Notion de préfixe dans la complexité de Kolmogorov et les modèles de calcul

Qu’est-ce que le calcul ?

Le calcul, c’est pouvoir faire des choses de façon mécanique, sans ambiguïté. Le calcul doit être reproductible à volonté, et ne pas faire appel à l’intelligence.

Illustration 1 (Borel, 1912) ≪ Je laisse intentionnellement de côté la plus ou moins grande longueur pratique des opérations ; l’essentiel est que chacune de ces opérations soit exécutable en un temps fini, par une méthode sûre et sans ambiguïté ≫

Trois visions différentes de la notion de calcul :

Vision logique fonctions récursives de Church-Rosser, formules ;

Vision par réécriture Post, Markov, puis machines de Kolmogorov-Uspensky ;

Vision machiniste Définition par Turing de la machine de Turing, puis par von Neumann des ordinateurs modernes.

- ≪ Thèse de Church ≫ : Ces différentes formalisations conduisent toutes à une même notion de ≪ fonction calculable ≫ qui correspond à l'intuition.

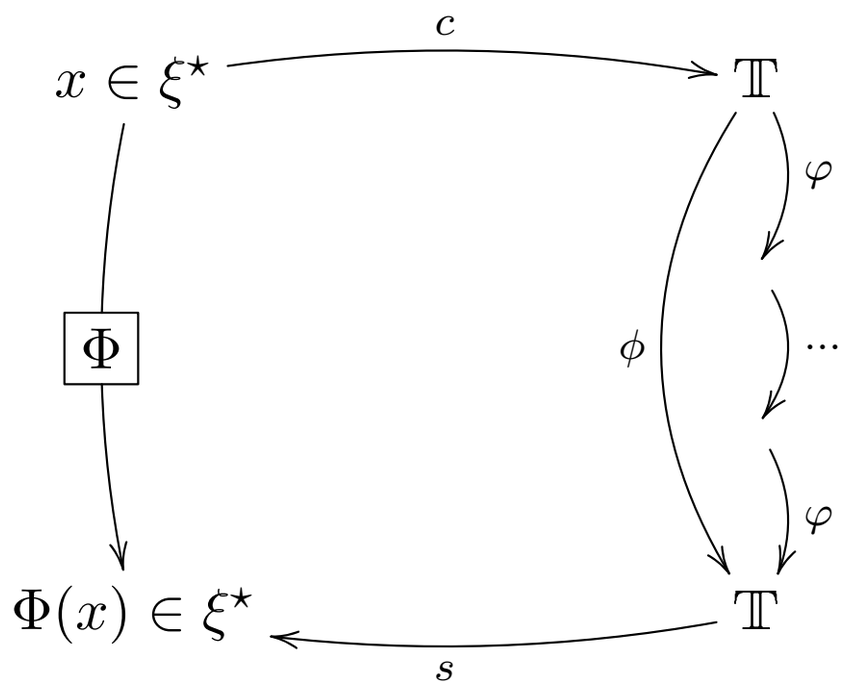
Modèles de calcul

Il est nécessaire de formaliser la façon dont le calcul est fait.

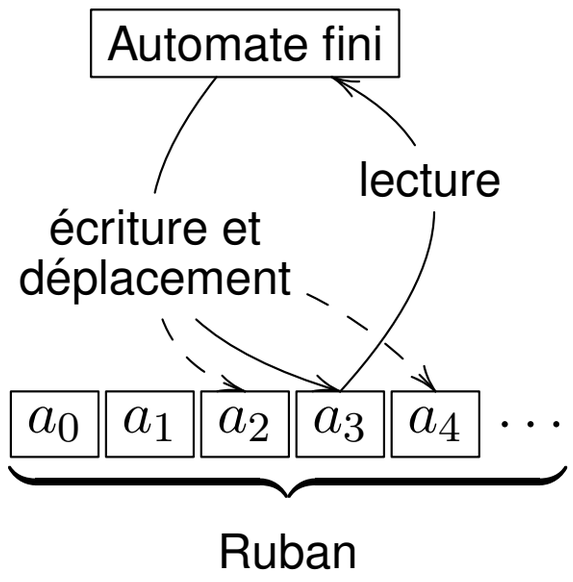
Définition classique (réécriture) : réagir toujours de la même façon face à la même situation locale, et stocker des données *arbitrairement nombreuses*, effectuer le calcul en un nombre d'étapes *arbitrairement grand*.

$\Rightarrow$ Il y a passage à l'infini pour l'espace comme pour le temps.

Observation des machines réelles : codage pour ≪ plonger ≫ l'entrée dans un espace de calcul infini.



La machine de Turing



— Itération d'un *automate fini* sur un ruban ;

— Ruban *infini*, utilisant un alphabet *fini*.

$\Rightarrow$ Modèle pour le calcul.

Problèmes afférents aux modèles de calcul :

Calculabilité ≪ Est-ce calculable ? ≫

Complexité ≪ Est-ce calculable avec des ressources limitées ? ≫

Algorithmique ≪ Comment est-il possible de faire ? ≫

Illustration 2 (Théorème de l'arrêt) La fonction caractéristique des machines qui s'arrêtent n'est pas calculable.

Qu’est-ce que l’information ?

Un mot contient de l'information. Qu'est-ce qui contient le plus d'information, entre 0802802802, 0478033333 et 0231936224 ?

$\Rightarrow$ Techniques de compression exactes. Rapport avec l'aléatoire.

Illustration 3 (Nombres aléatoires ?) La suite de chiffres $0{,}123456789101112131 4\ldots$ (nombre de Champernowe) est facile à fabriquer. La suite des chiffres de $\pi = 3{,}14159265\ldots$ l'est aussi.

La suite des résultats du lancer de pièces à pile (0) ou face (1), en revanche, est compliquée à décrire.

Définition 1 (Complexité de Kolmogorov – 1965)

$$\mathbf{K}_\psi(x) = \min\{\ell(p), \psi(p) = x\}.$$

$\Rightarrow$ longueur d'un plus petit programme qui construit un objet x.

Définition 2 (Additivité optimale) Soit $\mathcal{C}$ une classe de machines. Il existe une machine $\psi \in \mathcal{C}$ vérifiant :

$$\forall \phi \in \mathcal{C}, \exists c_\phi, \forall x \in \Xi, \quad \mathbf{K}_\psi(x) \leqslant \mathbf{K}_\phi(x) + c_\phi$$

Si $\mathcal{C}$ est l'ensemble de toutes les machines de Turing, alors ψ existe et on note la complexité $\mathbf{K}_\psi = \mathbf{KS}$ (théorème de Solomonoff-Kolmogorov).

Propriétés de base

— Changement de référence $\Rightarrow$ constante dans les inégalités ;

— Au pire, écrire brutalement :

$$0 \leqslant \mathbf{KS}(x) \leqslant \ell(x);$$

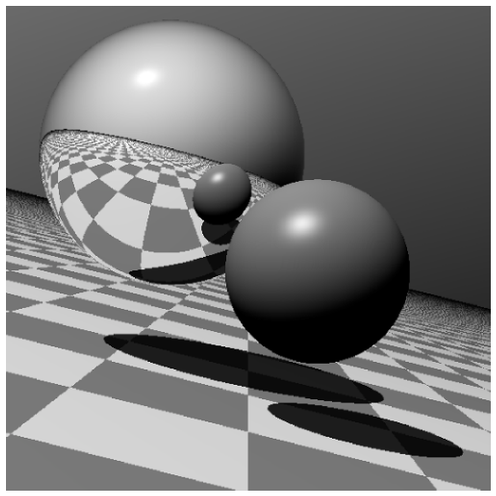
— Non-calculable ;

— Possibilité de borner par au-dessus ;

— Mesure de l'aléatoire d'un mot fini

$\Rightarrow$ taux de compression $\sim$ quantité d'aléatoire.

— Compression à plusieurs étages ;



```
%! Tiny RayTracing by HAYAKAWA,Takashi(h-takasi@isea.is.titech.ac.jp)
/p/floor/S/add/A/copy/n/exch/i/index/J/ifelse/r/roll/e/sqrt/H{count 2 idiv exch
repeat}def/q/gt/h/exp/t/and/C/neg/T/dup/Y/pop/d/mul/w/div/s/cvi/R/rlineto{load
def}H/c(j1idj2id42rd)/G(140N7)/Q(31C85d4)/B(V0R0VRVC0R)/K(WCVW)/U(4C577d7)300
T translate/I(3STinTinTinY)/l(993dC99Cc96raN)/k(X&E9!&1!J)/Z(blxC1SdC9n5dh)/j
(43r)/O(Y43d9rE3IaN96r63rvx2dcaN)/z(&93r6IQO2Z4o3AQYaNlxS2w!)/N(3A3Axe1nwc)/W
270 def/L(1i2A00053r45hNvQXz&vUX&UOvQXzFJ!FJ!J)/D(cjS5o32rS4oS3o)/v(6A)/b(7o)
/F(&vGYx4oGbxSd0nq&3IGbxSGY4Ixwca3AlvvUkbQkdbGYx4ofwnw!&vlx2w13wSb8Z4wS!J!)/X
(4I3Ax52r8Ia3A3Ax65rTdCS4iw5o5IxnwTTd32rCST0q&eCST0q&D1!&EYE0!J!&EYEY0!J0q)/V
1 def/x(jd5o32rd4odSS)/a(1CD)/E(YYY)/o(1r)/f(nY9wn7wpSps1t1S){[n{( )T 0 4 3 r
put T(/)q{T(9)q{cvn}{s}J}{($)q{[}{]}J}J cvx}forall]cvx def}H K{K{L setgray
moveto B fill}for Y}for showpage
```

Les codes préfixes

Le codage, qu'est-ce que c'est ?

$\Rightarrow$ C'est associer une représentation à une source.

$\Rightarrow$ Par contre, tout codage n'est pas *uniquement décodable*.

Illustration 4 Code $x \rightarrow x\star$: On ne peut pas séparer les mots codés.

Comment décoder le mot $xon \star vi\star$?

$\Rightarrow$ Transmission d'un fichier : comment marquer la fin ?

$\Rightarrow$ Comment séparer plusieurs éléments concaténés ?

On note Ξ l'ensemble des mots.

Définition 3 (Ordre préfixe)

$x < y$ si et seulement si $\exists u \in \Xi$ tel que $y = x \cdot u$.

Illustration 5 (Comparaisons)

- `Pierre` est plus petit que `Pierrette`.
- `Jean` et `Matthieu` sont incomparables.
- Le mot vide (ε) est plus petit que tout autre mot.

— Les codages de deux mots ne sont jamais comparables : le code est dit *préfixe*.
— Dans un code préfixe, déchiffrement immédiat.
— $(x, y) \to C(x) \cdot C(y)$ injective (extensibilité).
— Les codes préfixes ont des propriétés combinatoires intéressantes, comme la quantification de la répartition par l'inégalité de Kraft.

Illustration 6 (Quelques codes préfixes) La réécriture de 0 en 00, de 1 en 11, le tout suivi d'un 01 est un code préfixe.

$C(x) = 1^{\ell(x)} \cdot 0 \cdot x$ est un code préfixe classique. De même pour $C(x) = 1^{\ell(\ell(x))} \cdot 0 \cdot \ell(x) \cdot x$.

$\Rightarrow$ codes autodélimités.

- `Christ` est un *sous-mot* de `Jean-Christophe`
- Codes sans facteurs : aucun mot de code ne doit être un sous-mot d'un autre mot de code.

$\Rightarrow$ Code sans facteur $\Rightarrow$ Code préfixe.

Utiliser la notion de préfixe

Pour les mots finis, on a équivalence entre incompressibilité et aléatoire.
Cette démarche ne s'étend pas bien aux mots infinis.

On utilise des machines préfixes, dont l'entrée est interpretée comme étant un code préfixe.

- Utilisation pour caractériser l'aléatoire sur une suite infinie (théorème de Levin-Schnorr).

$\Rightarrow$ Un mot infini $\omega = \omega_1\omega_2...\omega_n...$ est aléatoire ssi la complexité *préfixe* de $\omega_{1:n}$ est plus grand que n.

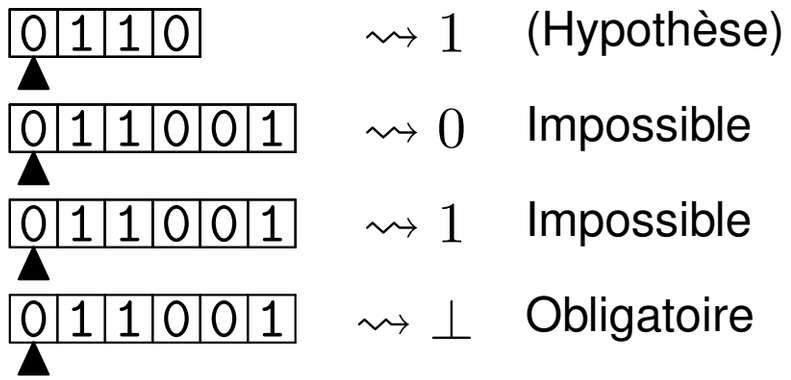
Illustration 7 (Nombre ω de Chaitin) Contrairement au réel π, les décimales de ce nombre lié à la probabilité d'arrêt d'une machine de Turing sont aléatoires.

Les machines préfixes

$M(x)\downarrow$ signifie M converge sur x $\quad$ $\perp$ signifie *non-convergence*

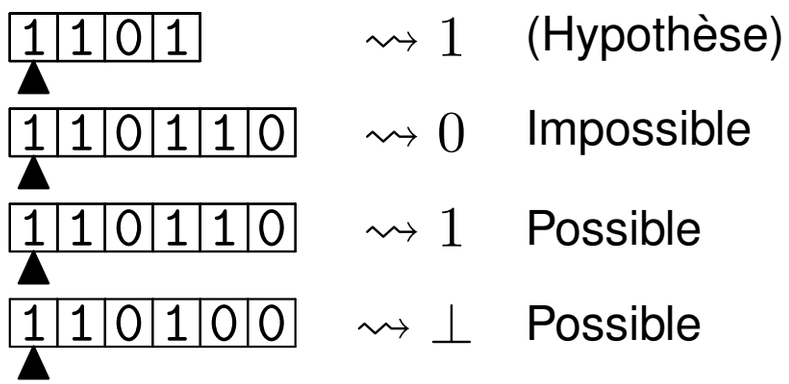
Définition 4 (Classe MPC) Une machine M appartient à la classe des machines $\mathcal{MPC}$ si et seulement si :

$$M(u)\downarrow \text{ et } M(u\cdot v)\downarrow \Longrightarrow\ v=\varepsilon.$$



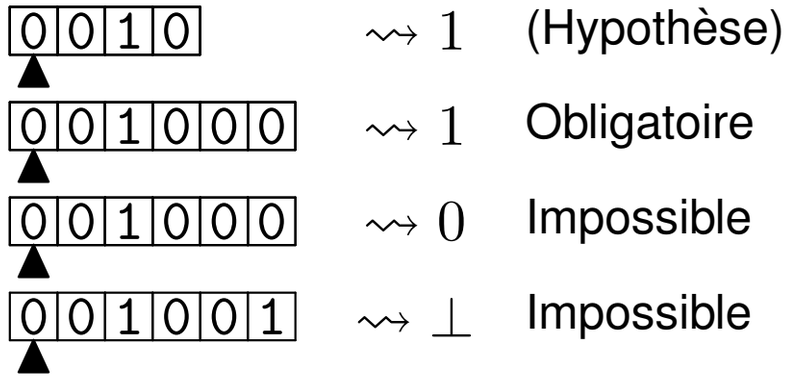
Définition 5 (Classe MPG) Une machine M appartient à la classe des machines $\mathcal{MPG}$ si et seulement si :

$$\left.\begin{array}{c} M(u)\downarrow \\ M(u\cdot v)\downarrow \end{array}\right\} \Rightarrow M(u) = M(u\cdot v).$$



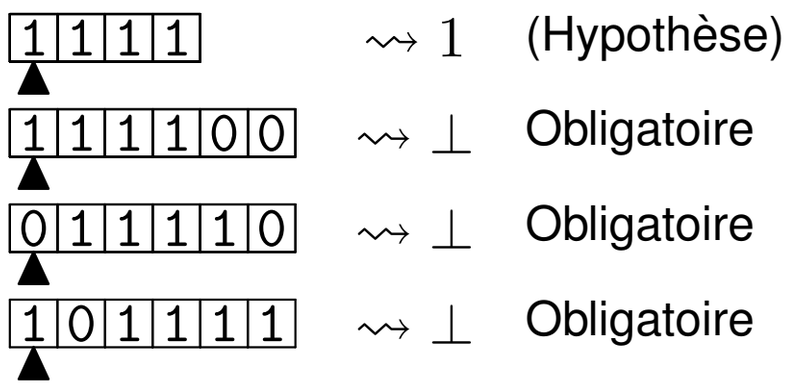
Définition 6 (Classe MPP) Une machine M appartient à la classe des machines $\mathcal{MPP}$ si et seulement si :

$$M(u)\downarrow \Rightarrow \forall v, M(u\cdot v)\downarrow \text{ et } M(u\cdot v) = M(u).$$



Définition 7 (Classe MDPC) Une machine M appartient à la classe des machines $\mathcal{MDPC}$ si et seulement si :

$$\left.\begin{array}{c} M(v)\downarrow \\ M(u\cdot v\cdot w)\downarrow \end{array}\right\} \Rightarrow u = w = \varepsilon.$$



$\Rightarrow$ Calcul dans le bruit naturel (machine préfixe pleine)

$\Rightarrow$ Recherche de l'extremité ou des extremités (machine préfixe creuse et doublement préfixe)

Alphabet : $0, 1, ..., 9, +, \star, -, /$

...2/3+7-45+6-42+216+1-3+16/4-4...

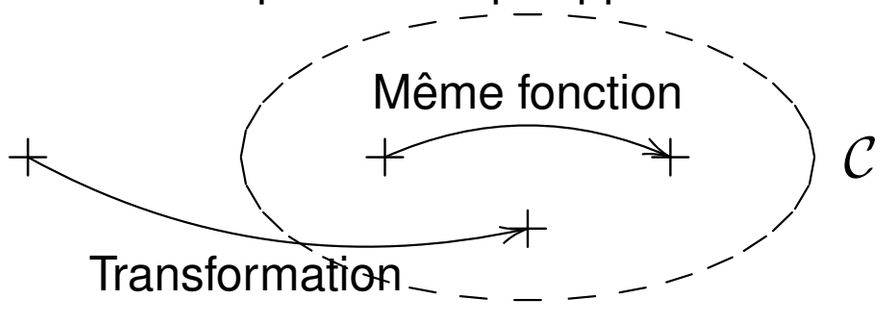
$\Rightarrow$ impossible de trouver le début de *notre* opération

...2/15++8-+++4+/2+/+/2+/1+/6++++1-++16/...

$\Rightarrow$ Si la tête commence dans le mot, elle trouve le bord !

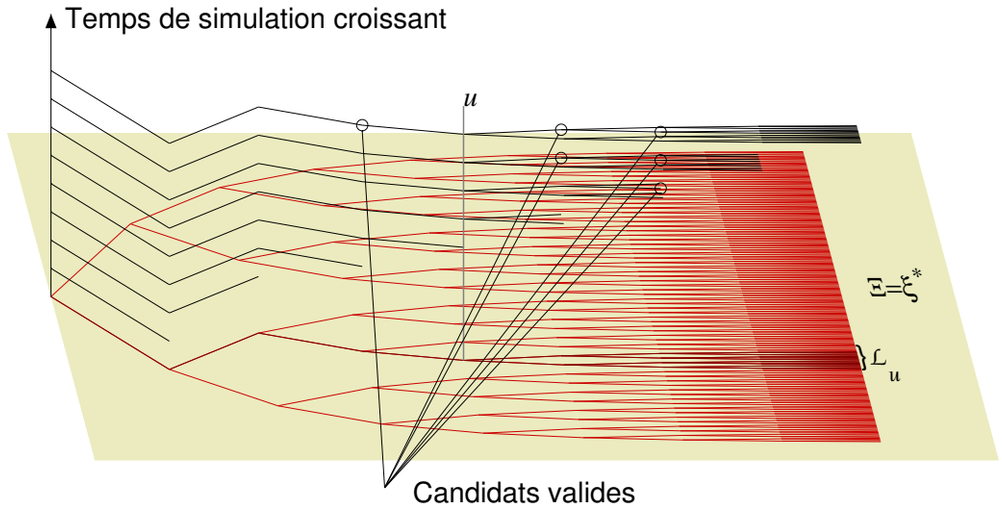
Qu'est-ce qui fait que le calcul est ce qu'il est ?

- Notions fondamentales : représentabilité, composition, universalité ;
- Théorèmes importants : *s-m-n* (composition et couplage), point fixe (*s-m-n* et universalité), Rice ;
- Outil : notion de projecteur sur $\mathcal{C}$ (programme qui transforme mécaniquement toute machine en une machine de $\mathcal{C}$, en laissant inchangé la fonction calculée par celles qui appartenaient déjà à $\mathcal{C}$).



Existence de projecteur

- u est l'entrée de la machine : essayer tous les préfixes et continuations de u, en temps croissant.



$\Rightarrow$ Décider quand u apparaît comme candidat accepté si on peut remplir

les règles de la classe considérée.

Résultats

— Notions fondamentales : oui, pour toutes les classes ;

  $\Rightarrow$ utilisation du projecteur.

— Théorèmes importants : oui, pour le point fixe et Rice ;

— Cas du *s-m-n* : sous certaines conditions sur le couplage employé, oui pour $\mathcal{MPC}$, $\mathcal{MPG}$ et $\mathcal{MPP}$;

— Question ouverte pour $\mathcal{MDPC}$.

La propriété **de l'additivité optimale** est vérifiée pour les machines préfixes ($\mathcal{MPC}$, $\mathcal{MPP}$, $\mathcal{MPG}$).

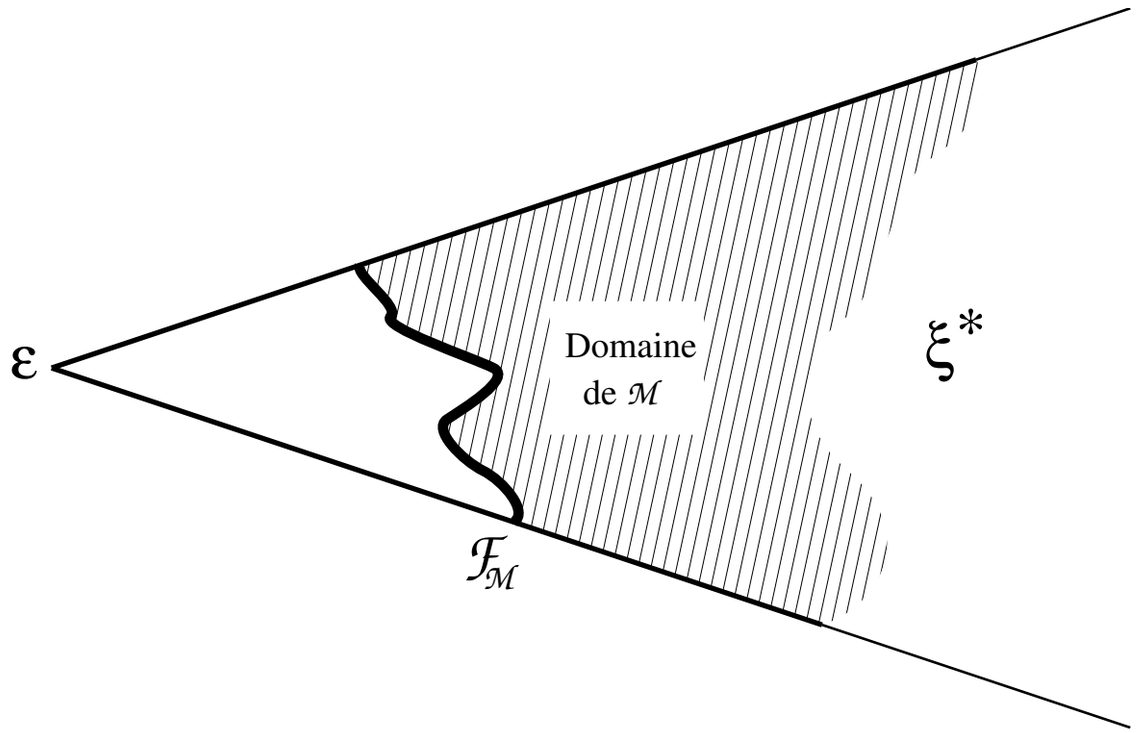
On peut donc définir trois classes de complexité : $\mathbf{KPC}$, $\mathbf{KPP}$ et $\mathbf{KPG}$.

Ces trois classes sont équivalentes.

$$\forall x, \quad \mathbf{KPG}(x) = \mathbf{KPP}(x) + \mathcal{O}(1) = \mathbf{KPC}(x) + \mathcal{O}(1)$$

Frontière

- Représentation en arbre :



Définition 8 (Frontière) Soit M une machine appartenant à $\mathcal{MPG}$, de domaine D. On appelle frontière de M et l'on note $\mathcal{F}(M)$ l'ensemble des mots qui sont minimaux pour l'ordre préfixe dans D. On note par $\mathcal{F}_{\hat{\mathcal{C}}}$ l'ensemble des frontières de toutes les machines de la classe $\hat{\mathcal{C}}$.

On a les inclusions suivantes :

$$\mathcal{F}_{\mathcal{MPC}} \lesssim \mathcal{F}_{\mathcal{MPG}} \qquad \mathcal{F}_{\mathcal{MPP}} = \mathcal{F}_{\mathcal{MPG}}$$

$\Rightarrow$ Il n'existe pas de transformation $\beta : \mathbb{N} \to \mathbb{N}$ telle que $\forall \phi_n \in \mathcal{MPP}$:

$$\phi_{\beta(n)}(x) = \phi_n(x) \text{si } \forall u, v, x = u \cdot v \Rightarrow \phi_n(u) = \perp$$

Et les machines doublement préfixes ?

Les deux équations suivantes sont vérifiées :

$$\forall \epsilon > 0, \exists f \in \mathcal{MDPC}, \exists c, \forall x \in \Xi,$$
$$\mathbf{K}_f(x) \leqslant (1+\epsilon)\mathbf{KS}(x) + c$$

$$\forall f \in \mathcal{MDPC}, \exists \epsilon > 0, \exists c, \forall x \in \Xi,$$
$$\mathbf{K}_f(x) \geqslant (1+\epsilon)\mathbf{KS}(x) + c$$

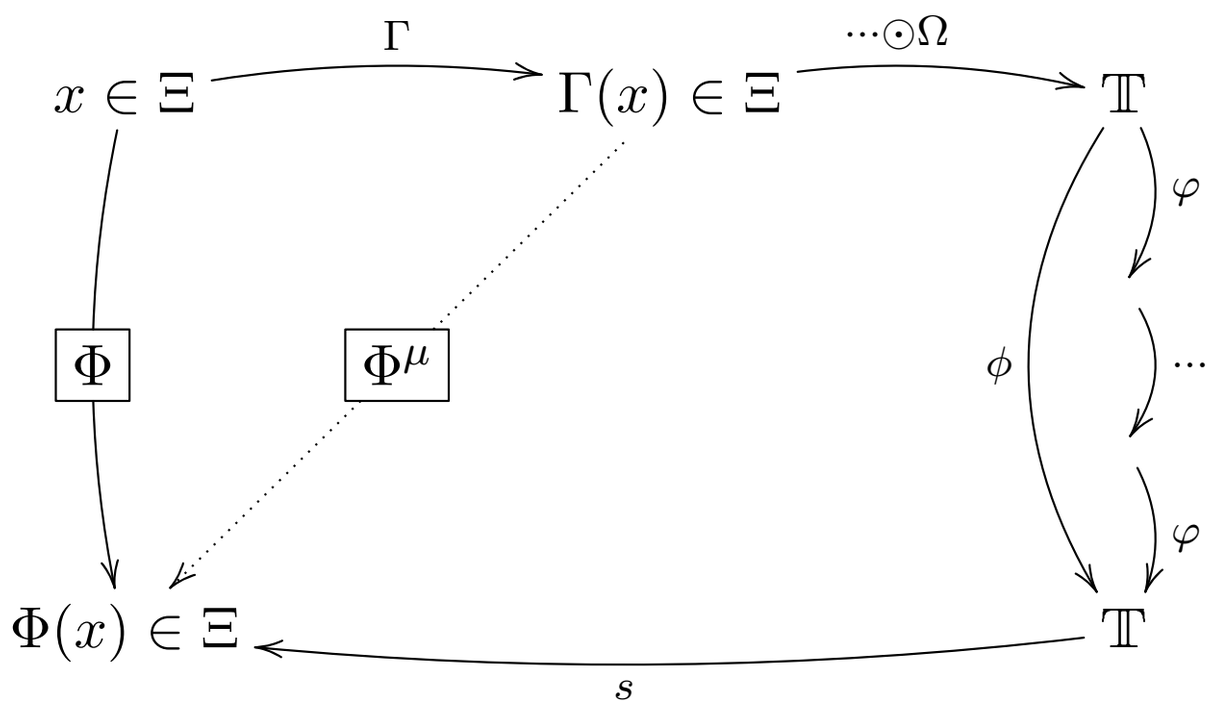
$\Rightarrow$ Il n'existe pas de machine additivement optimale pour la classe des machines doublement préfixe.

Modélisation préfixe du calcul

Le cas de la machine de Turing s'étend à beaucoup de variantes. Tous les calculs de ces variantes sont effectués dans un espace infini. La théorie de la calculabilité ne retient que les calculs à partir des mots finis.

- Étude des modèles minimaux de la machine de Turing (deux caractères et pas de blanc).

$\Rightarrow$ Sous certaines hypothèses sur le codage et le décodage, le plongement est forcément préfixe.



$\Rightarrow$ Dans certains cas, le plongement du mot d'entrée sur le ruban infini peut être vu comme un oracle.

Un oracle sur le ruban ?

- Théorie classique : on demande si un mot appartient à un langage, on obtient une réponse immédiate

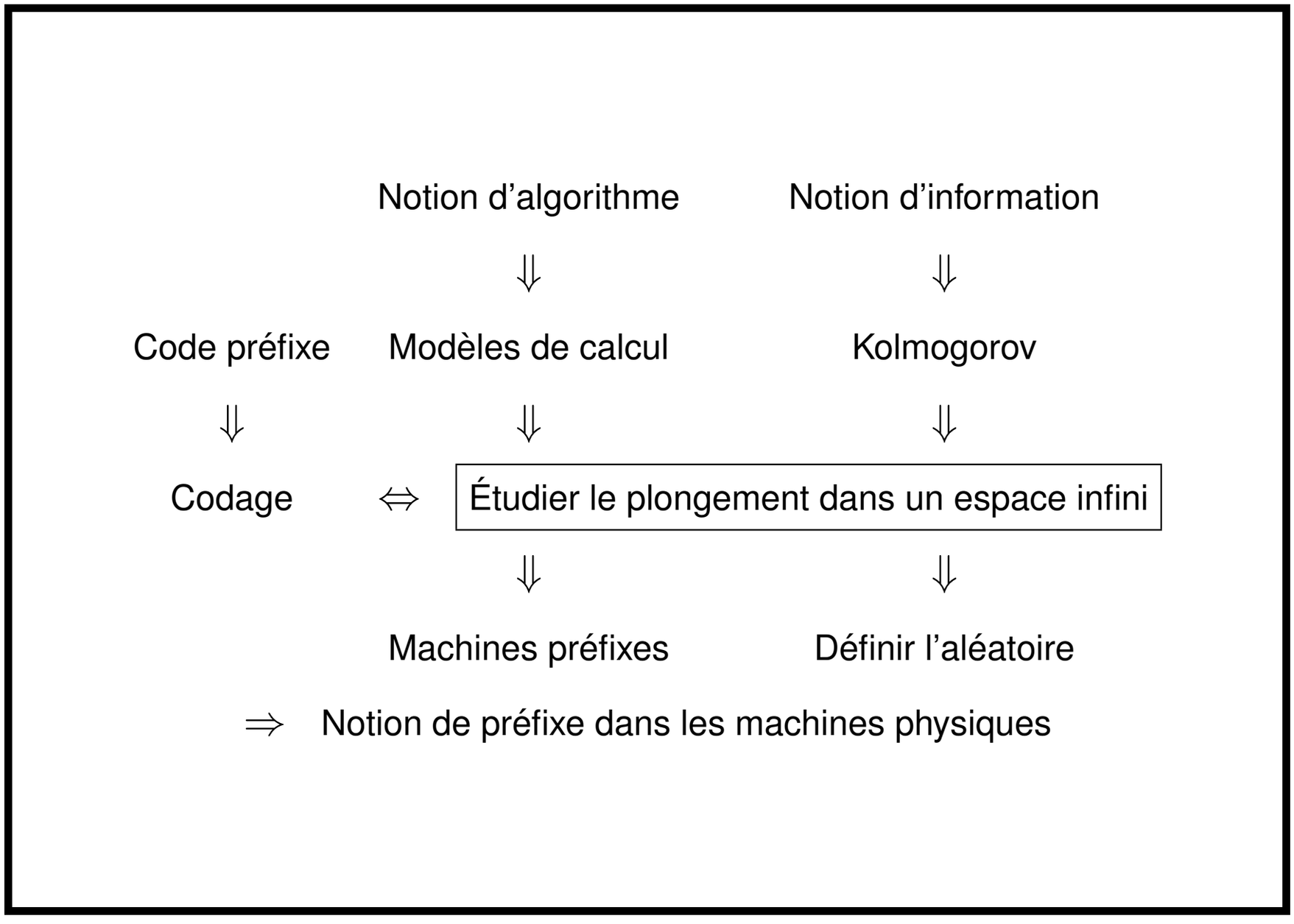
$\Rightarrow$ Existence d'une hiérarchie

- Théorie modifiée : on écrit la fonction caractéristique du langage sur le ruban.

$\Rightarrow$ Sous certaines conditions sur l'oracle, on obtient la même chose.

- Si $\ll$ l'oracle $\gg$ dépend de l'entrée, des conditions de cohérence imposent que chaque $\ll$ oracle $\gg$ soit de même niveau dans la hiérachie arithmétique.

Conclusion


Notion d'algorithme
Notion d'information
Code préfixe
Modèles de calcul
Kolmogorov
Codage
Étudier le plongement dans un espace infini
Machines préfixes
Définir l'aléatoire
Notion de préfixe dans les machines physiques

Conclusion

- Modélisation préfixe du calcul, simulation des oracles par le ruban infini.
- Application de la complexité de Kolmogorov à d'autres modèles.

$\Rightarrow$ Travail sur les automates cellulaires (*Kolmogorov complexity and CA classification*, Theor. Comp. Sci.)

- Développement de sous-classes de machines ayant une intégrité.

$\Rightarrow$ Recherche du point minimal entre machines préfixes et machines doublement préfixes vérifiant l'additivité optimale.