913

Machines de Turing. Applications

Motivations : étendre le pouvoir de calcul des automates, trouver un autre modèle de calcul.

I Machines de Turing

1) Définitions :

Description 1 : Une machine de Turing déterministe est composée :
- d'un ruban infini à droite divisé en cases.
- une tête de lecture
- un ensemble fini d'états (avec un état initial et des états acceptants)
- une fonction de transition

Formellement : Une machine de Turing déterministe est décrite par un heptuplet

$$M = (Q, \Gamma, \Sigma, q_0, S, \#, \delta)$$

états ; alphabet écriture ; alphabet de lecture ; état initial ; états acceptants ; symbole blanc $\in \Gamma$; fonction de transition

où $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow ; \rightarrow\}$ et $\Sigma \subsetneq \Gamma$, $\# \notin \Sigma$, $\# \in \Gamma$

Ex 2 : voir annexe.

Def 3 : Une configuration est l'état global de la machine à un instant donné
i.e. l'état de la machine $\in Q$
le contenu du ruban $\in \Gamma^*$
la position de la tête de lecture sur le ruban.

Ex 4 : $(g, q, d) \in \Gamma^* \times Q \times \Gamma^*$ est une configuration

Def 5 : Une étape de calcul d'une machine de Turing est une paire (C, C') de configurations notée $C \to C'$ telle que C' soit la configuration après exécution de δ sur C.

Ex 6 : $00q_1 11 \to 000q_1 1$ est une étape de calcul de l'exemple 2.

Def 7 : On appelle exécution une suite de configurations successives. Elle est dite acceptante si la première configuration est initiale et si la dernière est dans un état acceptant.

Def 8 : On dit que la machine M bloque sur w, si lors de l'exécution, une configuration qui n'est pas dans un état acceptant n'a pas de configuration suivante

2) Variantes

Def 9 : Une machine de Turing à ruban bi infini est composée d'un ruban infini à droite et à gauche

Def 10 : Une machine de Turing à k rubans dispose de k rubans chacun avec une tête de lecture indépendante et $\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{\leftarrow, \rightarrow\}^k$

Def 11 : Une machine de Turing non déterministe ne possède pas de fonction de transition mais d'une relation de transition $\subseteq Q \times \Gamma \times Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ laissant plusieurs choix possibles de configurations suivantes

Thm 12 : Toutes ces variantes de machines de Turing sont équivalentes (elles ont la même expressivité de calcul.)

II Décidabilité et indécidabilité

1) Langages R et RE

Def 13 : on dit qu'un mot $w \in \Sigma^*$ est accepté par $\mathcal{M}$ s'il existe une exécution acceptante partant du mot w.
On appelle $L(\mathcal{M})$ les mots acceptés par $\mathcal{M}$.

Def 14 : Un langage $L \subseteq \Sigma^*$ est récursivement énumérable (RE) s'il existe $\mathcal{M}$ telle que $L = L(\mathcal{M})$

Prop 15 : L'ensemble des langages récursivement énumérables RE est stable par union et intersection.

Def 16 : Un langage $L \subseteq \Sigma^*$ est décidable ($\in R$) si il existe $\mathcal{M}$ sans exécution infinie telle que $L = L(\mathcal{M})$

Prop 17 : L'ensemble des langages décidables R est stable par union, intersection et complémentaire

Def 18 : On dit que $L \subseteq \Sigma^*$ est dans co-RE si son complémentaire est dans RE.

Prop 19 : $coRE \cap RE = R$.

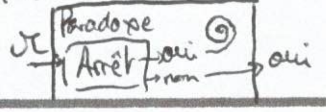
Prop 20 : Le langage $L_U = \{ \langle M, w \rangle \mid w \in L(M) \} \in RE$
Une machine $\mathcal{M}_U$ telle que $L(\mathcal{M}_U) = L_U$ est dite universelle.

2) Problèmes indécidables.

Pb 21 : ARRET
- entrée : une machine de Turing $\mathcal{M}$, w un mot
- sortie : oui si l'exécution de $\mathcal{M}$ sur w s'arrête, non sinon

est dans RE mais est indécidable

par construction d'une machine paradoxale



Def 22 : Une fonction $f : \Sigma^* \to \Gamma^*$ est dite calculable si il existe $\mathcal{M}$ telle que, pour toute entrée w, $\mathcal{M}$ s'arrête sur $f(w)$ sur le ruban.

Def 23 : Soient 2 problèmes A et B,
Une réduction de A à B est une fonction calculable $f : \Sigma_A^* \to \Sigma_B^*$ telle que $w \in L_A \Leftrightarrow f(w) \in L_B$.
(voir annexe)

Thm 24 : (Rice) Pour toute propriété P telle que $\emptyset \subsetneq P \subsetneq RE$ le problème de savoir si $L(\mathcal{M})$ (où $\mathcal{M}$ est en entrée) vérifie P est indécidable.
(par réduction de Arrêt à ce problème)

Prop 23 bis : Si A se réduit à B
- A indécidable $\Rightarrow$ B indécidable
- B décidable $\Rightarrow$ A décidable

Pb 25 : Les problèmes suivants sont indécidables.

Accept :
- entrée : $\mathcal{M}$ et w
- sortie : oui si $\mathcal{M}$ accepte w

POST :
- entrée : ensemble fini de tuiles $\left[\frac{u_i}{v_i}\right]$ où $u_i, v_i \in \Sigma^*$
- sortie : oui si il existe une concaténation de tuiles formant le même mot en haut et en bas.

VALID :
- entrée : une formule φ close en logique du 1er ordre
- sortie : oui, si φ est valide, non sinon.

III Modèles de calcul

Thèse de Church 26 : Les fonctions calculables par une procédure effective sont celles calculables par une machine de Turing.

Def 27 : calculable par machine de Turing (voir def 22)

1) Fonctions μ-récursives

Def 28 : Les fonctions primitives récursives sont le plus petit ensemble contenant la fonction zéro, la fonction successeur et les projections π_i^n qui est clos par composition et par récursion primitive contenu dans $\{\mathbb{N}^k \to \mathbb{N}, k \geq 0\}$

Ex 29 : l'addition est primitive récursive :

$$\text{plus}(n,k) = \begin{cases} n & \text{si } k=0 \\ \text{succ}(\text{plus}(n,k-1)) & \text{si } k>0 \end{cases}$$

Def 30 : On ajoute la minimisation non bornée

$$\left(\mu i(g(\vec{n},i)) = \begin{cases} \text{le plus petit } i \text{ tel que } g(\vec{n},i)=1 \\ 0 \text{ si un tel } i \text{ n'existe pas} \end{cases} \right)$$

aux fonctions primitives récursives pour ainsi obtenir les fonctions μ-récursives

Thm 31 : Toute fonction μ-récursive est calculable par une machine de Turing.

Thm 32 : Toute fonction (totale) calculable par une machine de Turing est μ-récursive **DEV 1**

2) λ-calcul

Def 33 : L'ensemble $\mathcal{L}$ des termes de λ-calcul est le plus petit ensemble tel que :

- les variables x, y, z ... sont des termes
- si u et v sont des termes, (uv) est un terme
- si x est une variable et t un terme, $\lambda x.t$ est un terme.

Thm 34 : Une fonction est calculable par une machine de Turing si et seulement si elle est représentable par un terme de λ-calcul.

IV Classes de complexité

Def 35 : La classe de problèmes P regroupe tous les problèmes décidés par une machine de Turing déterministe en temps polynomial.

Ex 36 : L'accessibilité dans un graphe est dans P

Def 37 : La classe de problèmes NP regroupe tous les problèmes décidés par une machine de Turing non déterministe en temps polynomial.

Prop 38 : $P \subseteq NP$

Def 39 : La réduction de A à B dans ce cadre nécessite une fonction $f : \Sigma_A^* \to \Sigma_B^*$ calculable en temps polynomial.

Def 40 : Un problème est NP-dur si tout problème NP peut se réduire à lui

Def 41 : Un problème est dit NP-complet s'il est dans NP et est NP-dur.

Thm 42 : Le problème SAT est NP complet **DEV 2**

SAT :
- entrée : une formule φ du calcul propositionnel
- sortie : oui si φ est satisfiable, non sinon

Pb 43 : Les problèmes suivants sont NP-complets

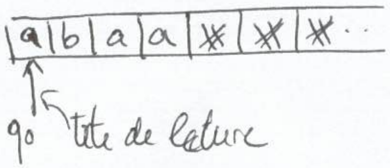
CNF SAT :
- entrée : φ en forme normale conjonctive
- sortie : oui si φ est satisfiable, non sinon

3 COL :
- entrée : un graphe $G=(S,A)$ non orienté
- sortie : oui si G est 3-coloriable, non sinon

VERTEX COVER :
- entrée : un graphe $G=(S,A)$ non orienté, un entier k
- sortie : oui si il existe $S' \subseteq S$, $|S'| \leq k$ et $\forall (u,v) \in A$, $u \in S'$ ou $v \in S'$

Description 1 :

$\Sigma = \{a, b\}$

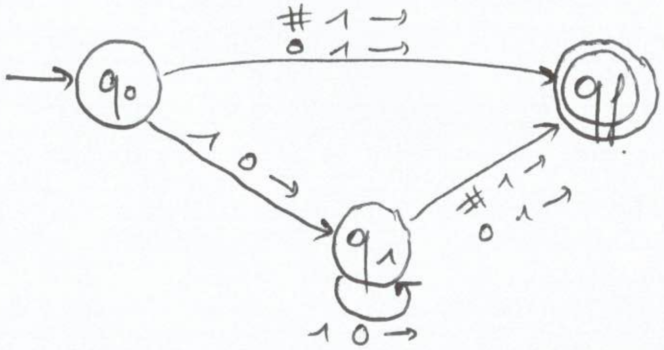


Exemple 2 : successeur pour un nombre écrit en binaire poids faible à gauche

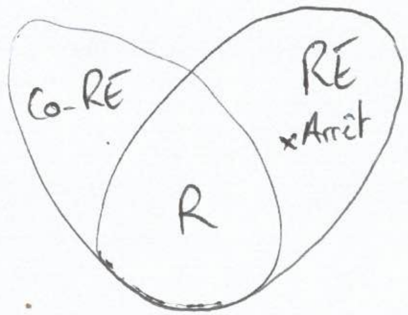
$Q = \{q_0, q_1, q_f\}$ $\quad A = \{q_f\}$

$\Sigma = \{0, 1\}$ $\quad \Gamma = \{0, 1, \#\}$

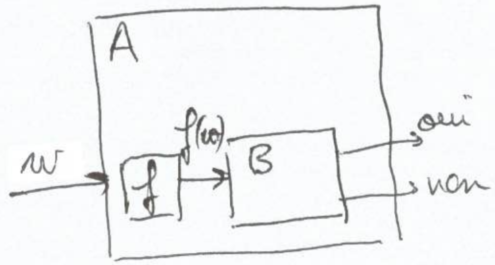
$$\delta = \{ \begin{array}{l} (q_0, 0) \to (q_f, 1, \rightarrow) \\ (q_0, 1) \to (q_1, 0, \rightarrow) \\ (q_0, \#) \to (q_f, 1, \rightarrow) \\ (q_1, 0) \to (q_f, 1, \rightarrow) \\ (q_1, 1) \to (q_1, 0, \rightarrow) \\ (q_1, \#) \to (q_f, 1, \rightarrow) \end{array} \}$$



Prop 19 :



Def 23 :



Partie IV : si $NP \neq P$