Calculabilité & Complexité [M1]

Hugo Férée, Valia Mitsou

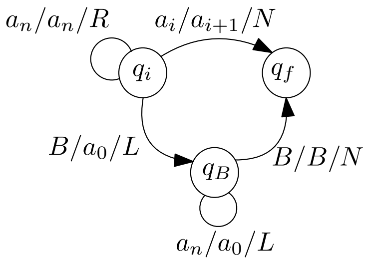
Feuille de TD n°1
Machines de Turing et Décidabilité

Exercice 1 *Parmi mes modifications suivantes du modèle des machines de Turing, dites lesquelles sont équivalentes en terme de calculabilité. Choisissez l'une d'elles et démontrez-le informellement.*

1. *d'interdire les transition N ;*
2. *de n'avoir qu'un ruban semi-infini (commençant par un symbole spécial) ;*
3. *de rajouter des rubans (dits « de travail ») ;*
4. *de changer d'alphabet de ruban ($\{0,1\} \bigcup \{B\}$ « suffit »)*

Exercice 2 *Construire une machine de Turing qui calcule la fonction successeur sur les mots de l'alphabet $\{a_0, \ldots, a_n\}$.*

Solution



Exercice 3 *Montrer que les machines de Turing déterministes et non-déterministes reconnaissent les mêmes langages (semi-décidables, donc). On pourra s'aider des exercices précédents.*

Exercice 4 *Montrez que votre langage de programmation préféré est Turing-complet, i.e. pour toute fonction calculable, il existe un programme qui la calcule.*

Solution Pour ça, il faut écrire un programme qui interprète les machines de Turing.

Exercice 5 *Montrer qu'un langage est semi-décidable si et seulement si il peut être énuméré par une fonction calculable*[1].

1. D'où le nom « récursivement énumérable ».

Solution **Lemme** (admis, inspiré de l'exercice 3) : pour toute machine $\mathcal{M}$, la fonction totale $\mapsto eval_{\mathcal{M}}$ est calculable ; avec $eval(w,k) = 1 \cdot \mathcal{M}(w)$ si $\mathcal{M}$ s'arrête sur w en moins de k étapes et 0 sinon.

$\Rightarrow$ Si L est reconnu par $\mathcal{M}$, on pose f la fonction qui sur u retourne le w de la $|u|-$ième paire (w,k) telle que $eval_{\mathcal{M}}(w,k) = 11$.

f énumère bien L car :

$f(\Sigma^*) = \{w | \exists k, eval(w,k) = 11\}$, et $w \in L \Leftrightarrow \exists k, eval_{\mathcal{M}}(w,k) = 11$.

$\Leftarrow$ Si f est calculée par $\mathcal{M}_f$ et énumère L, alors on peut définir la machine qui sur l'entrée w énumère les paires (w',k) et accepte si $eval_{M_f}(w',k) = 1 \cdot w$.

$w \in L \Leftrightarrow \exists k, eval_{M_f}(w',k) = 1 \cdot w \Leftrightarrow \mathcal{M}(w) = 1$.

À titre d'exercice : définir en quelques lignes les fonctions correspondant à $\mathcal{M}$ et $\mathcal{M}_f$ en ocaml, en partant d'une fonction *eval*.

Exercice 6

1. *Dites si les langages* ***décidables*** *et* ***semi-décidables*** *sont respectivement clos par : union, étoile, concaténation, complémentaire, intersection*
2. *Quel est le lien entre l'ensemble des langages décidables et semi-décidables ?*
3. *Montrer que tout langage semi-décidable infini contient un langage décidable infini.*
4. *Soit L le langage contenant un unique mot binaire encodant votre future note à ce module. L est-il décidable ?*

Exercice 7 *Prouver l'équivalence suivante :*

— *L est un langage semi-décidable* $\Leftrightarrow$

— *Il existe L' décidable tel que* $L = \{w \mid \exists w' \in \Sigma^*, \langle w, w' \rangle \in L'\}$.

Ressources utiles

— La page du cours (principalement les sujets de TD) :
`https://www.irif.fr/users/feree/calculabilite-complexite`

— Un livre de référence :
Introduction to the Theory of Computation (M. Sipser)

— Notes de cours de Leonore Blum :
`http://www.cs.cmu.edu/~lblum/flac/schedule.html`

— Le livre de Sylvain Périfel (en français) sur la complexité couvre aussi une petite partie de ce que nous abordons en calculabilité :
`https://www.irif.fr/users/sperifel/livre_complexite`