Modèles de calcul

Xavier Urbain

2020/2021

Motivations

TIME magazine en avril 1984, citant un éditeur de magazine sur le logiciel :

« Put the right kind of software into a computer, and it will do whatever you want it to. There may be limits on what you can do with the machines themselves, but there are no limits on what you can do with software. »

Motivations

Quelques définitions. . .

Quelques paradoxes :

- Zénon — infini ?
- Russel, Berry — existence ?

Proposition finitisme :

- Nombre fini d'objets,
- Nombre fini de règles de construction

$\rightsquigarrow \mathbb{R}$? . . . échec

Motivations

Quelques définitions. . .

Quelques paradoxes :

- Zénon — infini ?
- Russel, Berry — existence ?

Proposition formalisme :

- . . .
- \+ constructions telles que
  - Théorie obtenue cohérente — Pas d'obtention de φ et $\neg\varphi$

Programme de Hilbert (fondation des mathématiques)

On veut arithmétique non contradictoire + « vrai = prouvable » $\rightsquigarrow$ échec

Motivations

Procédure effective : texte fini sur alphabet fini indiquant sans ambiguïté les actions mécaniques à effectuer — sans réfléchir ni comprendre

Calculabilité :

- Preuve ?
- Calcul ?
- Construction ?

Déf. du mécanisme d'une preuve...

Pas tout faire ! ex : $\mathbb{N} \to \{0, 1\}$

Motivations

Amusons-nous :

$$\mathcal{B}(n) = \max\{[\mathtt{p}()] \mid \mathtt{p} \in \mathcal{L}, |\mathtt{p}| \leq n\}$$

($\mathcal{B}(1)$ en ocaml ?)

```
let b = (* code de ℬ, de taille 𝒯 *)

let c =
  let b = ··· in
  (b (5 * 𝒯)) + 1
```


Motivations

Trois grands modèles :

1. Machines de Turing (Turing 36, Post 36)
2. Fonctions récursives (Gödel 31, Kleene ∼40, Ackermann ∼40)
3. λ-calcul (Church ∼30)
   $\rightsquigarrow$ λ-calculs typés (Church ∼40...)

Machines de Turing

Alan Mathison Turing (1912–1954)

MT : automate + ruban (lect./écr.)

Machine de Turing $\mathscr{M}$: $(V, \mathtt{B}, Q, q_0, F, T)$

- V vocabulaire de $\mathscr{M}$
- $\mathtt{B} \in V$ caractère blanc
- Q états de $\mathscr{M}$
- $F \subseteq Q$ états finals
- $q_0 \in Q$ état initial
- $T : Q \times V \to V \times \{\lhd, \triangledown, \rhd\} \times Q$

Bande (ruban) : séquence infinie de symboles de V dont nombre fini $\neq \mathtt{B}$

Configuration : état + valeurs de bande + position de tête (w_1, q, w_2)

Bande : $\ldots \mathtt{B} w_1 w_2 \mathtt{B} \ldots$ Tête sur premier caractère de w_2

Machines de Turing — exécution

Exécution sur u $\rightsquigarrow$ Configuration de départ : (ε, q_0, u)

Sur $(v, q, \alpha w)$ (évt $\alpha = \mathtt{B}$ et $w = \varepsilon$)

- Si $T(q, \alpha) = (\beta, \rhd, q')$ alors $(v\beta, q', w)$
- Si $T(q, \alpha) = (\beta, \triangledown, q')$ alors $(v, q', \beta w)$
- Si $T(q, \alpha) = (\beta, \lhd, q')$ alors :
  - Si $v = v_1\gamma$ alors $(v_1, q', \gamma\beta w)$
  - Si $v = \varepsilon$ alors $(\varepsilon, q', \mathtt{B}\beta w)$
- Si $T(q, \alpha)$ indéfini : arrêt

C_1 à C_2 en un pas : $C_1 \rightarrow_{\mathscr{M}} C_2$

Arrêt sur C : $C \rightarrow_{\mathscr{M}} \perp$

Machines de Turing — langages

Pour $V_t \subset V$ (sans $\mathtt{B}$),
$\mathcal{L}_{V_t}(\mathscr{M}) = \{w \in V_t^\star \mid (\varepsilon, q_0, w) \rightarrow^\star_{\mathscr{M}} (u, q, v) \rightarrow_{\mathscr{M}} \perp, \quad q \in F\}$

q_0	B	$1 \rhd q_1$
	1	$1 \rhd \perp$
q_1	B	$\mathtt{B} \rhd q_2$
	1	$1 \rhd q_1$
q_2	B	$1 \lhd q_2$
	1	$1 \lhd q_0$

Machines de Turing — langages

Pour $V_t \subset V$ (sans $\mathtt{B}$),
$\mathcal{L}_{V_t}(\mathscr{M}) = \{w \in V_t^\star \mid (\varepsilon, q_0, w) \rightarrow^\star_{\mathscr{M}} (u, q, v) \rightarrow_{\mathscr{M}} \perp, \quad q \in F\}$

$$\mathscr{M} = (\overbrace{\{a, b\}}^{V_t} \cup \{X\}, \mathtt{B}, Q = \{q_0, \ldots, q_f\}, q_0, F = \{q_f\}, T)$$

$T =$

q_0	B	$\mathtt{B} \triangledown q_f$
	a	$a \rhd q_1$
q_1	a	$a \rhd q_1$
	b	$X \lhd q_2$
q_2	a	$X \rhd q_3$
	X	$X \lhd q_2$

q_3	X	$X \rhd q_3$
	b	$X \lhd q_2$
	B	$\mathtt{B} \lhd q_4$
q_4	X	$X \lhd q_4$
	B	$\mathtt{B} \triangledown q_f$

Machines de Turing — langages

Pour $V_t \subset V$ (sans $\mathtt{B}$),
$\mathcal{L}_{V_t}(\mathscr{M}) = \{w \in V_t^\star \mid (\varepsilon, q_0, w) \rightarrow^\star_{\mathscr{M}} (u, q, v) \rightarrow_{\mathscr{M}} \perp, \quad q \in F\}$

q_0	B	$1 \rhd q_1$
	1	$1 \lhd q_2$
q_1	B	$1 \rhd q_2$
	1	$1 \rhd q_1$

q_2	B	$1 \rhd q_3$
	1	$\mathtt{B} \lhd q_4$
q_3	B	$1 \lhd q_0$
	1	$1 \lhd q_3$
q_4	B	$1 \rhd \perp$
	1	$\mathtt{B} \lhd q_0$

47176870 étapes, 1 : 4098, 0 : 8191

Machines de Turing — décision

$w \in L?$ donnée : w, question : caract. de L

Problème de décision :

- Décidable si $\exists \mathscr{M}$ t.q. arrêt en temps fini $\forall w$ et sur acceptant ssi $w \in L$
- Semi-décidable si $\exists \mathscr{M}$ t.q. arrêt en temps fini et acceptant si $w \in L$
  non acceptant ou pas d'arrêt si $w \notin L$
- Co-semi-décidable si décision pour $\overline{L}$ semi-décidable

$\in L?$ décidable $\rightsquigarrow$ L récursif (réc.)
$\in L?$ semi-décidable $\rightsquigarrow$ L récursivement énumérable (r.é.)

$\rightsquigarrow$ L r.é. tel que $\exists \mathscr{M}$ reconnaissant L et s'arrêtant toujours : réc.

Machines de Turing — décision

$w \in L?$ donnée : w, question : caract. de L

Problème de décision :

- Décidable si $\exists \mathscr{M}$ t.q. arrêt en temps fini $\forall w$ et sur acceptant ssi $w \in L$
- Semi-décidable si $\exists \mathscr{M}$ t.q. arrêt en temps fini et acceptant si $w \in L$
  non acceptant ou pas d'arrêt si $w \notin L$
- Co-semi-décidable si décision pour $\overline{L}$ semi-décidable

$\in L?$ décidable $\rightsquigarrow$ L récursif (réc.)
$\in L?$ semi-décidable $\rightsquigarrow$ L récursivement énumérable (r.é.)

1. $\exists ?$ programme (une MT) pour répondre à : $w, L \mapsto w \in L?$
2. $\exists ? L$ récursif ? $\exists ? L$ r.é. non réc. ? $\exists ? L$ non r.é. ?

Machines de Turing — fonctions

Reconnaître, ok... mais avec quelque-chose sur la bande ?

Pour $V_1 \subset V$ et $V_2 \subset V$, sans $\mathtt{B}$, fonction $f : V_1^\star \to V_2^\star$ calculée par $\mathscr{M}$:
pour $w \in V_1^\star$

- Arrêt : $(\varepsilon, q_0, w) \to^\star_{\mathscr{M}} (w_1, q, u\alpha v) \to_{\mathscr{M}} \perp$, $u \in V_2^\star$, $\alpha \notin V_2$ $\quad f(w) = u$
- Pas d'arrêt sur w : $f(w)$ non défini

Passage à n arguments ? $f(w_1, \dots, w_n) \rightsquigarrow (\varepsilon, q_0, w_1\mathtt{B}w_2\mathtt{B}\dots\mathtt{B}w_n) \to \cdots$

f Turing calculable si $\mathscr{M}$ calcule f

Machines de Turing — fonctions

Exemple : multiplication par 2 dans $\mathbb{N}$ (modulo abus de notation)

$T =$

q_0	$x \in [0 \cdots 9]$	$x \rhd q_0$
	$\mathtt{B}$	$\mathtt{B} \lhd q_{pr}$
q_{pr}	$x \in [0 \cdots 4]$	$[\![2x]\!] \lhd q_{pr}$
	$x \in [5 \cdots 9]$	$[\![2x - 10]\!] \lhd q_r$
q_r	$x \in [0 \cdots 4]$	$[\![2x + 1]\!] \lhd q_{pr}$
	$x \in [5 \cdots 9]$	$[\![2x - 9]\!] \lhd q_r$
	$\mathtt{B}$	$1 \lhd q_\perp$

Et hop on vérifie que ça marche pour 7035...

Machines de Turing — « thèse » de Church

Procédure effective : texte fini sur alphabet fini indiquant sans ambiguïté les actions mécaniques à effectuer — sans réfléchir ni comprendre

« Thèse » de Church : les fonctions calculables sont les fonctions Turing calculables

Machines de Turing — variantes

Un seul état d'arrêt ?

- Ajouter q_f
- $\forall T(q, \alpha)$ non définie ajouter $(q, \alpha) \mapsto (\alpha, \nabla, q_f)$
- Aucune transition sur q_f

Machines de Turing — variantes

Pas de transitions sur place ?

Si $T(q, \alpha) = (\beta, \nabla, q')$:

- Ajouter r
- $(q, \alpha) \mapsto (\beta, \triangleright, r)$
- $(r, x) \mapsto (x, \triangleleft, q')$ pour tout $x \in V$

Machines de Turing — variantes

Alphabet ? limitation à $\{0, 1\}$ — idée : codage en binaire

- Pour tout $x \in V$, codage sur k bits — (k suffisant pour tout V)
  Codage de B par 0^k
- Case de $\mathscr{M}$, codage par k cases pour $\mathscr{M}'$

Par transition pour $\mathscr{M}$:

- Lecture des k cases : — (mémorisation dans état)

$$\begin{array}{lcl@{\qquad}lcl} (q,0) & \mapsto & (0,\triangleright,q_0) & (q_w,0) & \mapsto & (0,\triangleright,q_{w0}) \\ (q,1) & \mapsto & (1,\triangleright,q_1) & (q_w,1) & \mapsto & (1,\triangleright,q_{w1}) \end{array}$$

In fine : q_v où v : code sur k bits d'un $\alpha \in V$ — (2^k états...)

Machines de Turing — variantes

Alphabet ? limitation à $\{0,1\}$ — idée : codage en binaire

- Pour tout $x \in V$, codage sur k bits (k suffisant pour tout V)
  Codage de $\mathtt{B}$ par 0^k
- Case de $\mathscr{M}$, codage par k cases pour $\mathscr{M}'$

Par transition pour $\mathscr{M}$:

- Lecture des k cases : q_v où v : code sur k bits d'un $\alpha \in V$
- $T(q,\alpha)$ non défini : Retour sur k cases et arrêt
- $T(q,\alpha) = (\beta, \blacktriangle, q')$
  1. Retour sur k cases avec écriture du code de β (de droite à gauche)
  2. Déplacement de k cases selon $\blacktriangle$
  3. Passage à q'

Machines de Turing — machines simples

- Pas de transitions sur place
- Alphabet : $\{0,1\}$

$\rightsquigarrow$ Machine de Turing simple

Proposition.
Si L défini par une MT, alors L défini par MT simple

Si f Turing calculable, alors f calculable par MT simple