Leçon 913 : Machines de Turing. Applications

BOLLE Quentin

Objectif : - définir un modèle de calcul
- permettre des définitions souples

I) Premières notions

Def (Langage d'un problème) : On utilise un codage d'un ensemble E dans l'ensemble des mots Σ^*. Pour $x \in E$, on note $\langle x \rangle \in \Sigma^*$.
Le langage associé à un problème P sur E est $L_P = \{\langle x \rangle \mid x \in P\}$

Ex : • Pour $E = \mathbb{N}$, $n \mapsto \langle n \rangle$ peut être l'écriture en base 2
• Pour les graphes, on code les sommets et les arêtes.

Def (Machine de Turing) : Une machine de Turing est un 7-uplet $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{rejet})$ où Q, Σ, Γ sont des ensembles finis et
1. Q est l'ensemble des états
2. Σ est l'alphabet d'entrée ne contenant pas $\sqcup$.
3. Γ est l'alphabet de bande, avec $\sqcup \in \Gamma$ et $\Sigma \subset \Gamma$
4. $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ est la fonction de transition
5. $q_0 \in Q$ est l'état initial
6. $q_{accept} \in Q$ est l'état acceptant
7. $q_{rejet} \in Q$ est l'état de rejet avec $q_{accept} \neq q_{rejet}$

Fonctionnement

Début : $\boxed{q_0}$
↓ tête de lecture/écriture
| a_1 | a_2 | ... | a_n | $\sqcup$ | $\sqcup$ | $\sqcup$ | $\sqcup$ | ... avec $a_i \in \Sigma$

Déroulement : Dans l'état q_i, la tête de lecture lit le symbole e. Si $\delta(q_i, e) = (q_j, d, \rightarrow)$ (resp. $(q_j, d, \leftarrow)$), on passe dans l'état q_j et la tête d'écriture écrit d et se déplace à droite (resp. gauche).

Fin : Dans l'état q_{accept}, le calcul est accepté
Dans l'état q_{rejet}, le calcul est rejeté.

Def (Langage accepté) : Soit M une machine de Turing. Un mot $w \in \Sigma^*$ est accepté par M si le calcul est accepté à partir de la configuration initiale w.
On note $L_M = \{w \in \Sigma^* \mid w \text{ est accepté par } M\}$ le langage accepté par M.

Def (graphe des configurations) : Soit M une machine de Turing.
Le graphe des configurations de M est le graphe dont l'ensemble des sommets est l'ensemble de toutes les configurations de M et dont les arêtes sont les paires (C, C') de configurations telles que $C \xrightarrow{\delta} C'$.
Un chemin dans ce graphe est donc un calcul de M.

II) Variantes - Stabilité du modèle

1) Bandes bi-infinies

Def : Une machine à bande bi-infini est formellement identique à une machine de Turing, mais la bande est indicée par tous les entiers de $\mathbb{Z}$.

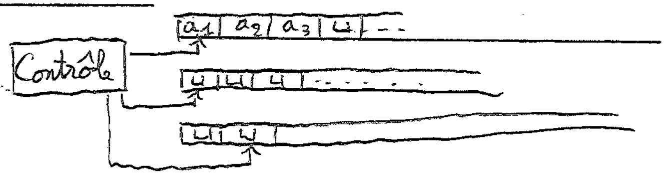
$\boxed{q_0}$
↓
... | $\sqcup$ | $\sqcup$ | $\sqcup$ | a_1 | a_2 | a_3 | a_4 | $\sqcup$ | $\sqcup$ | ...

Prop : Toute machine de Turing à bande bi-infini est équivalente à une machine de Turing (cad. qui accepte le même langage). Inversement, toute machine de Turing est équivalente à une machine de Turing à bande bi-infinie.

2) Machine à plusieurs bandes

Déf. Une machine à k bandes est une machine disposant de k bandes, chacune lue par une tête de lecture indépendante. Une transition est alors un élément de $Q \times \Gamma^k \Rightarrow Q \times \Gamma^k \times \{\leftarrow, \rightarrow, \downarrow\}^k$ où $\downarrow$ permet à une tête de rester immobile.

Fonctionnement:



Prop. Toute machine de Turing à plusieurs bandes est équivalente à une machine de Turing (à une seule bande).

3) Machine non-déterministe

Déf. Une machine de Turing non-déterministe est une machine de Turing où la fonction de transition est de la forme $Q \times \Gamma \mapsto \mathcal{P}(Q \times \Gamma \times \{\leftarrow, \rightarrow\})$.

Un calcul d'une machine non-déterministe est un arbre où chaque branche correspond à un calcul effectif. Un calcul non-déterministe est accepté si au moins l'un des calculs effectifs est accepté.

Prop. Une machine de Turing non-déterministe est équivalente à une machine de Turing (déterministe).

III) Décidabilité et Calculabilité

Déf. (Langage récursivement énumérable): Un langage $L \subseteq \Sigma^*$ est récursivement énumérable s'il existe une machine de Turing M telle que $L = L_M$.

Déf. (Langage décidable) Un langage $L \subseteq \Sigma^*$ est décidable s'il existe une machine de Turing M qui ne fait aucun calcul infini et telle que $L = L_M$.

Prop. Soient L et L' deux langages récursivement énumérables (resp. décidables). Alors:
i) $L \cup L'$ est récursivement énumérable (resp. décidable)
ii) $L \cap L'$ est récursivement énumérable (resp. décidable)
iii) si L est décidable, alors $\Sigma^* \setminus L$ est décidable.

Prop. Soit $L \subset \Sigma^*$.
L est décidable si et seulement si L et $\Sigma^* \setminus L$ sont récursivement énumérables.

Déf. On note $L_\in = \{\langle M, w \rangle \mid w \in L(M)\}$ le langage d'acceptation.

Prop-Déf (Machine universelle):
$L_\in$ est récursivement énumérable : on appelle machine universelle toute machine M_U telle que $L_\in = L_{M_U}$.

Prop. $L_\in$ n'est pas décidable.

Cor. $\Sigma^* \setminus L_\in$ n'est pas récursivement énumérable.

Déf. Une fonction $f : \Sigma^* \to \Gamma^*$ est calculable s'il existe une machine de Turing qui pour toute entrée w s'arrête avec $f(w)$ sur la bande.

Def (Réduction) : Soit L_A, L_B deux langages sur respectivement Σ_A^* et Σ_B^*. Une réduction de L_A à L_B est une fonction calculable $f : \Sigma_A^* \to \Sigma_B^*$ telle que $w \in L_A \iff f(w) \in L_B$. On note $L_A \leq_m L_B$.

Prop : Si $L_A \leq_m L_B$ et L_B décidable, alors L_A est décidable.

Coro : Si $L_A \leq L_B$ et L_A indécidable, alors L_B est indécidable.

DEV1 ←

Prop : Le problème de l'arrêt $L_a = \{\langle M, w\rangle \mid M$ s'arrête sur $w\}$ est indécidable

Th (Rice) Si P est une propriété non triviale sur les langages, alors $L_P = \{\langle M\rangle \mid L_M$ vérifie $P\}$ est indécidable.

IV) Complexité

Def (Temps et espace) : Soit une machine de Turing M (à priori non déterministe) et soit $\gamma = q_0 w \to C_1 \to \dots \to C_m$ un calcul sur l'entrée w où C_m est une configuration acceptante :
- le temps $t_M(\gamma)$ est m.
- l'espace $s_M(\gamma)$ est $\max_{i \in [1,m]}$ taille(C_i).

On définit ainsi la complexité pour une entrée w :
- $t_M(w) = \max_\gamma t_M(\gamma)$ et $s_M(w) = \max_\gamma s_M(\gamma)$ (où $\gamma = q_0 w \dots$)

Enfin, on définit les complexités de la machine M :
- $t_M(n) = \max_{w : |w| = n} t_M(w)$ et $s_M(n) = \max_{w : |w| = n} s_M(n)$

Rem : On a $s_M(n) \geq n$ avec cette définition. Avec un ruban d'entrée, de travail et de sortie, on peut avoir $s_M(n) = o(n)$ (en ne regardant que l'espace du ruban de travail)

Def : On note, pour une fonction $f : \mathbb{N} \to \mathbb{R}^+$, les classes
- TIME$(f(n))$ l'ensemble des langages décidés par une machine M déterministe avec $t_M(n) = O(f(n))$
- NTIME$(f(n))$ l'ensemble des langages décidés par une machine de M non déterministe avec $t_M(n) = O(f(n))$

Rem : En remplaçant t_M par s_M, on définit de même SPACE$(f(n))$ et NSPACE$(f(n))$.

Def : On note $P = \bigcup_{k \in \mathbb{N}} \text{TIME}(n \mapsto n^k)$, $NP = \bigcup_{k \in \mathbb{N}} \text{TIME}(n \mapsto n^k)$

$$EXP = \bigcup_{k \in \mathbb{N}} \text{TIME}(n \mapsto 2^{n^k}).$$

Prop : On a $P \subseteq NP \subseteq EXP$ et $P \neq EXP$.

Conjecture : $P \overset{?}{=} NP$ est indécidable

Def : On note $PSPACE = \bigcup_{k \in \mathbb{N}} \text{SPACE}(n \mapsto n^k)$ et $NPSPACE = \bigcup_{k \in \mathbb{N}} \text{NSPACE}(n \mapsto n^k)$

Th (Savitch) : Pour $f : \mathbb{N} \to \mathbb{R}^+$ avec $f(n) \geq n$, on a

$$NSPACE(f(n)) \subseteq SPACE(f(n)^2)$$

← DEV2

Coro : $PSPACE = NPSPACE$