Calculabilité et Complexité: CM2

Florian Bridoux

Université de Caen

2021-2022

Table des matières

Table of Contents

Code d'une machine de Turing

Dans la suite du cours, nous allons étudier des problèmes dans lesquels une machine de Turing doit répondre à une question sur les machines de Turing.
Pour cela, il faut pouvoir donner en entrée à une machine de Turing la définition (le code, le programme) d'une autre machine de Turing. Deux possibilités :

- en donnant le numéro de la machine dans une énumération des machines de Turing,
- en écrivant le code de la machine sur la ruban.

Pour la première possibilité, il faut fixer une *énumération* des machines de Turing, c'est-à-dire fixer une bijection entre $\mathbb{N}$ et l'ensemble des machines de Turing, afin de pouvoir désigner *la* machine numéro 0, *la* machine numéro 1, *la* machine numéro 2, etc.

Code d'une machine de Turing

Pour la seconde possibilité, il faut *encoder* la définition d'une machine de Turing dans le mot d'entrée.

Notation

$\langle M \rangle$ est le **code de la machine de Turing** M.

Il y a de nombreuses façons d'encoder les machines de Turing sur le ruban. Par exemple, en numérotant de q_1 à q_n les états (avec q_1 l'état initial et q_n l'état final) et de a_1 à a_m les symboles de ruban (avec a_m le symbole blanc B) d'une machine M, et en fixant $\leftarrow = 0$ et $\rightarrow = 00$, il est possible d'encoder chaque transition $\delta(q_i, a_j) = (q_k, a_\ell, \rightarrow)$ de M par la séquence

$$\text{transition} = \underbrace{0\ldots0}_{i}1\underbrace{0\ldots0}_{j}1\underbrace{0\ldots0}_{k}1\underbrace{0\ldots0}_{\ell}1\underbrace{00}_{\rightarrow}.$$

Code d'une machine de Turing

Notation

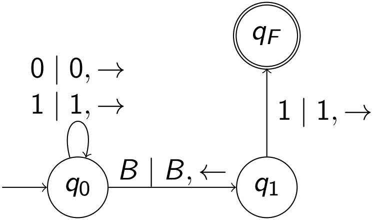
$\langle M \rangle$ est le **code de la machine de Turing** M.

On peut alors encoder une machine complète en commençant par dire combien elle a d'états, combien elle a de symboles de ruban, puis en listant ses x transitions une à une :

$$\langle M \rangle = \underbrace{0\ldots0}_{n}11\underbrace{0\ldots0}_{m}11\text{transition}_1 11\ldots11\text{transition}_x.$$

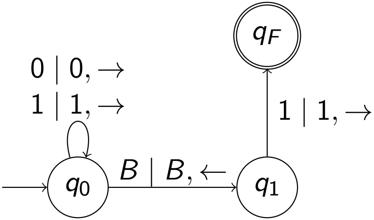
Par convention, nous pouvons énumérer les transitions dans l'ordre lexicographique selon l'état courant et le symbole lu (pratique pour décoder, mais pas nécessaire). Un tel codage est injectif, c'est-à-dire qu'une suite de bits correspond à au plus une machine de Turing. On se convaincra que le résultat suivant est vrai.

Code d'une machine de Turing: Exemple



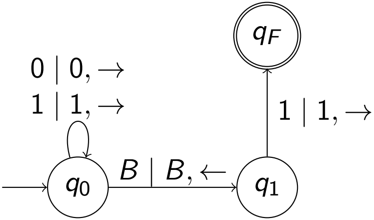
$$\langle M \rangle = \ldots$$

Code d'une machine de Turing: Exemple



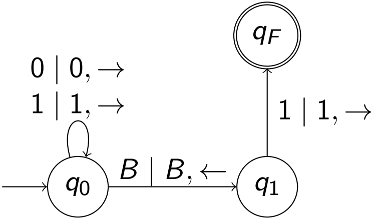
$$\langle M \rangle = \underbrace{000}_{n=|Q|} 11 \ldots$$

Code d'une machine de Turing: Exemple



$$\langle M \rangle = \underbrace{000}_{n=|Q|} 11 \underbrace{000}_{m=|\Gamma|} 11 \dots$$

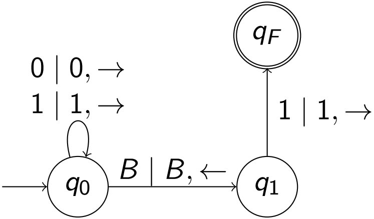
Code d'une machine de Turing: Exemple



$$\langle M \rangle = \underbrace{000}_{n=|Q|}11\underbrace{000}_{m=|\Gamma|}11t_1 11t_2 11t_3 11t_4$$

avec:

Code d'une machine de Turing: Exemple

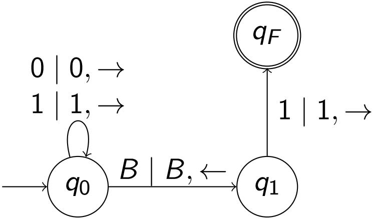


$$\langle M \rangle = \underbrace{000}_{n=|Q|} 11 \underbrace{000}_{m=|\Gamma|} 11 t_1 11 t_2 11 t_3 11 t_4$$

avec:

$$t_1 = \underbrace{0}_{q_0} 1 \underbrace{0}_{0} 1 \underbrace{0}_{q_0} 1 \underbrace{0}_{0} 1 \underbrace{00}_{\rightarrow},$$

Code d'une machine de Turing: Exemple

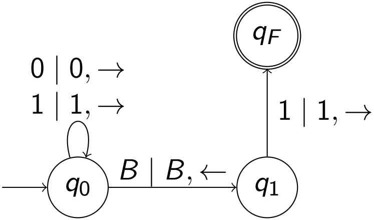


$$\langle M \rangle = \underbrace{000}_{n=|Q|} 11 \underbrace{000}_{m=|\Gamma|} 11 t_1 11 t_2 11 t_3 11 t_4$$

avec:

$$t_1 = \underbrace{0}_{q_0} 1 \underbrace{0}_{0} 1 \underbrace{0}_{q_0} 1 \underbrace{0}_{0} 1 \underbrace{00}_{\rightarrow}, \quad t_2 = \underbrace{0}_{q_0} 1 \underbrace{00}_{1} 1 \underbrace{0}_{q_0} 1 \underbrace{00}_{1} 1 \underbrace{00}_{\rightarrow},$$

Code d'une machine de Turing: Exemple



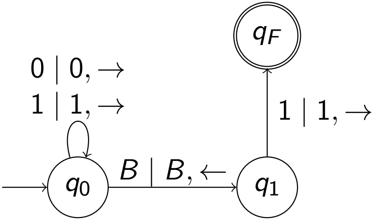
$$\langle M \rangle = \underbrace{000}_{n=|Q|}11\underbrace{000}_{m=|\Gamma|}11t_1 11t_2 11t_3 11t_4$$

avec:

$$t_1 = \underbrace{0}_{q_0}1\underbrace{0}_{0}1\underbrace{0}_{q_0}1\underbrace{0}_{0}1\underbrace{00}_{\rightarrow}, \quad t_2 = \underbrace{0}_{q_0}1\underbrace{00}_{1}1\underbrace{0}_{q_0}1\underbrace{00}_{1}1\underbrace{00}_{\rightarrow},$$

$$t_3 = \underbrace{0}_{q_0}1\underbrace{000}_{B}1\underbrace{00}_{q_1}1\underbrace{000}_{B}1\underbrace{0}_{\leftarrow},$$

Code d'une machine de Turing: Exemple



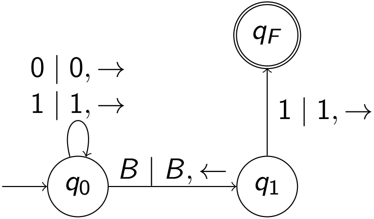
$$\langle M \rangle = \underbrace{000}_{n=|Q|} 11 \underbrace{000}_{m=|\Gamma|} 11 t_1 11 t_2 11 t_3 11 t_4$$

avec:

$$t_1 = \underbrace{0}_{q_0} 1 \underbrace{0}_{0} 1 \underbrace{0}_{q_0} 1 \underbrace{0}_{0} 1 \underbrace{00}_{\rightarrow}, \quad t_2 = \underbrace{0}_{q_0} 1 \underbrace{00}_{1} 1 \underbrace{0}_{q_0} 1 \underbrace{00}_{1} 1 \underbrace{00}_{\rightarrow},$$

$$t_3 = \underbrace{0}_{q_0} 1 \underbrace{000}_{B} 1 \underbrace{00}_{q_1} 1 \underbrace{000}_{B} 1 \underbrace{0}_{\leftarrow}, \quad t_4 = \underbrace{00}_{q_1} 1 \underbrace{00}_{1} 1 \underbrace{000}_{q_F} 1 \underbrace{00}_{1} 1 \underbrace{00}_{\rightarrow}.$$

Code d'une machine de Turing: Exemple



$$\langle M \rangle = \underbrace{000}_{n=|Q|} 11 \underbrace{000}_{m=|\Gamma|} 11 t_1 11 t_2 11 t_3 11 t_4$$

avec:

$$t_1 = \underbrace{0}_{q_0} 1 \underbrace{0}_{0} 1 \underbrace{0}_{q_0} 1 \underbrace{0}_{0} 1 \underbrace{00}_{\rightarrow}, \quad t_2 = \underbrace{0}_{q_0} 1 \underbrace{00}_{1} 1 \underbrace{0}_{q_0} 1 \underbrace{00}_{1} 1 \underbrace{00}_{\rightarrow},$$

$$t_3 = \underbrace{0}_{q_0} 1 \underbrace{000}_{B} 1 \underbrace{00}_{q_1} 1 \underbrace{000}_{B} 1 \underbrace{0}_{\leftarrow}, \quad t_4 = \underbrace{00}_{q_1} 1 \underbrace{00}_{1} 1 \underbrace{000}_{q_F} 1 \underbrace{00}_{1} 1 \underbrace{00}_{\rightarrow}.$$

Au final: $\langle M \rangle = 0001100011010101010011010010100100110100010010001011001001000100100$

Code d'une machine de Turing

Lemme

Le langage $L_{enc} = \{w \in \{0,1\}^* \mid w = \langle M \rangle$ pour une MT $M\}$ est décidable.

Autrement dit, on peut écrire une machine de Turing qui reconnaît les mots binaires qui encodent une machine de Turing de la façon que l'on vient de définir.

Table of Contents

Problème de l'arrêt

Définition la fonction **halt**

$$\text{halt} : (\langle M \rangle, w) \mapsto \begin{cases} 0 \text{ si } M(w) \uparrow \\ 1 \text{ sinon} \end{cases} .$$

où $M(w) \uparrow$ signifie que M lancée sur l'entrée w entre dans une boucle infinie (et ne termine donc pas).

Théorème de l'arrêt

La fonction **halt** n'est pas calculable.

Problème de l'arrêt

Preuve du théorème de l'arrêt:

- Par l'absurde, supposons qu'il existe M_{halt} qui prend un mot $\langle M \rangle \# w$ et accepte si et seulement si $M(w)$ s'arrête.
- Nous pouvons alors construire (*preuve au tableau*) la machine M_{diag} suivante :

$$M_{diag}(i) = \begin{cases} 1 \text{ si } M_{halt}(i,i) = 0 \\ \uparrow \ \text{ si } M_{halt}(i,i) = 1 \end{cases}$$

Lançons maintenant M_{diag} sur son propre code. Deux cas possibles:

- Si $M_{diag}(\langle M_{diag} \rangle) = 1$ alors, par définition de M_{diag}, nous avons $M_{halt}(\langle M_{diag} \rangle, \langle M_{diag} \rangle) = 0$ ce qui signifie, par définition de M_{halt}, que $M_{diag}(\langle M_{diag} \rangle) \uparrow$, une contradiction.
- Si $M_{diag}(\langle M_{diag} \rangle) \uparrow$ alors, par définition de M_{diag}, nous avons $M_{halt}(\langle M_{diag} \rangle, \langle M_{diag} \rangle) = 1$ ce qui signifie, par définition de M_{halt}, que $M_{diag}(\langle M_{diag} \rangle)$ s'arrête, une contradiction.

Dans les deux cas nous arrivons à une contradiction.

Problème de l'arrêt

Théorème

Le langage $L_{halt} = \{\langle M\rangle\#w \mid M(w) \text{ s'arrête } \}$ n'est pas décidable.
Il est en revanche semi-décidable et non co-semi-décidable

La preuve de semi-décidabilité est très compliquée à écrire rigoureusement (beaucoup de détails techniques). Idée de preuve:

- On peut crée une machine de Turing qui prend en entrée le code d'une autre machine de Turing M et la simule pas à pas sur une entrée w quelconque. (*début de preuve dans le TD2 exercice 6*)
- Si la machine simulée s'arrête sur l'entrée w alors la machine stimulante aussi et le mot $\langle M\rangle\#w$ est reconnu (le langage est donc semi-décidable).
- En revanche, si la machine M ne s'arrête pas sur l'entrée w alors la machine stimulante non plus.

Comme le langage est semi-décidable mais pas décidable alors il n'est pas co-semi-décidable.

Problème de l'arrêt

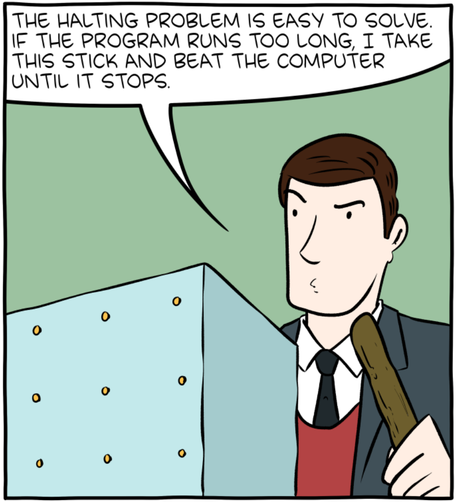
Théorème

Le langage $L_{halt} = \{\langle M\rangle\#w \mid M(w) \text{ s'arrête } \}$ n'est pas décidable. Il est en revanche semi-décidable et non co-semi-décidable

Question

Pouvez-vous trouver un langage co-semi-décidable mais pas semi-décidable?

Problème de l'arrêt



What if Alan Turing had been an engineer?