# Course - Simulating Turing Machines by Analytic Functions

Olivier Bournez

November 16, 2018

## 1 Coding of configurations of Turing machines

Source: [1]

### 1.1 Turing machines

We want to obtain a map that captures the behavior of the transition function of a Turing Machine.

Without loss of generality, consider a Turing machine M using 10 symbols, the blank symbol $B = 0$, and symbols $1, 2, ...9$. Let

$$\ldots BBBa_{-k}a_{-k+1}\ldots a_{-1}a_0a_1\ldots a_nBBB\ldots.$$

(1) represent the tape contents of the Turing machine $M$. We suppose the head to be reading symbol $a_0$ and $a_i \in \{0, 1, \ldots, 9\}$ for all $i$. We also suppose that M has $m$ states, represented by numbers 1 to $m$. For convenience, we consider that if the machine reaches a halting configuration it moves to the same configuration. We assume that, in each transition, the head either moves to the left, moves to the right, or does not move.

### 1.2 Coding a configuration: using integers.

Take

$$\begin{aligned} y_1 &= a_0 + a_1 10 + \cdots + a_n 10^n, \\ y_2 &= a_{-1} + a_{_2} 10 + \cdots + a_{-k} 10^{k-1}, \end{aligned}$$

and let $q$ be the state associated to the current configuration.

Then the triple

$$\gamma_{\mathbb{N}^3}^{TM}(C) = (y_1, y_2, q) \in \mathbb{N}^3$$

encodes the current configuration $C$ of $M$ by an element of $\mathbb{N}^3$.

### 1.3 Coding a configuration: using integers (variant).

Olivier: modifié légérement codage par rapport à [2]

⚠ ATTENTION

Then the triple

$$\gamma_{\mathbb{N}^2}^{TM}(C) = (q + (m+1)y_1, y_2, q) \in \mathbb{N}^3$$

encodes the current configuration $C$ of $M$ by an element of $\mathbb{N}^2$.

### 1.4 Coding a configuration: using $(0,1)$ and $\arctan$.

Consider $\nu : \mathbb{N} \to (0,1)$ defined by

Then the triple

$$\gamma_{(0,1)^2}^{TM}(C) = \left( \frac{2}{\pi} \arctan(y_1), \frac{2}{\pi} \arctan(y_2), \frac{2}{\pi} \arctan(q) \right)$$

### 1.5 Coding a configuration: using $[0,1]$

Take

$$\begin{array}{rcl} y_1 & = & a_0 10^{-1} + a_1 10^{-2} + \cdots + a_n 10^{-n-1}, \\ y_2 & = & a_{-1} 10^{-1} + a_{_2} 10^{-2} + \cdots + a_{-k} 10^{-k}, \end{array}$$

and let $q$ be the state associated to the current configuration.

Then the triple

$$\gamma_{[0,1]}^{TM}(C) = (y_1, y_2, q) \in [0,1]^2 \times \{1, 2, \ldots, m\}$$

encodes the current configuration $C$ of $M$ by an element of $[0,1]^2 \times \{1, 2, \ldots, m\}$.

## 2 Discrete time simulation: $\gamma_{\mathbb{N}^2}^{TM}$

### 2.1 Koiran-Moore 99's theorem

**Definition 1** *Let $U_n$ be the smallest class of functions $f : \mathbb{R}^n \to \mathbb{R}$ containing rational constants, $\pi$, the $n$ projections $x \mapsto x_i$ and satisfying the following closure properties:*

- *if $f, g \in U_n$ then $f \oplus g \in U_n$, where $\oplus \in \{+, -, \times\}$*
- *if $f \in U_n$ then $\sin(f) \in U_n$*

*We will say that $f : \mathbb{R}^n \to \mathbb{R}^n$ is elementary if its $n$ components are in $U_n$.*

../.. could be potentially: disucssion about dimension 1 of [2]: simulation in exponential time using counter

**Theorem 1 (Koiran-Moore's Theorem [2])** *For any Turing machine $M$ and any input $w$, there is an elementary function $f$ on two variables and constants $a$ and $b$ such that $M$ halts on input $w$ after $t$ steps if and only if $f^{[t]}(a+bw,0) = (0,0)$.*

BEGIN-PERSO: Important que linear time dans ce résultat, sinon plus facile en utilsant dimension 1: cf papier. END-PERSO

### 2.1.1 Proof of Koiran-Moore 99's theorem

If we define

$$h_p(x) = \left(\frac{\sin(\pi x)}{p \sin \frac{\pi x}{p}}\right)^2$$

then we have (for integer $t$ and $a$)

$$\begin{aligned} h_{10(m+1)}(x - (t + (m+1)a)) &= \begin{cases} 1 & \text{if } q = t \text{ and } a_0 = a \\ 0 & \text{otherwise} \end{cases} \\ h_{10}(y - a) &= \begin{cases} 1 & \text{if } a_{-1} = a \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Then let $q_{next} = q_{next}(q, a_0)$ be the Turing machine's new state, $a' = a'(q, a_0)$ be the symbol it writes on the tape, $H = H(q, a_0)$ its movement left or right with the convention that $H = 0$ for halt states and $+/-1$ for non-halting states.

Then if we define

$$(x_{right}, y_{right}) = (q_{next} + \frac{x - q - (m+1)a_0}{10}, 10y + a')$$

$$(x_{left}, y_{left}) = (q_{next} + 10(x - q + (m+1)(a' - a_0)) + (m+1)a_{-1}, \frac{y - a_{-1}}{10})$$

corresponding to shifting the machine to the right or left, we can simulate the Turing machine with the function:

$$f(x,y) = \sum_{q=1}^{m} \sum_{a_0=0}^{10} H^2(s, a_0) \cdot h_{10(m+1)}(x - (q + (m+1)a_0)) \times$$

$$\left[ \left(\frac{1 + H_{q,a_0}}{2}\right) \cdot (x_{right}, y_{right}) + \left(\frac{1 - H_{q,a_0}}{2}\right) \sum_{a_{-1}=0}^{10} h_{10(y-a_{-1})} \cdot (x_{left}, y_{left}) \right]$$

An initial TM state $q_0$ with an input $w$ on the right half of the tape corresponds to an initial point $(q_0 + nw, 0)$. If the machine erases the tape before halting, and if the state is $s = 0$, halting is indicated by arriving at $(0,0)$.

# 3 Discrete time simulation: $\gamma_{\mathbb{N}^3}^{TM}$

## 3.1 Graça-Campagnolo-Buescu's Theorem 1

We now can state the first main result of this paper as follows:

**Theorem 2 (Graça-Campagnolo-Buescu's Theorem 1 [1])** *Let $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of a Turing machine $M$, under the encoding $\gamma_{\mathbb{N}^3}^{TM}$ described above and let $0 < \delta < \epsilon < 1/2$. Then $\theta$ admits an analytic extension $f_M : \mathbb{R}^3 \to \mathbb{R}^3$, robust to perturbations in the following sense: for all $f$ such that $\|f - f_M\| \leq \delta$, and for all $\overline{x_0} \in \mathbb{R}^3$ satisfying $\|\overline{x_0} - x_0\| \leq \epsilon$, where $x_0 \in \mathbb{N}^3$ represents an initial configuration,*

$$\left\| f^{[j]}(\overline{x_0}) - \theta^{[j]}(x_0) \right\| \leq \epsilon \text{ for all } j \in \mathbb{N}.$$

A few remarks are in order. First, and as noticed before, we implicitly assumed that if $y$ is a halting configuration, then $\theta(y) = y$. Secondly, we notice that the upper bound $1/2$ on $\epsilon$ results from the chosen encodingn, which is over the integers. In fact, the bound is maximal with respect to that encoding.

## 3.2 Some basic functions

### 3.2.1 Mod function $\omega$

Variante de la fonction d'avant en réalité

ATTENTION

This section is devoted to the presentation of results that, while not very inter- esting on their own, will be useful when proving Theorem 1.

As our first task, we introduce an analytic extension $\omega : \mathbb{R} \to \mathbb{R}$ for the function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(n) = n \mod 10$ . This function will be necessary when simulating Turing machines. It will be used to read symbols written in the tape.

To achieve this purpose, we can use a periodic function, of period 10, such that $\omega(i) = i$, for $i = 0, 1, ..., 9$. Then, using trigonometric interpolation (cf. [22, pp. 176-182]), one may take

$$\omega(x) = a_0 + a_5 \cos(\pi x) + \left( \sum_{j=1}^{4} a_j \cos\left( \frac{j\pi x}{5} + b_j \sin(\frac{j\pi x}{5} \right) \right), \tag{1}$$

where $a_0, \ldots, a_4, b_1, \ldots, b_4$ are computable coefficients taht can be explicitely obtained by solving a systeme of linear equations.

It is easy to see that $\omega$ is uniformly continuous in $\mathbb{R}$. Hence, for every $\epsilon \in (0, 1/2)$, there will be some $\eta_\epsilon > 0$ satisfying

$$\forall n,\ x \in [n - \eta_\epsilon, n + \eta_\epsilon] \Rightarrow |\omega(x) - n \mod 10| \leq \epsilon. \tag{2}$$

### 3.2.2 Error correcting function $\sigma$

When simulating a Turing machine, we will also need to keep the error under control. In many cases, this will be done with the help of the error-contracting function defined by

$$\sigma(x) = x - 0.2 \sin(2\pi x).$$

The function $\sigma$ is a contraction on the vicinity of integers:

**Lemma 1** *Let $n \in \mathbb{Z}$, and let $\epsilon \in [0, 1/2)$. Then there is some contracting factor $\lambda_\epsilon \in (0, 1)$ such that $\forall \delta \in [-\epsilon, \epsilon]$, $|\sigma(n + \delta) - n| < \lambda_\epsilon \delta$.*

Throughout the remainder of this paper, we suppose that $\epsilon \in [0, 1/2)$ is fixed and that $\lambda_\epsilon$ is the respective contracting factor given by Lemma 1.

The function $\sigma$ will be used in our simulation to keep the error controlled when bounded quantities are involved (e.g., the actual state, the symbol being read, etc.).

### 3.2.3 Error correcting function $l_3$

We will also need another error-contracting function that controls the error for unbounded quantities. This will be achieved with the help of the function $_3 : \mathbb{R}^2 \to \mathbb{R}$, that has the property that whenever $\overline{a}$ is an approximation of $a \in \{0, 1, 2\}$, then $|l_3(\overline{a}, l)| < 1/y$, for $y > 0$. In other words, $l_3$ is an error-contracting map, where the error is contracted by an amount specified by the second argument of$l_3$.

We start by defining a preliminary function l2 satisfying similar conditions, but only when $a \in \{0, 1\}$.

**Lemma 2** *Let $l_2 : \mathbb{R}^2 \to \mathbb{R}$ be given by $l_2(x, y) = \frac{1}{\pi} \arctan(4y(x - 1/2)) + 1/2$. Suppose that $a \in \{0, 1\}$. Then, for any $\overline{a}, y \in \mathbb{R}$ satisfying $|a - \overline{a}| \leq 1/4$ and $y > 0$, we get $|a - l_2(\overline{a}, y)| < 1/y$.*

**Lemma 3** *Let $a \in \{0, 1, 2\}$ and let $l_3 : \mathbb{R}^2 \to \mathbb{R}$ be given by*

$$l_3(x, y) = l_2((\sigma^{[d+1]}(x) - 1)^2, 3y)\dot{(}2l_2(\sigma^{[d]}(x)/2, 3y) - 1) + 1,$$

*where $d = 0$ if $\epsilon \leq 1/4$ and $d = \lceil -\log(4\epsilon)/\log \lambda_\epsilon \rceil$ otherwise. Then for any $\overline{a}, y \in \mathbb{R}$ satisfing $|a - \overline{a}| \leq \epsilon$ and $y \geq 2$ we have $|a - l_3(\overline{a}, y)| < 1/y$.*

### 3.2.4 An observation

The following lemma can be easily proved by induction on $n$.

**Lemma 4** *If $|\alpha_i|$, $|\overline{\alpha}_i| \leq K$ for $i = 1, \ldots, n$ then*

$$\alpha_1 \ldots \alpha_n - \overline{\alpha}_1 \ldots \overline{\alpha}_n| \leq (|\alpha_1 - \overline{\alpha}_1| + \ldots |\alpha_n - \overline{\alpha}_n|)K^{n-1}.$$

## 3.3 Another statement

In this section we show, in a constructive manner, how to simulate a Turing machine with an analytic map robust to (small) perturbations. We will first prove the following theorem.

**Theorem 3** *Let $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of some Turing machine. Then, given some $0 \leq \epsilon < 1/2$, $\theta$ admits an analytic extension $h_M : \mathbb{R}^3 \to \mathbb{R}^3$ with the property that*

$$\|(y_1, y_2, q) - (\overline{y}_1, \overline{y}_2, \overline{q})\| \leq \epsilon \Rightarrow \|\theta(y_1, y_2, q) - \theta(\overline{y}_1, \overline{y}_2, \overline{q})\| \tag{3}$$

## 3.4 Proof of this statement

**Proof**: We will show how to construct $h_M$ with analytic functions:

1. **Determine the symbol being read.** Let $a_0$ be the symbol being actually read by the Turing machine $M$. Then $\omega(y_1) = a_0$, where $\omega$ is given by (1).

   But what about the effect of the error present in $\overline{y}_1$?

   Since $|y_1 - \overline{y}_1| \leq \epsilon$ , ,

   $$a_0 - \omega \circ \sigma^{[l]}(\overline{y}_1)| \leq \epsilon, \quad \text{with } l = \left]\left|\frac{\log(\chi_\epsilon/\epsilon)}{\log \lambda_\epsilon}\right|\right[, \tag{4}$$

   where $\chi_\epsilon$ is given (2). Then pick $\overline{y} = \omega \circ \sigma^{[l]}(\overline{y}_1)$ as an approximation of the symbol being currently read. Similarly, $\omega \circ \sigma^{[l]}(\overline{y}_2)$ gives an approximation of $a_{-1}$, with error bounded by $\epsilon$.

2. **Determine the next state.** The map that returns the next state is defined by polynomial interpolation. This can be done as follows. Let $y$ be the symbol being currently read and $q$ the current state. Recall that $m$ denotes the number of states and $k = 10$ is the number of symbols. One may take

   $$q_{next} = \sum_{i=0}^{9} \sum_{j=1}^{m} \left( \prod_{r=0, r\neq i}^{9} \frac{(y-r)}{(i-r)} \right) \left( \prod_{s=1, s\neq j}^{m} \frac{(q-s)}{(j-s)} \right) q_{i,j},$$

   where $q_{i,j}$ is the sate that follows symbol $i$ and state $j$.

   However, we are dealing with the approximations $\overline{q}$ and $\overline{y}$.

   Therefore, we define

   $$q_{next} = \sum_{i=0}^{9} \sum_{j=1}^{m} \left( \prod_{r=0, r\neq i}^{9} \frac{(\sigma^{[n]}(\overline{y}) - r)}{(i-r)} \right) \left( \prod_{s=1, s\neq j}^{m} \frac{(\sigma^{[n]}(\overline{q}) - s)}{(j-s)} \right) q_{i,j}, \tag{5}$$

with

$$n = \left\lceil \frac{\log(10m^2K^{m+7}(m+8)}{-\log \lambda_\epsilon} \right\rceil, \quad K = \max\{9.5, m+1/2\}.$$

With this choice for n, the error of $\sigma^{[n]}(\overline{y})$ and $\sigma^{[n]}(\overline{q})$ is such that

$$9|y - \sigma^{[n]}(\overline{y})| + (m-1)|q - \sigma^{[n]}(\overline{q})| \leq \frac{\epsilon}{10m^2K^{m+7}}. \tag{6}$$

Thus from (5), (6) and Lemma 4, w conclude that $|\overline{q}_{next} - q_{next}| \leq \epsilon$.

3. **Determine the symbol to be written on the tape.** Using a similar construction, the symbol to be written, $s_{next}$, can be approximated with precision $\epsilon$ , i.e. $|s_{next} - \overline{s}_{next}| \leq \epsilon$.

4. **Determine the direction of the move for the head.** Let $h$ denote the direction of the move of the head, where $h = 0$ denotes a move to the left, $h = 1$ denotes a "no move", and $h = 2$ denotes a move to the right. Then, again, the "next move" $h_{next}$ can be approximated by means of a polynomial interpolation as in steps 3 and 4, therefore obtaining $|h_{next} - \overline{h}_{next}| \leq \epsilon$.

5. **Update the tape contents.** We define functions $\overline{P}_1$, $\overline{P}_2$, $\overline{P}_3$, which are intended to approximate the tape contents after the head moves left, does not move, or moves right, respectively. Let $H$ be a "sufficiently good" approximation of $h$ , yet to be determined. Then, the next value of $y$ , $_1^{next}$, can next be approximated by

$$\overline{y}_1^{next} = \overline{P}_1\frac{1}{2}(1-H)(2-H) + \overline{P}_2H(2-H) + \overline{P}_3(-\frac{1}{2})H(1-H), \tag{7}$$

with

$$\begin{aligned}
\overline{P}_1 &= 10(\sigma^{[j]}(\overline{y}_1) + \sigma^{[j]}(\overline{s}_{next}) - \sigma^{[j]}(\overline{y}) + \sigma^{[j]} \circ \omega \circ \sigma^{[j]}(\overline{y}_2) \\
\overline{P}_2 &= \sigma^{[j]}(\overline{y}_1) + \sigma^{[j]}(\overline{s}_{next}) - \sigma^{[j]}(\overline{y}) \\
\overline{P}_3 &= \frac{\sigma^{[j]}(\overline{y}_1) - \sigma^{[j]}(\overline{y})}{10},
\end{aligned}$$

where $j \in \mathbb{N}$ is sufficiently large and l is given by (4). Notice that when exact values are used, $\overline{y}_1^{next} = y_1^{next}$. ynext. The problem in this case is that $\overline{P}_1$ depends on $\overline{y}_1$, which is not a bounded value. Thus, if we simply take $\overline{H} = \overline{h}_{next}$ the error of the term $(1-H)(2-H)/2$ is arbitrarily amplified when this term is multiplied by $\overline{P}_1$. Hence, $\overline{H}$ must be a sharp estimate of $h_{next}$, proportional to $\overline{y}_1$. Therefore, using Lemma 3 and the definition of $y_1$, one can see that it is suffices to take

$$H_3 = l_3(\overline{h}_3, 10000(\overline{y}_1 + 1/2) + 2).$$

Using the same argument for $\overline{P}_2$ and $\overline{P}_3$ we conclude that $|overliney_1^{next} - y_1^{next}| < \epsilon$.

Similarly, and for the left side of the tape, we can define $\overline{y}_2^{next}$ such that $|overliney_2^{next} - y_2^{next}| < \epsilon$.

Finally, $h_M : \mathbb{R}^3 \to \mathbb{R}^3$ is defined by $h_M(\overline{y}_1, \overline{y}_2, \overline{y}_1, \overline{q}_{next})$.

□

### 3.5 Proof of Graça-Campagnolo-Buescu's Theorem 1

Let $0 \leq \delta < \epsilon$ . Then, using Theorem 3, one can find a map $h_M$ such that (3) holds. Let $i \in \mathbb{N}$ satisfy $\sigma^{[i]}(\epsilon) \leq \epsilon - \delta$ . Define a map $f_M = \sigma^{[i]} \circ h_M$ . Then, if $x_0 \in \mathbb{N}^3$ is an initial configuration,

$$\|\overline{x}_0 - x_0\| \leq \epsilon \Rightarrow \|f_M(\overline{x}_0) - \theta(\overline{x}_0)\| \leq \epsilon - \delta.$$

Thus by triangular inequality, if $\|\overline{x}_0 - x_0\|$, then

$$\|f_M(\overline{x}_0) - \theta(\overline{x}_0)\| \leq \|f_M(\overline{x}_0) - f_M(\overline{x}_0))\| + \|f_M(\overline{x}_0)) - \theta(\overline{x}_0)\| \leq \delta + (\epsilon - \delta) = \epsilon$$

This proves the result for $j = 1$. For $j > 1$, we proceed by induction.

## References


[1] Daniel S. Graça, Manuel L. Campagnolo, and Jorge Buescu. Robust simulations of Turing machines with analytic maps and flows. In B. Cooper, B. Loewe, and L. Torenvliet, editors, *Proceedings of CiE'05, New Computational Paradigms*, volume 3526 of *Lecture Notes in Computer Science*, pages 169–179. Springer-Verlag, 2005.

[2] P. Koiran and Cristopher Moore. Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. *Theoret. Comput. Sci.*, 210(1):217–223, 1999.