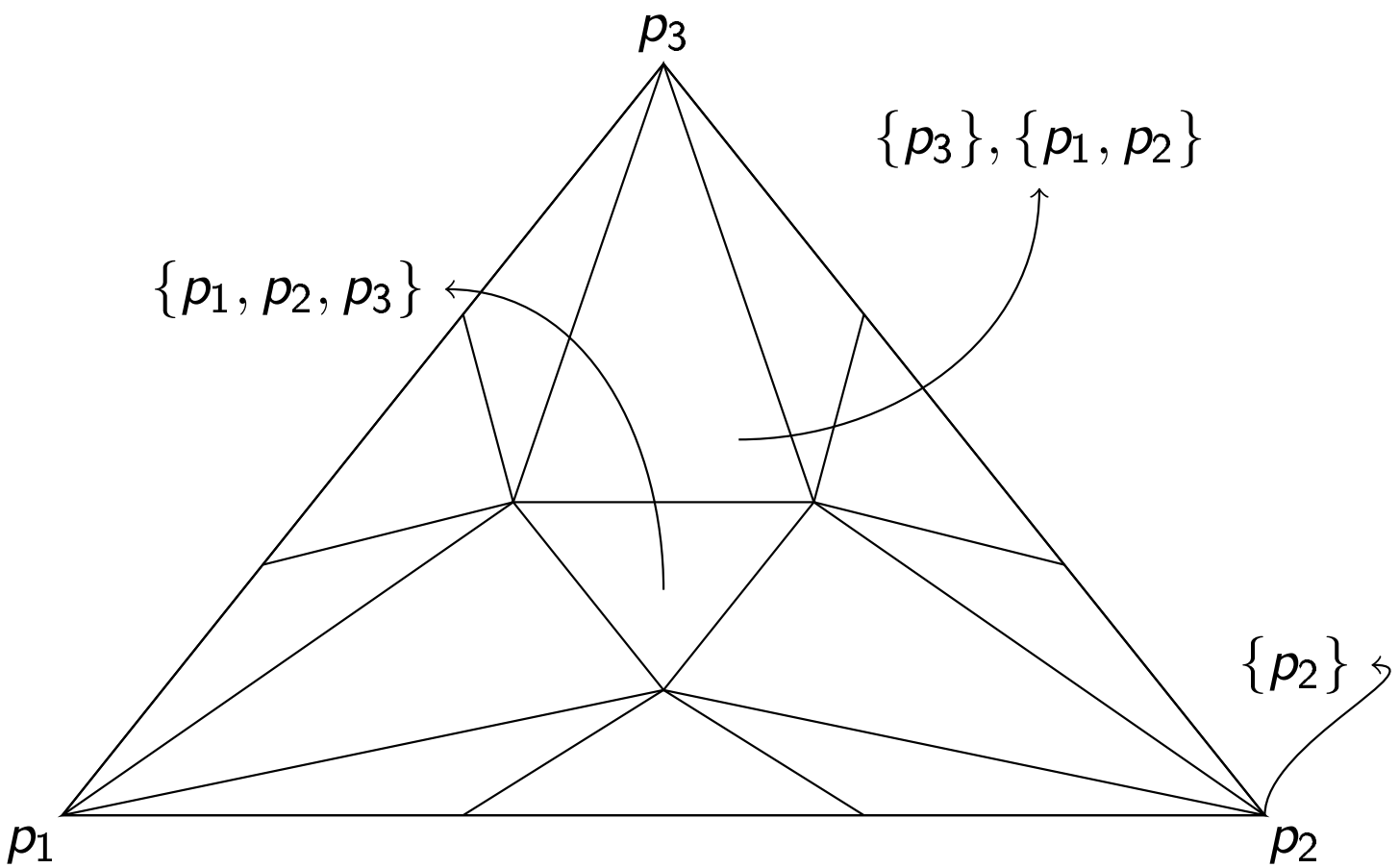
Complexe de protocole

Dimension 2



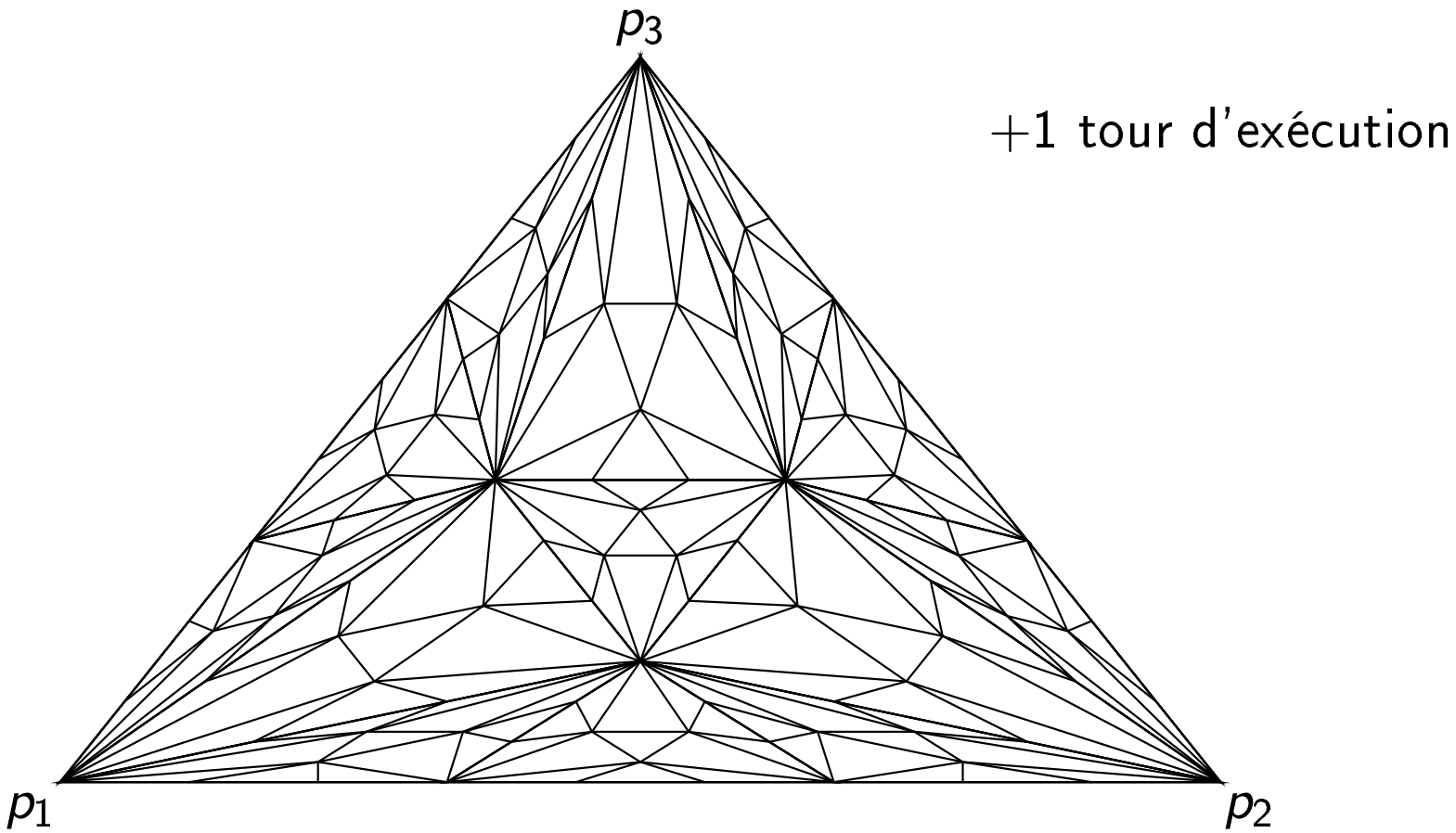
Complexe de protocole

Dimension 2



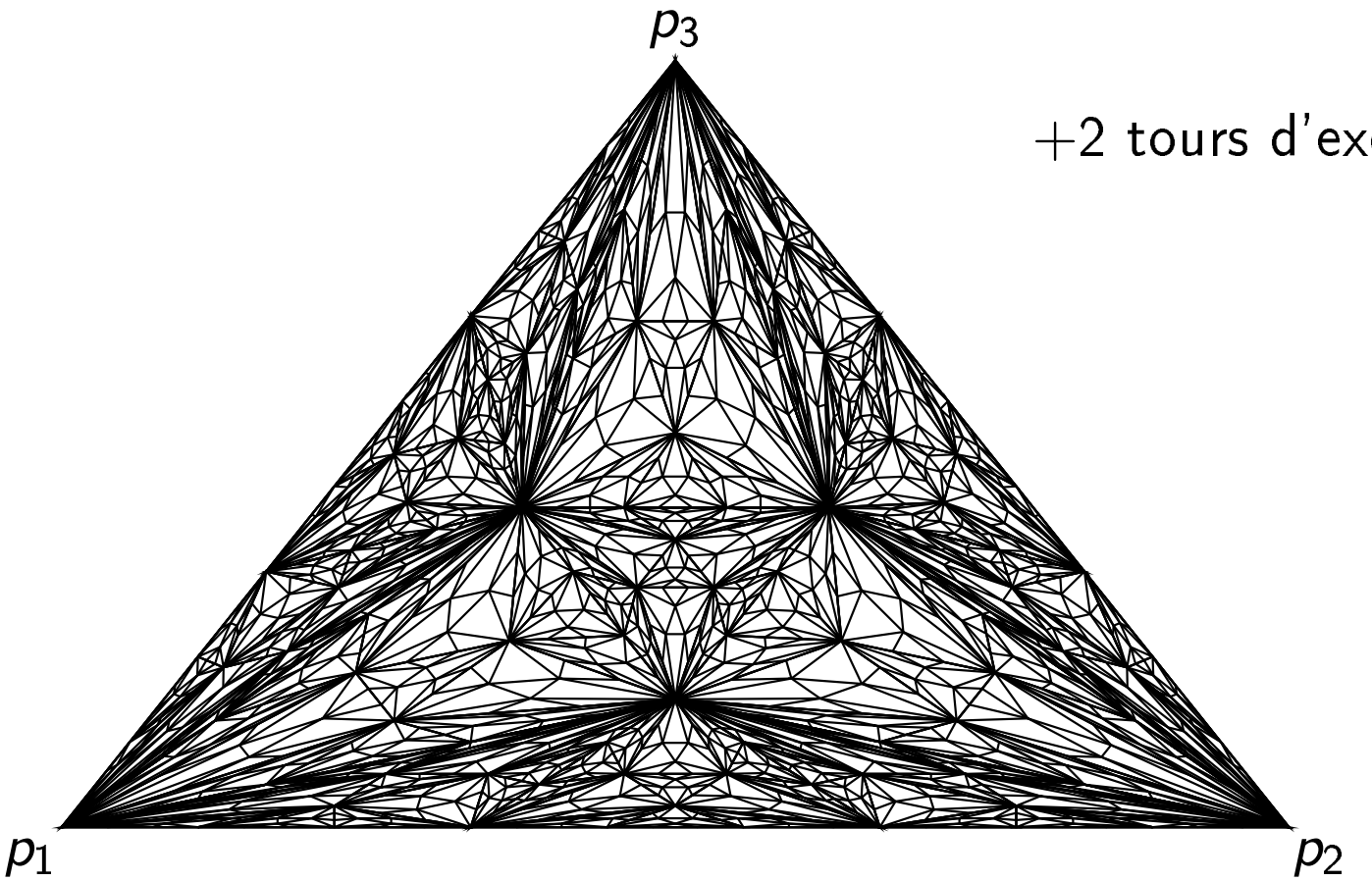
Complexe de protocole

Dimension 2



Complexe de protocole

Dimension 2



+2 tours d'exécution

Calculabilité asynchrone

Théorème (Herlihy-Shavit 1999)

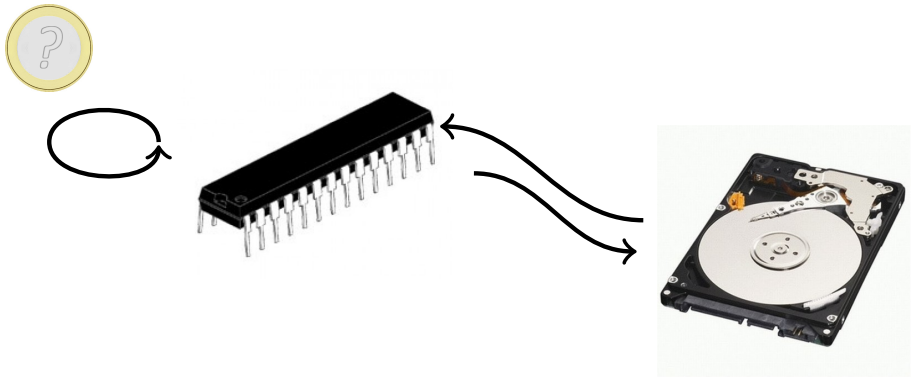
Algorithme qui résout une tâche $\Leftrightarrow$ Propriétés topologiques du complexe de protocole

Applications

- Le consensus (tous les processus retournent une valeur commune)
  $\rightarrow$ n'admet pas d'algorithme
- L'accord d'ensemble k (les processus retournent une valeur parmi un ensemble de k valeurs)
  $\rightarrow$ n'admet pas d'algorithme

Algorithmes probabilistes

Certaines tâches, comme le consensus et l'accord d'ensemble k admettent pourtant une solution si l'on introduit des probabilités dans le modèle.



- Ben-Or (1983)
- Aspnes-Herlihy (1990)
- Chor-Israeli-Li (1994)
- Mostefaoui-Raynal (2001)
- ...

Question

La topologie combinatoire peut-elle nous apprendre quelque chose sur ces algorithmes ?

Oui

(La topologie combinatoire peut nous apprendre quelque chose sur les algorithmes probabilistes)

Définitions

- $q_{r,k}$ est la probabilité qu'un algorithme n'atteigne **pas** l'accord d'ensemble k au bout de r tours.

  $(\rightarrow k = 1$ (consensus) : Attiya-Censor 2010)
- $f(n)$ est le nombre de faces du complexe de protocole de dimension $n-1$

Théorème (section 4.5.3.2)

$$q_{r,k} \geq \frac{1}{(f(k))^r}$$

Et alors ?

Résumé

- Le développement de Taylor est un outil puissant, entre syntaxe et sémantique, qui s'adapte à une grande variété de systèmes.
- Une étude géométrique et combinatoire des réseaux de preuve permet d'obtenir des résultats syntaxiques et sémantiques
- La topologie combinatoire peut être adaptée à un modèle probabiliste et mener à des résultats sur les probabilités d'échec

Perspectives

- Développement de Taylor pour le λ-calcul et les réseaux infinitaires
- Réseaux pour l'Appel-Par-Pousse-Valeur
- Appliquer les nouvelles méthodes d'analyse de ressources à la complexité algorithmique
- Complexes de protocoles probabilistes pour d'autres tâches, d'autres résultats.

Appel-Par-Pousse-Valeur

Λ_{pv}

$$M, N ::= x \mid \lambda x M \mid \langle M \rangle N \mid (M, N) \mid \pi_i(M) \mid \iota_i(M) \mid$$
$$\mathbf{case}(M, y \cdot N_1, z \cdot N_2) \mid M^! \mid \mathbf{der}(M) \mid \mathbf{fix}_x(M)$$

- Subsume l'Appel-Par-Nom et l'Appel-Par-Valeur au niveau des sémantiques opérationnelle et dénotationnelle.
- Résultats en sémantique quantitative (notamment extensions probabilistes)

Duplication : Exponentielles VS Morphismes de coalgèbre

Types positifs : $A ::= !I \mid A \otimes A \mid A \oplus A$
Types généraux : $I ::= A \mid A \multimap I \mid \top$

	Λ	Λ_{pv}
Application	MN	$M(V_1, V_2)$
Approximation	$\langle m \rangle [n_1, \ldots, n_k]$	$\langle m \rangle (v_1, v_k)$
Interprétation	$N : !A$	$(V_1, V_1) : P_1 \otimes P_2$

$$P_1 \otimes P_2 \xrightarrow{h} (P_1 \otimes P_2) \otimes \ldots\ldots \otimes (P_1 \otimes P_2)$$

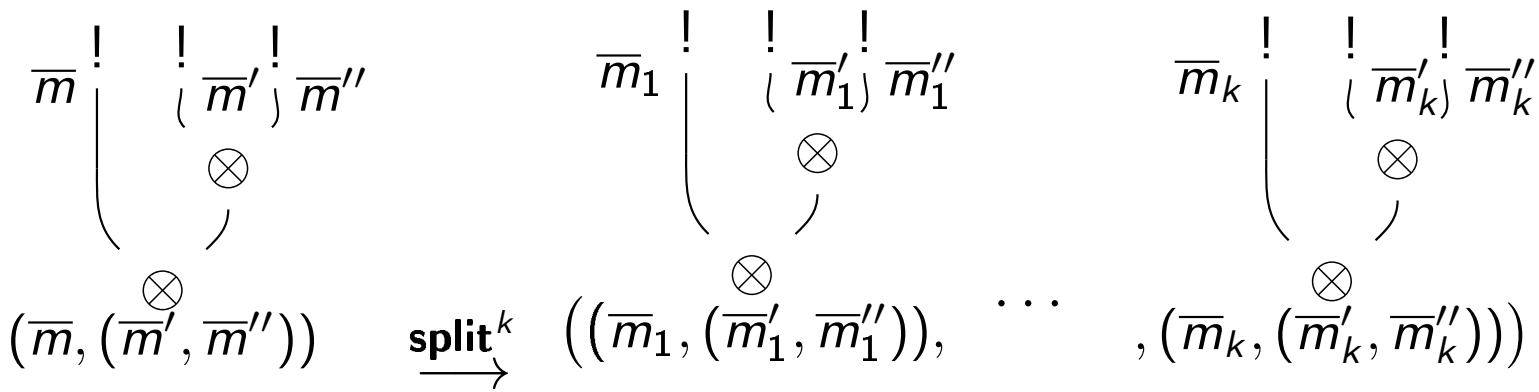
h est un morphisme issu de la structure de coalgèbre des types positifs Si la

duplication existe dans la sémantique, *quid* du développement de Taylor syntaxique ?

Réduction de ressources avec scission des arguments

Opérateur de découpage « split »

$\mathbf{split}(([m,m],[m,m])) = ([m],[m]),([m],[m])$
En général : $\mathbf{split}^k(v) = (v_1,\ldots,v_k)$ où les v_i ont le même arbre syntaxique que v.



Réduction pour Δ_{pv}

$$\langle \lambda x M\rangle V \to \sum_{(v_1,\ldots,v_k)\in \mathbf{split}^k(v)} M[v_1/x_1,\ldots,v_k/x_k]$$