Calculabilité
Cours 2 : Machines de Turing

Un autre type de modèle de calcul

Les fonctions récursives et les fonctions $\lambda-$représentables définissent des modèles de calculs dans $\bigcup_{k\in\mathbb{N}} \mathbb{N}^k \rightarrow \mathbb{N}$

Nous nous intéressons maintenant à des modèles de calcul travaillant sur des chaînes de caractères.

- mots définis sur un alphabet Σ

Nous nous restreignons (principalement) aux problèmes binaires :

- la procédure de calcul doit permettre de décider si un mot appartient ou non à une certaine classe de Σ^*
  Exemple de classe : l'ensemble des mots représentant les programmes C qui s'arrêtent toujours.

Machine de Turing : Définition

- Mémoire infinie sous forme de ruban divisé en cases ; alphabet de ruban.
- Tête de lecture.
- Ensemble fini d'états. Etats accepteurs.
- Fonction de transition qui pour chaque état de la machine et symbole se trouvant sous la tête de lecture précise
  - l'état suivant,
  - un caractère qui sera écrit sur le ruban,
  - un sens de déplacement de la tête de lecture.

Exécution

- Initialement, mot d'entrée au début du ruban, symbole blanc partout ailleurs, tête de lecture sur la première case.
- A chaque étape, la machine
  - lit le symbole se trouvant sous sa tête de lecture,
  - remplace ce symbole suivant la fonction de transition,
  - déplace sa tête de lecture d'une case vers la gauche ou vers la droite suivant la fonction de transition,
  - change d'état comme indiqué par la fonction de transition.
- Mot accepté si état accepteur atteint.

Formalisation

Septuplet $M = (Q, \Gamma, \Sigma, \delta, s, B, F)$, où :

- Q est un ensemble fini d'états,
- Γ est l'alphabet de ruban,
- $\Sigma \subseteq \Gamma$ est l'alphabet d'entrée,
- $s \in Q$ est l'état initial,
- $F \subseteq Q$ est l'ensemble des états accepteurs,
- $B \in \Gamma - \Sigma$ est le *symbole blanc* (#),
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ est la fonction (partielle) de transition.

Configuration

Information nécessaire :

1. l'état,
2. le contenu du ruban,
3. la position de la tête de lecture.

Remarque : à tout moment dans l'exécution, il existe une position sur le ruban à partir de laquelle il n'y a que des #.

Représentation : triplet (q, α_1, α_2) contenant

1. l'état de la machine ($q \in Q$),
2. le mot sur le ruban avant la tête de lecture $(\alpha_1 \in \Gamma^*)$,
3. le mot sur le ruban après la tête $(\alpha_2 \in \{\varepsilon\} \cup \Gamma^*(\Gamma - \{\#\}))$.

Dérivation d'une configuration à une configuration successeur

Configuration (q, α_1, α_2) écrite sous la forme $(q, \alpha_1, b\alpha_2')$ avec $b = \#$ si $\alpha_2 = \varepsilon$.

- Si $\delta(q, b) = (q', b', R)$ nous avons

$$(q, \alpha_1, b\alpha_2') \vdash_M (q', \alpha_1 b', \alpha_2')$$

- Si $\delta(q, b) = (q', b', L)$ et si $\alpha_1 \neq \varepsilon$ et est donc de la forme $\alpha_1' a$ nous avons

$$(q, \alpha_1' a, b\alpha_2') \vdash_M (q', \alpha_1', ab'\alpha_2')$$

- Sinon ($\delta(q, b)$ n'est pas définie), la machine est bloquée.

Langage accepté

Une configuration C' est dérivable en plusieurs étapes de la configuration C par la machine M ($C \vdash_M^* C'$) s'il existe $k \geqslant 0$ et des configurations intermédiaires $C_0, C_1, C_2, \ldots, C_k$ telles que

- $C = C_0$,
- $C' = C_k$,
- $C_i \vdash_M C_{i+1}$ pour $0 \leqslant i < k$.

Définition (Langage accepté)

$M = (Q, \Gamma, \Sigma, \delta, s, B, F)$

$$\mathcal{L}(M) = \{w \in \Sigma^* \mid \exists q \in F,\ \exists \alpha_1, \alpha_2 \in \Gamma^*,\ (s, \varepsilon, w) \vdash_M^* (q, \alpha_1, \alpha_2)\}$$

Exemple : $\mathcal{L}(M) =?$

Machine de Turing $M = (Q, \Gamma, \Sigma, \delta, s, \#, F)$ avec

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$,
- $\Gamma = \{a, b, X, Y, \#\}$,
- $\Sigma = \{a, b\}$,
- $s = q_0, F = \{q_4\}$,
- δ donné par

	a	b	X	Y	$\#$
q_0	(q_1, X, R)	—	—	(q_3, Y, R)	$(q_4, \#, R)$
q_1	(q_1, a, R)	(q_2, Y, L)	—	(q_1, Y, R)	—
q_2	(q_2, a, L)	—	(q_0, X, R)	(q_2, Y, L)	—
q_3	—	—	—	(q_3, Y, R)	$(q_4, \#, R)$
q_4	—	—	—	—	—

Langage accepté / langage décidé

Machine de Turing = procédure effective ? Pas toujours.

Considérons la suite des transitions obtenues à partir de (s, ε, w)

1. La suite des configurations contient un état accepteur : w est accepté.
2. La suite de configurations se termine parce que soit
   - la fonction de transition n'est pas définie,
   - soit elle impose un déplacement à gauche au début du ruban.
3. La suite de configurations ne passe jamais par un état final et est infinie.

Dans les cas 1 et 2 on a une procédure effective, dans le cas 3 pas.

Langage décidé

On appelle *exécution* d'une machine de Turing sur un mot w la suite de configurations

$$(s, \varepsilon, w) \vdash_M C_1 \vdash_M C_2 \vdash_M \cdots \vdash_M C_k \vdash_M \cdots$$

maximale.

Définition (Langage décidé)

Un langage L est décidé par une machine M si

- M accepte L : $L = \mathcal{L}(M)$
- M n'a pas d'exécution infinie

Autres définitions des machines de Turing

1. Un seul état d'arrêt et fonction de transition est partout définie. Dans l'état d'arrêt, résultat sur le ruban : "accepte" (1) ou "n'accepte pas" (0).

2. Deux états d'arrêt : q_Y et q_N , et la fonction de transition est partout définie.

Thèse de Turing-Church

Les langages reconnus par une procédure effective sont ceux décidés par une machine de Turing.

Justification.

1. Si décidé par machine de Turing, alors calculable : immédiat.
2. Si calculable alors décidé par machine de Turing :
    - Extensions des machines de Turing et autres machines.
    - Autres modélisations.

Extensions des machines de Turing

- Ruban infini dans les deux sens
    - cf TD
- Ruban multiples
- Machine à mémoire à accès direct

Machines de Turing non déterministes

Relation de transition :

$$\delta : (Q \times \Gamma) \times (Q \times \Gamma \times \{L, R\})$$

L'exécution n'est plus unique.
Un mot est accepté s'il existe une exécution avec un état accepteur.

Théorème

Tout langage accepté par une machine de Turing non déterministe est aussi accepté par une machine de Turing déterministe.

Machines de Turing universelles

Machine de Turing qui simule n'importe quelle machine de Turing.

- prend en entrée
  - une machine de Turing M
  - et une donnée w pour M,
- simule l'exécution de M sur w.

Théorème

Il existe des machines universelles.

Fonctions calculables par une machine de Turing

Définition (Fonction calculables)

Une machine de Turing calcule une fonction $f : \Sigma^* \rightarrow \Sigma^*$ si, pour tout mot d'entrée w, elle s'arrête toujours dans une configuration où $f(w)$ se trouve sur le ruban.

Les fonctions calculables par une procédure effective sont les fonctions calculables par une machine de Turing

Équivalence des modèles de calcul

Théorème

$$\begin{aligned} \textit{les fonctions calculables par MT} &\equiv \textit{les fonctions } \mu\textit{-récursives} \\ &\equiv \textit{les fonctions } \lambda\textit{—représentables} \end{aligned}$$

Plan