ROUHLING Damien

913 : Machines de Turing. Applications.

Réf : Sipser
Hopcroft, Ullman
Delhommé : Mathématiques de l'informatique
Wolper : Introduction à la calculabilité

I Formalisation

1 - Définitions et encodage

Déf 1 : Une machine de Turing est un 7-uplet $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ où :
- Q est l'ensemble fini des états
- Γ est l'alphabet de ruban (fini)
- $\Sigma \subseteq \Gamma$ est l'alphabet d'entrée
- $B \in \Gamma \setminus \Sigma$ est un symbole appelé blanc
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ est la fonction de transition
- $q_0 \in Q$ est l'état initial
- $F \subseteq Q$ est l'ensemble des états finaux

Calcul : On représente M par un ruban infini associé à une tête de calcul et un état : si M est dans l'état q avec $\boxed{\cdots|x|\alpha|y|\cdots}$ et si $\delta(q, \alpha) = (q', \beta, L)$, alors M passe dans l'état q' et $\boxed{\cdots|x|\beta|y|\cdots}$ (la tête se décale à droite si $\delta(q, \alpha) = (q', \beta, R)$). On suppose qu'au départ la tête est placée sur le premier symbole de l'entrée (premier différent de B). La machine s'arrête en entrant dans un état final.

Prop 1 : On peut encoder les machines de Turing dans les entiers. On note $\langle M \rangle$ le code de la machine M.

Thm 1 : Il existe une machine de Turing "universelle" M_U qui sur l'entrée $(\langle M \rangle, w)$ simule M sur l'entrée w et retourne son résultat si elle termine.

Rq 1 : Avec cette définition, M calcule une fonction partielle $\Sigma^* \to \Gamma^*$ où la sortie est le contenu du ruban. On peut aussi s'en servir pour reconnaître des langages.

Déf 2 : Soit $w \in \Sigma^*$. M accepte le mot w ssi sur l'entrée w, M termine avec pour sortie l'entier 1.
$L \subseteq \Sigma^*$ est accepté par M ssi $L = \{w \in \Sigma^*, M \text{ accepte } w\}$. On dit alors que L est récursivement énumérable ou semi-décidable.
Si de plus M termine sur toutes les entrées on dit que M décide L et L est décidable.

2 - Modèles équivalents

Déf 3 :
- machine avec ruban fini d'un côté : on suppose que si la tête est sur la première case du ruban, la transition vers la gauche est interdite
- machine à plusieurs rubans : ici, $\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R\}^k$ et il y a une tête par ruban
- machine à plusieurs têtes : même type pour δ qu'avec la machine à plusieurs rubans mais il y a k têtes sur un ruban

Thm 2 : L est accepté par une machine de l'un de ces types ssi L est accepté par une machine de Turing

Déf 4 : machine de Turing non déterministe : ici, $\delta \subseteq (Q \times \Gamma) \times (Q \times \Gamma \times \{L, R\})$

Calcul : dans l'état q avec $\boxed{\cdots|x|\alpha|y|\cdots}$, la machine tire l'un des (q', β, d) tels que $((q, \alpha), (q', \beta, d)) \in \delta$ et effectue la transition déterministe correspondante.

Rq 2 : On peut supposer qu'il y a toujours r tels triplets et alors considérer $\delta : Q \times \Gamma \times [\![1, r]\!] \to Q \times \Gamma \times \{L, R\}$

Déf 5 : w est accepté par une machine de Turing non déterministe M ssi il existe un calcul terminant de M acceptant w.

Thm 3 : L est accepté par une machine de Turing non déterministe ssi L est accepté par une machine de Turing.

Déf 6 : Un énumérateur fonctionne comme une machine de Turing sans entrée, non terminante, et qui écrit sur son ruban une suite de mots de Σ^* séparés par un marqueur $\# \in \Gamma \setminus (\Sigma \cup \{B\})$

Thm 4 : L est récursivement énumérable ssi L est énuméré par un énumérateur

II Calculabilité, décidabilité

1. Calculabilité

Déf 7 : Une fonction $f : \Sigma^* \to \Sigma^*$ est Turing-calculable ssi il existe une machine de Turing calculant f.

Rq 3 : En démultipliant les rubans, on peut calculer des fonctions $f : \Sigma^{*p} \to \Sigma^{*q}$

Ex 1 : La concaténation $\Sigma^{*2} \to \Sigma^*$ est Turing-calculable.

Thm 5 : Une fonction $\mathbb{N} \to \mathbb{N}$ est Turing-calculable ssi elle est récursive.

Prop 2 : La composée de deux fonctions Turing-calculables est Turing-calculable

<table>
<tr><th>Développement 1</th></tr>
<tr><td>Thm 6 : La fonction du castor affairé n'est pas Turing-calculable

Cor 1. Le problème de l'arrêt (ie $\{\langle M \rangle, M$ machine de Turing dont le calcul sur l'entrée vide termine$\}$) est indécidable</td></tr>
</table>

Déf 8 : Un réel x est Turing-calculable en base B si la fonction qui à n associe le nè chiffre de la décomposition de x en base B est Turing-calculable.

Prop 3 : L'ensemble des réels Turing-calculables forme un sous-corps dénombrable de $\mathbb{R}$, contenant les nombres algébriques, e, π, $\log(2)$...

2. Décidabilité

Rq 4 : Nous avons déjà vu la définition de la décidabilité et un exemple de langage indécidable. Ici, nous voyons des techniques pour prouver la décidabilité, l'indécidabilité.

Prop 4 : L est décidable ssi L et L^c sont récursivement énumérables.

Prop 5 : Le problème de l'arrêt est semi-décidable

Déf 9 : L se réduit à L', noté $L \leq_T L'$, ssi sous l'hypothèse de la décidabilité de L', L est décidable.

Thm 7 (Rice) : Si $\emptyset \subsetneq P \subsetneq \{\langle M \rangle, M$ machine de Turing$\}$ alors P est indécidable.

Thm 8 : Si T est une machine de Turing calculant $t : \Sigma^{*2} \to \Sigma^*$, il y a une machine de Turing R calculant $r : \Sigma^* \to \Sigma^*$ telle que ~~pour~~ tout $w \in \Sigma^*$ $r(w) = t(\langle R \rangle, w)$

Ex 2 : M est minimale s'il n'y a pas de machine M' équivalente à M telle que $|\langle M' \rangle| < |\langle M \rangle|$
$\{\langle M \rangle, M$ minimale$\}$ n'est pas semi-décidable.

III Complexité

1- Généralités

Déf 10 : • Si pour toute entrée de taille n la machine de Turing M effectue au plus $T(n)$ pas de calculs avant de s'arrêter, on dit que M a pour complexité temporelle $T(n)$

• Si pour toute entrée de taille n la machine de Turing M scanne au plus $S(n)$ cases du ruban, on dit que M a pour complexité spatiale $S(n)$

Rq 5 : • Cette définition se généralise aux langages acceptés

• Cette définition se généralise aux machines de Turing non déterministes : il suffit que pour chaque mot une séquence de choix satisfasse la définition.

Déf 11 : • On note $DTIME(T(n))$ l'ensemble des langages de complexité temporelle $T(n)$ et $NTIME(T(n))$ celui des langages de complexité temporelle non déterministe $T(n)$.

• On définit similairement $DSPACE(S(n))$ et $NSPACE(S(n))$

Rq 6 : Ces classes de langages dépendent du modèle de machine de Turing considéré.

Prop 6 : Si $T(n) \geq n$, toute machine de Turing à plusieurs rubans de complexité temporelle $T(n)$ admet une machine de Turing à un ruban équivalente de complexité $O(T(n)^2)$

Prop 7 : Si $T(n) \geq n$, $NTIME(T(n)) \subseteq DTIME(2^{O(T(n))})$

Déf 12 : $S(n) \geq \log(n)$ est constructible en espace s'il existe une machine de Turing de complexité spatiale $S(n)$ qui utilise effectivement $S(n)$ cases pour au moins une entrée de taille n.

Thm 9 (Savitch) : Si $S(n)$ est constructible en espace, alors $NSPACE(S(n)) \subseteq DSPACE(S(n)^2)$

2- P et NP

Déf 13 : $P = \bigcup_{k \in \mathbb{N}} DTIME(n^k)$ $NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k)$

Déf 14 : Un vérificateur pour L est une machine de Turing M telle que pour tout $w \in \Sigma^*$, $w \in L$ ssi il existe $c \in \Sigma^*$ avec $M(w,c) = 1$

Thm 10 : $L \in NP$ ssi L admet un vérificateur de complexité temporelle polynomiale avec des certificats de taille polynomiale

Ex 3 : $\{(G,s,t)$, G graphe orienté dans lequel il y a un chemin de s à $t\}$ appartient à P

$\{(G,k)$, G graphe non orienté contenant une k-clique$\} \in NP$.

Prop 8 : $P \subseteq NP$

Rq 7 : L'inclusion réciproque est un problème ... difficile.

Déf 15 : L est réductible à L' en temps polynomial ssi il existe $f : \Sigma^* \to \Sigma^*$ calculable en temps polynomial telle que pour tout $w \in \Sigma^*$, $w \in L$ ssi $f(w) \in L'$. On note $L \leq_p L'$

Prop 9 : Si $L \leq_p L'$ et $L' \in P$ alors $L \in P$.

Déf 16 : • L est NP-difficile ssi pour tout $L' \in NP$, $L' \leq_p L$

• L est NP-complet ssi $L \in NP$ et L est NP-difficile

Développement 2
Thm 11 (Cook) : $SAT = \{F$, F formule du calcul propositionnel en forme normale conjonctive satisfiable$\}$ est NP-complet.