leçon 915 : Classe de complexité, exemples

"C'est mal de ne pas mettre quelque chose qui est dans le Carton, c'est mal de mettre quelque chose qui n'est pas dans le Carton." David Cachera.

Cadre : On se place dans le cadre de problèmes décidables.

I Outils : les machines de Turing

Définition 1 : Une machine de Turing est un septuplet $(Q, \Sigma, \Gamma, E, q_0, F, \#)$ où

- Q est l'ensemble des états
- Σ est l'alphabet d'entrée
- Γ est l'alphabet de bande
- E est l'ensemble des transitions
- q_0 est l'état initial
- F est l'ensemble des états finaux
- $\#$ est le symbole blanc.

Définition 2 : Soit M une machine de Turing. On suppose que chaque calcul termine (cas décidable).
Soit w une entrée et $C_w = \{c ;$ calcul possible par M à partir de $w\}$. On note $c : q_0 w \to C_1 \to C_2 \to \dots \to C_m$.
On définit le temps et l'espace de calcul t_M et s_M par ; $\forall c \in C_w$; $t_M(c) = m$; $s_M(c) = \max_{0 \leq i \leq m} |C_i|$

On définit les complexités pour une entrée w par

$t_M(w) = \max_{c \in C_w} t_M(c)$ (temporelle) $\quad s_M(w) = \max_{c \in C_w} s_M(c)$ (spatiale).

On définit les complexités d'une machine M par

$t_M(n) = \max_{|w|=n} t_M(w)$ (temporelle) $\quad s_M(n) = \max_{|w|=n} s_M(w)$ (spatiale)

Remarque 3 : Pour une machine de Turing à plusieurs bandes, on ne prend pas en compte la taille du ruban d'entrée dans la taille d'une configuration.

Proposition 4 : $\forall M$ machine de Turing, $\exists K \in \mathbb{R}^+$ tel que

- $s_M(n) \leq \max(t_M(n) ; n)$
- $t_M(n) \leq 2^{K s_M(n)}$

II Complexité en temps

Théorème 5 : Théorème d'accélération (temporelle)

Soit $k \in \mathbb{N}^*$, M une machine de Turing.
Si $n = o(t_M(n))$, alors $\exists M'$ une machine de Turing, équivalente à M, telle que $t_{M'}(n) \leq \frac{t_M(n)}{k}$

Proposition 6 : Toute machine de Turing à k bandes est équivalente à une machine de Turing à 1 bande telle que $t_{M'}(n) = O((t_M(n))^2)$ $(k \geq 2)$.

Proposition 7 : Toute machine de Turing M non déterministe est équivalente à une machine de Turing déterministe M' telle que $t_{M'}(n) = O(t_M(n) k^{t_M(n)}) = 2^{O(t_M(n))}$

Définition 8 : Soit $f : \mathbb{N} \to \mathbb{R}^+$. On pose

$TIME(f) = \{$ Problèmes décidés par une machine de Turing déterministe à plusieurs bandes en temps $O(f)$ $\}$

et $NTIME(f)$ l'équivalent pour des machines de Turing non déterministe.

Définition 9 : On définit les classes de complexité :

$P = \bigcup_{k \geq 0} TIME(n^k)$; $\quad NP = \bigcup_{k \geq 0} NTIME(n^k)$

$EXPTIME = \bigcup_{k \geq 0} TIME(2^{n^k})$; $\quad NEXPTIME = \bigcup_{k \geq 0} NTIME(2^{n^k})$

Remarque 10 : On choisit comme modèle de calculs les machines de Turing à plusieurs bandes. La prop. 6 assure que la définition 9 reste vraie en prenant des machines de Turing à une bande.

Exemple 11 :

Soit un graphe $G=(V;E)$; une fonction capacité qui pondère les arcs ; $k \in \mathbb{R}^{+*}$; une source et un puit. Le problème de décision, associé au problème du flot maximal, qui demande s'il existe un flot de valeur supérieure à k entre la source et le puit dans G est dans P.

Exemple 12 :

Le problème de satisfiabilité des formules sous forme normale conjonctive en logique propositionnelle est dans NP

Théorème 13 :

Un problème de décision est dans NP si et seulement si il existe pour toute instance positive un certificat de taille polynomiale en la taille de l'instance qui permet de vérifier la réponse en temps polynomiale.

Exemple 14 :

Soit le problème du chemin hamiltonien (Ham-Path), et G un graphe, instance positive de Ham Path.
La donnée d'un chemin hamiltonien de G est un certificat.
Ham Path est dans NP.

Proposition 15 :

On a la suite d'inclusion suivante :

$$P \subseteq NP \subseteq EXPTIME \subseteq NEXPTIME.$$

Definition 16 : Soit A et B deux problèmes de décision codés par les langages L_A et L_B sur les alphabet Σ_A et Σ_B. Une réduction polynomiale est une fonction $f : \Sigma_A^* \to \Sigma_B^*$; calculable en temps polynomiale telle que $w \in L_A \Leftrightarrow f(w) \in L_B$. On note $A \leq B$ (A se réduit à B).

Définition 17 : Soit $C = P$ ou $C = NP$
Un problème Q est dit C-dur si tout problème de C peut se réduire au problème Q avec une réduction polynomiale. Si $Q \in C$, il est C-complet.

Théorème 18 : (Cook)

Les problèmes SAT et 3SAT sont NP-Complet.

Application 19 : Le problème Ham Path est NP-Complet — Dev 1 PE

Application 20 : Le problème de 3-coloration dans un graphe est NP-complet — Dev 1 bis Rémi

III Complexité en espace

Théorème 21 (Savitch) : DEV 2

Soit s une fonction de $\mathbb{N}$ dans $\mathbb{R}^+$ telle que $s(n) \geq n$ pour tout n assez grand. Toute machine de Turing non déterministe qui fonctionne en espace $s(n)$ est équivalente à une machine déterministe qui fonctionne en espace $O(s^2(n))$

Définition 22 : Soit $f : \mathbb{N} \to \mathbb{R}^+$. On définit :

SPACE(f) = {Problèmes décidés par une machine de Turing déterministe à plusieurs rubans en espace $O(f)$}

NSPACE(f) = l'équivalent pour des machines non déterministes.

Définition 23 : On définit les classes suivantes :

$PSPACE = \bigcup_{k \geq 0} SPACE(n^k)$ $\quad$ $NPSPACE = \bigcup_{k \geq 0} NSPACE(n^k)$

$ExpSPACE = \bigcup_{k \geq 0} SPACE(2^{n^k})$ $\quad$ $NExpSPACE = \bigcup_{k \geq 0} NSPACE(2^{n^k})$

Remarque 24 : Le théorème de Savitch montre que PSPACE = NPSPACE et ExpSPACE = NExpSPACE.

Exemple 25 : Le problème de l'universalité est de savoir si un automate fini A d'alphabet Σ reconnait Σ^*.
Ce problème est dans PSPACE.

Remarque 26 : On peut définir les notions de PSPACE-dureté et PSPACE-complétude de manière analogue à la classe P.

Définition 27 : Prenons la convention que l'on travaille avec des machines de Turing à plusieurs ruban dont la taille des configurations ignore les rubans d'entré et de sortie.
On définit alors

$L = SPACE(\log)$ $\quad$ $NL = NSPACE(\log)$

Définition 28 : Soient A et B des problèmes codés respectivement par L_A et L_B sur les alphabets Σ_A et Σ_B. Une réduction logarithmique de A à B est une fonction $f : \Sigma_A^* \to \Sigma_B^*$ calculable en espace logarithmique par une machine de Turing déterministe telle que : $w \in L_A \Leftrightarrow f(w) \in L_B$. On note $A \leq_{\log} B$.

Proposition 29 : La composée de deux fonctions calculables en espace logarithmique est encore calculable en espace logarithmique.

Corollaire 30 : Si $A \leq_{\log} B$ et $B \in L$ alors $A \in L$.

Définition 31 : Un problème A est NL-complet si $A \in NL$ et pour tout $B \in NL$, $B \leq_{\log} A$.

Exemple 32 : Le problème PATH est de savoir s'il existe un chemin dans un graphe entre deux sommets donnés.
Ce problème est NL-complet.

IV Hiérarchie des classes.

Définition 33 : Une fonction $f : \mathbb{N} \to \mathbb{N}$ est dite constructible en temps (respectivement en espace) s'il existe une machine de Turing qui calcule le mot $1^{f(n)}$ pour une entrée 1^n en temps $O(f(n))$ (respectivement en espace $O(f(n))$).

Théorème 34 : Soient $f : \mathbb{N} \to \mathbb{N}$ et $g : \mathbb{N} \to \mathbb{N}$ deux fonctions telles que $g = o(f)$. Si f est constructible en temps (respectivement en espace), alors $TIME(g) \subsetneq TIME(f)$ $(SPACE(g) \subsetneq SPACE(f))$.

Corollaire 35 : $P \subsetneq EXPTIME$ et $PSPACE \subsetneq ExpSPACE$

Théorème 36 : $L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq ExpTIME$

$ExpTIME \subseteq NExpTIME \subseteq ExpSPACE$

Référence :

Carton