Olivier Bournez

Version of October 22, 2021

Chapter 1

Preliminaries

These are Course Notes for MPRI Course 2.33.1.
Theories of Computation.

Any comment (even about orthography) welcome: send an email to bournez@lix.polytechnique.fr

Contents

Chapter 2

Richardson Theorem

Chapter 3

Formal Calculus: Richardson's Theorem

This chapter presents the following two chapters: the purpose of all these chapters is to prove that simplification in formal calculus is not possible in the general case.

3.1 Richardson 68's Theorem

3.1.1 The theorem

Theorem 3.1 (Richardson 68)

- *Let E be a set of expressions representing real partial functions $f : \mathbb{R} \to \mathbb{R}$. Let E^* be the set of functions represented by expressions in E.*
- *Assume that $E*$:*
  - *contains identify, rational numbers as constant functions,*
  - *is closed*[a] *under addition, subtraction, multiplication and composition.*
  - *Assume that E^* contains* $\log(2)$, π, e^x, $\sin(x)$.
- *Then, given an expression A in E, determining whether there is some real number x with $A(x) < 0$ is unsolvable.*

[a] There is an effective procedure for finding expressions in E to represent $A(x) + B(x)$ from representation of $A(x)$ and $B(x)$, and similarly for other operations.

(remark: the theorem is not stated here with minimal hypotheses. It is stated in its original form. We will see in chapter 4 that for example constant π and $\log(2)$ can be avoided, using Matiyasevich's result, instead of Davis-Putnam-Robinson's theorem, as it was originally done by Richardson).

3.1.2 Proof Idea

A formal proof is given in next two chapters.

Basically,

Diophantine equations

- There is some polynomial $P(y, x_1, \cdots, x_n)$, with integral coefficients in $y, x_1, \cdots, x_n$, $2^{x_1}, \cdots, 2^{x_n}$, for which the predicate

  $$\exists x_1 \cdots x_n \in \mathbb{N} \text{ s.t. } P(y, x_1, \cdots, x_n) = 0$$

  is not recursive as y varies over the natural numbers.

can be embedded into $\mathbb{R}$:

- Hence, the predicate

  $$\exists x_1 \cdots x_n \in \mathbb{R} \text{ s.t. } P^2(y, x_1, \cdots, x_n) + \sum_{t=1}^{n} \sin^2(\pi x_i) = 0$$

  is not recursive as y varies over the natural numbers.

- Hence, the predicate

  $$\exists x_1 \cdots x_n \in \mathbb{R} \text{ s.t. } K(y, x_1, \cdots, x_n) * (P^2(y, x_1, \cdots, x_n) + \sum_{t=1}^{n} \sin^2(\pi x_i)) < 1$$

  is not recursive as y varies over the natural numbers.

- Hence, the predicate

  $$\exists x_1 \cdots x_n \in \mathbb{R} \text{ s.t. } K(y, x_1, \cdots, x_n) * (P^2(y, x_1, \cdots, x_n) + \sum_{t=1}^{n} \sin^2(\pi x_i)) - 1 < 0$$

  is not recursive as y varies over the natural numbers.

3.1.3 Richardson 68's Theorem (continued)

Theorem 3.2 (Richardson 68)

- *Let E be a set of expressions representing real partial functions $f : \mathbb{R} \to \mathbb{R}$. Let E^* be the set of functions represented by expressions in E.*
- *Assume that $E*$:*
  - *contains identify, rational numbers as constant functions,*
  - *is closed[a] under addition, subtraction, multiplication and composition.*
  - *Assume that E^* contains* $\log(2)$, π, e^x, $\sin(x)$.
- *Then, given an expression A in E, determining whether there is some real*

number x with $A(x) \equiv 0$ is unsolvable.

[a] There is an effective procedure for finding expressions in E to represent $A(x) + B(x)$ from representation of $A(x)$ and $B(x)$, and similarly for other operations.

3.1.4 Proof Idea

A formal proof will be given in two chapters.

Basically,
$\exists x\ G(n,x) < 1$ iff $|G(n,x) - 1| - (G(n,x) - 1) \equiv 0$.

3.2 Consequences

From a Formal Calculs point of view:

- simplification is hard in the general case.
- computer algebra is about isolating classes for which algorithms exist.

3.3 Results in this spirit

- Determining whether a polynomial dynamical system has a Hopf bifurcation is undecidable [da-Costa Doria 94].
- "Dynamical Systems where proving chaos is equivalent to proving Fermat's conjecture" [da-Costa Doria Amaral 92].

Chapter 4

Richardson's Theorem

We denote by Σ_k the set of functions of k variables, built from constant 1, addition, subtraction, multiplication, and sinus. We note by Σ the union of the Σ^k.

We admit in this chapter the following result, proved in Chapter 5:

Theorem 4.1 *There is no algorithm that can decide for a Diophantine equation (that is to say an equation $P(x_1, \cdots, x_k) = 0$, for P a polynomial) whether or not it has a solution in natural numbers.*

4.1 From Integers to Reals

Lemma 4.1 *Let $a/b < c/d$ be two rational numbers. There exists some polynomial with integer coefficients whose set of real roots, projected on the first coordinate, is exactly interval $[a/b, c/d]$.*

Proof: We start from equation $x - y^2 = 0$ whose solution is the set of (x, y^2) where y is in $\mathbb{R}$ and $x \geq 0$. By translation and change of sign, we consider $(x - a/b - y^2)^2 + (c/d - x - z^2)^2$ then

$$(bx - a - y^2)^2 + (c - dx - z^2)^2 = 0$$

whose real roots (x, y, z) have their first coordinate in $[a/b, c/d]$. □

Lemma 4.2 *There is a function $f \in \Sigma$ whose only real root has π has first coordinate.*

Proof: Using the fact that $\pi \in [3, 22/7]$, previous Lemma, and the fact that a sum of squares is null iff each term is null, combined with $\sin(x) = 0$, we just need to consider

$$f(x, y, z) = \sin^2(x) + (x - 3 - y^2)^2 + (22 - 7x - z^2)^2 = 0.$$

□

Proposition 4.1 *There is no algorithm that takes as input a function $f \in \Sigma_k$ for an arbitrary k, and decides whether equation $f(x_1,\cdots,x_k) = 0$ has a solution (over $\mathbb{R}^k$).*

Proof: We reduces the problem of solving Diophantine equations to this problem. If $P \in \mathbb{Z}[X_1,\dots,X_k]$, then the existence of integer solutions to P is equivalent to the existence of real solutions to the equation in $k+3$ variables

$$\Phi(x,y,z,x_1,\dots,x_k) = \sin^2(x) + (x-3-y^2)^2 + (22-7x-z^2)^2 + P(x_1,\dots,x_k)^2 + \sin^2(x_1 x) + \sin^2(x_2 x) + \cdots + \sin^2(x_k x) = 0.$$

□

4.2 From Equalities to Inequalities

Lemma 4.3 *For all $f \in \Sigma_k$, there exists a polynomial $g \in \mathbb{Z}[X_1,\dots,X_k]$ such that*

1. *$g(x_1,\dots,x_k) > 1$ for all $(x_1,\dots,x_k) \in \mathbb{R}^k$;*
2. *$f(x_1+\delta_1,\dots,x_k+\delta_k) < g(x_1,\dots,x_k)$ for all $(x_1,\dots,x_k) \in \mathbb{R}^k$ and all reals δ_i with $|\delta_i| < 1$, $i = 1,\dots,k$.*

We says in that case that f is dominated by g.

Proof: By induction on the construction of f. Constant 1 is dominated by 2, each variable x_i is dominated by $x_i + 2$. Then, if f_1 and f_2 are dominated by g_1 and g_2, then $f_1 - f_2$ and $f_1 + f_2$ are dominated by $g_1 + g_2$, and $f_1 f_2$ by $g_1 g_2$, whereas $\sin f_1$ is dominated by 2. □

Proposition 4.2 *There does not exist an algorithm that takes as input a function $f \in \Sigma_k$ for some arbitrary k, and that decides if inequality $f(x_1,\dots,x_k) < 1$ has a real solution.*

Proof: Starting from $P \in \mathbb{Z}[X_1,\dots,X_k]$, we consider previous function Φ. We consider $M^2\Phi \le 1$ with M still to be determined. If $(x,y,z,y_1,\dots,y_k)$ is a real solution to this inequality, one wants to see how M sufficiently large force the y_i to be integers.

First, we can control the approximation of π. Inequality $M^2\Phi \le 1$ implies

$$-\frac{1}{M} < x-3-y^2 < \frac{1}{M}, \quad -\frac{1}{M} < 22-7x-z^2 < \frac{1}{M}, \quad |\sin(x)| < \frac{1}{M}.$$

From first two inequalities, we deduce first that $x \in [3-1/M, 22/7+1/(7M)]$. Then we can link $x-\pi$ to M by mean value theorem (Théorème des accroisssements finis).

$$\sin(x) = \sin(x) - \sin(\pi) = (x-\pi)\cos(\theta),$$

with θ in the above interval, and hence taking $M > 2$, we get $|x-\pi| < 2/M$.

Then, we control the distance between y_i and its integer part, that we will denote by x_i. Inequality $M^2\Phi \le 1$ implies $|\sin(y_i x)| < 1/M$, and so there exists some multiple $k_i\pi$ of π with $k_i \in \mathbb{Z}$ such that $|y_i x - k_i\pi| < 1/M$. But then, $|y_i - k_i| < |y_i - y_i x/\pi| + |y_i x/\pi - k_i| < (|y_i|+1)/M$, and hence by considering M as a polynomial that dominates $X_i + 1$, we deduce that $k_i = x_i$ and a controlled distance.

We can hence control the polynomial P in the integer part of the solution of $M^2\Phi \le 1$:

$$\begin{aligned} P(x_1,\dots,x_k) &\le |P(y_1,\dots,y_k)| + |P(x_1,\dots,x_k) - P(y_1,\dots,y_k)| \\ &< \frac{1}{M} + \sum_{i=1}^{k} \left|\frac{\partial P}{\partial x_i}\right| |x_i - y_i| < \frac{1}{M}\left(1 + \sum_{i=1}^{k}\left|\frac{\partial P}{\partial x_i}\right| |y_i + 1|\right). \end{aligned}$$

To conclude it remains to take a polynomial M that dominates the function between parentheses. Then $|P(x_1,\dots,x_k)| < 1$ and since this polynomial has integer coefficients, it takes integer values at integer arguments, and hence must be 0 in $x_1,\dots,x_k$. In other words, given P, determining a real root to $M^2\Phi \le 1$ is determining an integer solution to P, which is undecidable. □

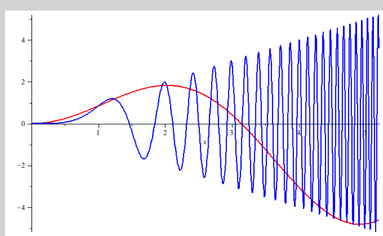
4.3 From Multivariate to Univariate

Consider

$$e_1(x) = x\sin(x), \qquad h(x) = x\sin(x^3), \qquad e_{i+1}(x) = e_i(h(x)) \quad (i > 0).$$

Lemma 4.4 *For any $(x_1, x_2) \in \mathbb{R}^2$ and any $\epsilon > 0$, there exists a real y such that*

$$h(y) = x_2, \quad \left|e_1(y) - x_1\right| < \epsilon.$$



Proof: In any interval $[2k\pi - \pi/2, 2k\pi + \pi/2]$ with $k \in \mathbb{N}$ function sin takes values in all $[-1,1]$. Function e_1 is continuous and takes values in all $[-2k\pi, 2k\pi]$. Hence, for all k such that $2k\pi > |x_1|$, this interval contains some y_k with $e_1(y_k) = x_1$. Moreover, still by continuity, there exists η_k with $|x - y_k| < \eta_k$ implies $|e_1(x) - x_1| < \epsilon$. The derivative of e_1 can be bounded by $|e_1'(x)| = |x\cos(x) + \sin(x)| < (2k+1)\pi$, so that $\eta_k = \epsilon/((2k+1)\pi)$ is sufficient. In the same interval, function x^3 has an amplitude bigger than 2π for k sufficiently large:

$$(y_k + \eta_k)^3 - (y_k - \eta_k)^3 = 6y_k^2\eta_k + 2\eta_k^3 > 6y_k^2\eta_k \ge \frac{6(2k-1/2)^2\pi^2\epsilon}{(2k+1)\pi}.$$

Hence, for k sufficiently large, function h takes values in all interval $[-2k\pi + \pi/2, 2k\pi - \pi/2]$. Increasing k if necessary, x_2 belongs to this intervals. This concludes. □

Lemma 4.5 *For all* $(x_1, \ldots, x_k) \in \mathbb{R}^k$ *and for all* $\epsilon > 0$, *there exists some real* y *such that*

$$h(y) = x_k, \qquad |e_i(y) - x_i| < \epsilon \quad (1 \le i < k).$$

Proof: For $k = 2$, the result is previous lemma. If the property holds for k, there exists $y^\star$ such that

$$h(y^\star) = x_{k+1}, \qquad |e_i(y^\star) - x_{i+1}| < \epsilon \quad (1 \le i < k).$$

By previous lemma, there exists y such that $y^\star = h(y)$ and $|e_1(y) - x_1| < \epsilon$. This proves the property for $k+1$, by the definition of functions e_i. □

Theorem 4.2 *There does not exist an algorithm that takes as input a function of one variable* $f(x) \in \Sigma_1$, *and that decides if there is a real solution to inequality* $f(x) < 0$.

Solution 4.1 *We start from inequality* $M^2(x_1, \ldots, x_k)\Phi(x, y, z, x_1, \ldots, x_k) < 1$ *as above, where we replace* $(x_1, \ldots, x_k, x, y, z)$ *by expressions* $e_1(y), \ldots, e_{k+3}(y)$ *from previous Lemma. This gives an inequality of the form* $\Psi(y) - 1 < 0$ *with* $\Psi - 1 \in \Sigma_1$. *By continuity, this inequality has a real solution iff the inequality of previous Lemma. This concludes.*

Theorem 4.3 *There does not exist an algorithm that takes as input a function of one variable* $f(x) \in \Sigma_1$, *and that decides if there exists a real solution to equation* $f(x) = 0$.

Proof: If polynomial P has a integer root, then function Ψ of previous theorem takes positive values arbitrarily close to 0. It takes also arbitrarily large values (consider for example z to be very large). By continuity, there exists a solution to equation $2\Psi(y) - 1 = 0$ iff P has integer solutions. □

4.4 Application to Simplification

Theorem 4.4 *There does not exist an algorithm that takes as input a function of one variable* $f(x)$ *built from constant* 1, *addition, subtraction, multiplication, sinus, and absolute value and that decides if this function is equal to the null function.*

Absolute value can be replaced by square root.

Proof: Deciding if there exists $x \in \mathbb{R}$ with $f(x) < 0$ is equivalent to decide if function $|f(x)| - f(x)$ is null over all $\mathbb{R}$. The second part follows from identity $|x| = \sqrt{x^2}$. □

Chapter 5

Diophantine Equations

The purpose of this chapter is to prove Davis-Putnam-Robinson's theorem, and then use Matiyasevich's theorem to prove that any recursively enumerable set is Diophantine.

This chapter is based on [Jones, 1997].

5.1 Preliminaries

A function $f : \mathbb{N}^n \to \mathbb{N}$ is said *exponential polynomial* if it can be written $f(x_1, \ldots, x_n) = t$, where t is either x_i, or N, or $t_1 * t_2$, or $t_1 + t_2$, or $t_1 - t_2$, or ${t_1}^{t_2}$, where $1 \leq i \leq n$, $N \in \mathbb{N}$, and where t_1 and t_2 are in turn exponential polynomial functions of $x_1, \ldots, x_n$.

A exponential polynomial function that can be built without the case ${t_1}^{t_2}$ corresponds to a polynomial function.

For example:

- $f(x, y, z) = 3x + 5y - 71z^5$ is a polynomial function, where of course z^5 is $z * z * z * z * z$, and $3x$ is $x + x + x$.
- $f(p, q, r, n) = (p+1)^{n+3} + (q+1)^{n+3} - (r+1)^{n+3}$ is exponential polynomial.

A set $A \subset \mathbb{N}^n$ is said *exponential Diophantine* (respectively: *Diophantine*) if there exists some integer m and a exponential polynomial function (respectively: polynomial) $f : \mathbb{N}^{n+m} \to \mathbb{N}$ such that

$$(a_1, \ldots, a_n) \in A \text{ if an only if } \exists x_1 \in \mathbb{N}, \ldots, \exists x_m \in \mathbb{N}\ f(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0.$$

For example:

- The set of integers x such that there exist y, z with $3x + 5y - 71z^5 = 0$ is Diophantine.
- The set $\{x, y, z | \exists k \in \mathbb{N}\ x^k + y^k = z^k\}$ is exponential Diophantine.
- For a given (fixed) k, the set $\{x, y, z | x^k + y^k = z^k\}$ is Diophantine.

5.2 Encoding of finite sequences

We will need to encode finite sequences of integers into integers. There are several such techniques available. The best known, first employed by Gödel, uses the Chinese Remainder theorem. In the present setting this technique has the disadvantage tat it makes it rather hard to express certain necessary operations as exponential Diophantine equations. Therefore, we will use another technique invented by Matiyasevich.

We will use the following trick: a sequence $a_0, a_1, \dots, a_n \in \{0,1\}^{n+1}$ can be encoded by integer $\sum_{i=0}^{n} a_i 2^i$.

We will use the following result:

Lemma 5.1 *The set of (k,n,m) such that $m = \binom{n}{k} = \frac{n!}{(n-k)!k!}$ (we fix $\binom{n}{k} = 0$ for $k > n$) is a subset of $\mathbb{N}^3$ that is exponential Diophantine.*

Proof: First, the less-than relation is exponential Diophantine, since

$$a < b \Leftrightarrow \exists x\ a + x + 1 = b.$$

Second, let $[N]_k^B$ te the kth digit of N written in base B. The relation $d = [N]_k^B$ is exponential Diophantine since

$$d = [N]_k^B \Leftrightarrow \exists c, e\ N = cB^{k+1} + dB^k + e \wedge d < B \wedge e < B^k.$$

By the binomial theorem

$$(B+1)^n = \sum_{k=0}^{n} \binom{n}{k} B^k.$$

It follows that $\binom{n}{k}$ is the kth digit of $(B+1)^n$ written in base B, provided $\binom{n}{k} < B$ for all k. This in turn, holds if $B > 2^n$ (left as an exercice). Hence, $m = \binom{n}{k}$ is exponential Diophantine:

$$m = \binom{n}{k} \Leftrightarrow \exists B\ B = 2^n + 1 \wedge m = [(B+1)^n]_k^B.$$

□

Let $a_0, a_1, \dots, a_n \in \{0,1\}^{n+1}$ and $b_0, b_1, \dots, b_n \in \{0,1\}^{n+1}$ two sequences of $n+1$ bits. we consider the encoding a and b of these sequences: we write $a \ll b$ for $\forall 0 \le i \le n,\ a_i \le b_i$.

We will admit the following result:

Lemma 5.2 *$a \ll b$ if and only if $\binom{b}{a} = \frac{b!}{(b-a)!a!}$ is odd.*

Lemma 5.3 *The relation $a \ll b$ (seen as a subset of $\mathbb{N}^2$) is exponential Diophantine.*

Proof: We know that $m = \binom{n}{k}$ is exponential Diophantine by Lemma 5.1. We have m odd if and only if $\exists x\ m = 2x + 1$, hence the result follows after substitution. □

5.3 Davis-Putnam-Robinson's theorem

This section is devoted to prove the following result:

Theorem 5.1 *Any recursively enumerable set $A \subset \mathbb{N}^n$ is exponential Diophantine.*

The idea of the proof is to prove that one can express the execution of a two counters machine with a system of exponential Diophantine equations. Then a system of exponential Diophantine equations is equivalent to a unique exponential Diophantine equation by considering the sum of the square of the equations.

Recall what a two counters machine is: such a machine has two counters x_1 and x_2. Initially, $x_2 = 0$ and $x_1 \in \mathbb{N}$ is the input x. Such a machine has a finite number n of instructions. For each $i \in \{1, \dots, n\}$, instruction i is of the following possible form:

1. $\mathsf{Incr}(c)$ that increments x_c;
2. $\mathsf{Decr}(c)$ that decrements x_c if it is non-null;
3. $\mathsf{IsZero}(c, j)$ tests if x_c is null, go to instruction j if this is true;
4. Halt that halts the program.

It is well known that two counters machines can simulate Turing's machines.
We can even suppose that

1. a two counters machine always halt with all its counter null
    - (if not, one can consider another machine that simulate it, but decrements its counters until there are 0 when it detects that the simulated machine halts, and then halts).
2. a null counter is never decremented: every time an instruction of type $\mathsf{Decr}(c, j)$ is executed, counter x_c has a value ≥ 1.
    - (indeed, we can always replace each instruction $\mathsf{Decr}(c, j)$ by two instructions:
        (a) an instruction that tests whether counter x_c is null, and if this is true, sends to instruction j,
        (b) then instruction $\mathsf{Decr}(c, j)$).

In other words, we get:

Proposition 5.1 *Any recursively enumerable set correspond to the set of integers on which a two counters machine with properties* 1. *and* 2. *above halts.*

To prove Davis-Putnam-Robinson's theorem, we will encode the execution of a two counters machine as a matrix of integers:

Take an example: The machine with the following program.

1. IsZero(1, 4)
2. Decr(1)
3. IsZero(2, 1)
4. Halt

The whole execution of the machine on $x_1 = 2, x_2 = 0$ can be described by the following matrix whose:

- columns correspond to time t (increasing t corresponds to increasing number of column, numbering columns going from right to left);
- and first two rows to values of counters x_1, x_2;
- and since the above program contains 4 instructions (1., 2., 3. et 4.), the following 4 rows are build with 0 and 1, with a 1 if and only if the corresponding instruction is executed.

7	6	5	4	3	2	1	0 = t
0	0	0	1	1	1	2	2 $=x_{1,t}$
0	0	0	0	0	0	0	0 $=x_{2,t}$
0	1	0	0	1	0	0	1 $=i_{1,t}$
0	0	0	1	0	0	1	0 $=i_{2,t}$
0	0	1	0	0	1	0	0 $=i_{3,t}$
1	0	0	0	0	0	0	0 $=i_{4,t}$

Even more precisely: the first two rows represent the value of the counters at time t, when considering that the first step is at time $t = 0$. For example, x_1 has value 2 before step 0 and 1 (that is to say, $x_{1,t} = 2$ for $t = 0$ and $t = 1$). It then has value 1 before step 2 and 3. Etc. The raw i determine which instruction is executed at time t. For example, at time 0, the instruction 1. (that is to say IsZero(1, 4, 2)) is executed, and hence $i_{1,0}$ values 1, and in step 2, instruction 3. (that is to say IsZero(2, 1, 4)) is executed, and hence $i_{3,2} = 1$.

Given a two counters machine with n instructions, the purpose is now to build well-chosen exponential polynomial equations that check whether a matrix with $n+2$ raws represents an execution of the machine.

To do so, we will represent such a matrix by $n+2$ integers $x_1, x_2, i_1, i_2, \ldots, i_n$. These $n+2$ integers will be a solution of the equations if and only if the matrix represents an execution of the machine.

Each of these $n+2$ integers encode a raw of the matrix. For example, the raw of counter x_1, that is to say raw $(x_{1,t})_t$, will be encoded by integer

$$x_1 = \sum_{t=0}^{y} x_{1,t} b^t,$$

where b is an integer greater than all numbers in the matrix, and $y = 7$ is the computation time.

Doing so for each raw, the previous matrix becomes

$$\begin{array}{l}
0*b^7+0*b^6+0*b^5+1*b^4+1*b^3+1*b^2+2*b+2 = x_1\\
0*b^7+0*b^6+0*b^5+0*b^4+0*b^3+0*b^2+0*b+0 = x_2\\
0*b^7+1*b^6+0*b^5+0*b^4+1*b^3+0*b^2+0*b+1 = i_1\\
0*b^7+0*b^6+0*b^5+1*b^4+0*b^3+0*b^2+1*b+0 = i_2\\
0*b^7+0*b^6+1*b^5+0*b^4+0*b^3+1*b^2+0*b+0 = i_3\\
1*b^7+0*b^6+0*b^5+0*b^4+0*b^3+0*b^2+0*b+0 = i_4
\end{array}$$

Doing so, all the matrix can be represented by 6 integers, the integers $x_1, x_2, i_1, i_2, i_3, i_4$. In the general case, if we have 2 counters and n instructions, we need x_1, x_2, et i_1, i_2, $\dots$, i_n, that is to say $n+2$ integers.

For a given machine, we will now produce some exponential Diophantine equations on the variables x (the input), y (the number of steps), x_1, x_2, i_1, $i_2, \cdots$, i_n and b, whose solutions represent the execution of the machine on input x.

First, we choose a basis b sufficiently large: we set the exponential Diophantine equation

$$b = 2^{x+y+n}, \tag{5.1}$$

using the fact that in time y no counter can reach a value greater than $x + y$. Taking such a b guarantees also two useful properties: b is a power of 2, and $b > n$.

We need some integer U whose radix b representation is a list of 1 of length y: one just need to write equation

$$1 + bU = U + b^y : \tag{5.2}$$

indeed, the number $b^{y-1} + b^{y-2} + \cdots + b + 1$ satisfies this equation, and this is the only integers to do so.

We will then express many facts using formulas built from the integers x_i and integers i_j, x, y, b, U and relation $\ll$.

Let $1 \le l \le n$. We write

$$i_l \ll U : \tag{5.3}$$

this imply that all the coefficients of i_l are only 0 and 1's.

We can express the fact that at any time, at most one instruction is executed: one just need to add equation

$$U = \sum_{i=1}^{n} i_l : \tag{5.4}$$

this equation implies that there is exactly one 1 on each column of the i_l (no carry can happen in the sum since $b > n$).

We require all the coefficients to be strictly less than $b/2$: we add

$$x_j \ll (b/2-1)U, \tag{5.5}$$

for $j = 1,2$.

By adding equation

$$1 \ll i_1, \tag{5.6}$$

we guarantee that the first instruction is instruction number 1. By adding equation

$$i_n = b^{y-1}, \tag{5.7}$$

that the last executed instruction is instruction number n (we can suppose without loss of generality that this is the only one containing instruction Halt).

We can also express that after an instruction of type Incr(c) at line l, then instruction at line $l+1$ is executed: one just need to write

$$bi_l \ll i_{l+1} \tag{5.8}$$

for each instruction of number l of type Incr(c): observe how one use the fact that a multiplication by b correspond to a shift to the left.

We can do in a same way for each instruction of type Decr(c):

By adding

$$bi_l \ll i_j + i_{l+1}, \tag{5.9}$$

we express the fact that instruction l of type IsZero(c, j) is followed either by instruction j or instruction $l+1$.

We can then write:

$$x_1 = x + b(x_1 + \sum_{l \in A(1)} i_l - \sum_{l \in S(1)} i_l) \tag{5.10}$$

and

$$x_2 = b(x_j + \sum_{l \in A(2)} i_l - \sum_{l \in S(2)} i_l), \tag{5.11}$$

where $A(j)$is the list of the instructions that increment x_j, and $S(j)$ the list of instructions that decrement x_j: this implies that the counters are updated in a correct way.

It only remains to express that each instruction l of type IsZero(c, j) goes to instruction j if $x_c = 0$, and to instruction $l+1$ otherwise.

This can be expressed by

$$bi_l \ll i_{l+1} + U - 2x_c, \tag{5.12}$$

for each such instruction l.

This is based on the following observation.

- Consider that $x_c = 0$ before and, hence also after, step t, i.e. that

  $$x_c = \cdots + 0 * b^{t+1} + 0 * b^t + \ldots.$$

  Then

  $$2x_c = \cdots + 0 * b^{t+1} + 0 * b^t + \ldots.$$

  (we use here the fact that all coefficients are less than $b/2$, and hence no coefficient of b^{t-1} is shifted in the coefficient of b^t by the multiplication by 2).

  In that case,

  $$U - 2x_c = \cdots + 1 * b^{t+1} + \ldots.$$

  I.e.: the coefficient at b^{t+1} of $U - 2x_c$ is odd.

- Consider that $x_c = v > 0$ before, and hence also after, step t, then

  $$x_c = \cdots + v * b^{t+1} + v * b^t + \ldots.$$

  Then

  $$2x_c = \cdots + 2v * b^{t+1} + 2v * b^t + \ldots.$$

  (we use here again the fact that all coefficients are less than $b/2$, and hence that no coefficient is shifted).

  In that case,

  $$U - 2x_c = \cdots + (b - 2v) * b^{t+1} + \ldots.$$

  I.e: the coefficient at b^{t+1} of $U - 2x_c$ is even.

We have

$$bi_l = \cdots + 1 * b^{t+1} + \ldots.$$

This means, that Equation (5.12) to hold, we must have that coefficient at b^{t+1} at i_{l+1} is 0 whenever $U - 2x_c$ is odd, and 1 whenever $U - 2x_c$ is even (because $1 \ll 0 + 0$ is wrong, and $1 \ll 1 + 1$ is wrong).

That is to say, next instruction is instruction number $l + 1$ iff $x_c = v > 0$.

We then obtain the proof of Davis-Putnam-Robinson's theorem: Any recursively enumerable subset $A \subset \mathbb{N}^n$ is exponential Diophantine.

Indeed, let $A \subset \mathbb{N}^n$ be a recursively enumerable set. It corresponds to the set of n-tuples on which a two counters machine halts. By all previous considerations, there exists a system of exponential Diophantine equations such that the n-tuples on which the machine halts are exactly the solutions of the system.

Write $f_1(x_1, \ldots, x_n) = 0$, $f_2(x_1, \ldots, x_n) = 0$, $\ldots$, $f_k(x_1, \ldots, x_n) = 0$ this system of equations. We can then consider

$$f(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_n)^2 + \cdots + f_k(x_1, \ldots, x_n)^2.$$

Then, using that a sum of square is null iff each term is null, $f(x_1, \ldots, x_n) = 0$ is a unique equation whose solutions are the solutions of the system: its solutions are exactly the n-tuples on which the machine halts.

5.4 Matiyasevich's theorem

We will admit the following result, due to Matiyasevich:

Theorem 5.2 (Matiyasevich) *The set of integers u, v, w such that $u = v^w$ is Diophantine.*

We get:

Corollary 5.1 *Any recursively enumerable set is Diophantine.*

Proof: By Davis-Putnam-Robinson's theorem, any recursively enumerable set $A \subset \mathbb{N}^n$ is exponential Diophantine: one can build $f(x, z_1, \ldots, z_n) = 0$ such that $x \in A$ iff $\exists z_1, \ldots, \exists z_n\ f(x, z_1, \ldots, z_n) = 0$.

By Matiyasevich's theorem, there is a Diophantine equation

$$e(u, v, w, y_1, \ldots, y_m) = 0$$

such that $u = v^w$ iff $\exists y_1, \ldots, \exists y_m\ e(x, y_1, \ldots, y_n) = 0$.

Replace any occurrence in $f(x, z_1, \ldots, z_n)$ of $t_1^{t_2}$ by a new variable u. Add to original equation $f(x, z_1, \ldots, z_n) = 0$ the equations $v = t_1$, $w = t_2$ and $e(u, t_1, t_2, y_1, \ldots, y_m) = 0$. All these equations can be combined into a unique diophantine equation by considering that the sum of the square of the equations must be 0. □

5.5 On the impossibility of solving Diophantine equations

Corollary 5.2 *There is no algorithm that can decide for a Diophantine equation whether or not it has a solution in natural numbers.*

Proof: Let $A \subset \mathbb{N}$ be a recursively enumerable, non-recursive set. By above theorem, there exists an Diophantine equation $f(x, z_1, \cdots, z_n) = 0$ such that $x \in A$ iff $f(x, z_1, \cdots, z_n) = 0$ has a solution. Since we can construct effectively the equation $f(x, z_1, \cdots, z_n) = 0$ given x, it follows that an algorithm to decide for each x whether $f(x, z_1, \cdots, z_n) = 0$ has a solution would imply a decision procedure for A, which is impossible since A is non-recursive. □

Chapter 6

Basics about dynamical systems

Let's consider that we are working in $\mathbb{R}^n$ (in general, we could consider any vector space with a norm). Let us consider $f : E \to \mathbb{R}^n$, where $E \subset \mathbb{R}^n$ is open.

6.1 Ordinary Differential Equations

An Ordinary Differential Equation (ODE) is given by $y' = f(y)$ and its solution is a differentiable function $y : I \subset \mathbb{R} \to E$ that satisfies the equation.

For any $x \in E$, the fundamental existence-uniqueness theorem (see e.g. [Hirsch et al., 2003]) for differential equations states that if f is Lipschitz on E, i.e. if there exists K such that $||f(y_1) - f(y_2)|| < k||y_1 - y_2||$ for all $y_1, y_2 \in E$, then the solution of

$$y' = f(y), \qquad y(t_0) = x \tag{6.1}$$

exists and is unique on a certain maximal interval of existence $I \subset \mathbb{R}$. In the terminology of dynamical systems, $y(t)$ is referred to as the *trajectory*, $\mathbb{R}^n$ as the *phase space*, and the function $\phi(t, x)$, which gives the position $y(t)$ of the solution at time t with initial condition x, as the *flow*. The graph of y in $\mathbb{R}^n$ is called the *orbit*.

In particular, if f is continuously differentiable on E then the existence-uniqueness condition is fulfilled [Hirsch et al., 2003]. Most of the mathematical theory has been developed in this case, but can be extended to weaker conditions. In particular, if f is assumed to be only continuous, then uniqueness is lost, but existence is guaranteed: see for example[Coddington and Levinson, 1972]. If f is allowed to be discontinuous, then the definition of solution needs to be refined. This is explored by Filippov in [Filippov, 1988]. Some hybrid system models use distinct and ad hoc notions of solutions. For example, a solution of a piecewise constant differential equation in [Asarin et al., 1995] is a continuous function whose right derivative satisfies the equation.

6.2 Dynamical Systems

In general, a dynamical system can be defined as the action of a subgroup $\mathcal{T}$ of $\mathbb{R}$ on a space X, i.e. by a function (a flow) $\phi : \mathcal{T} \times X \to X$ satisfying the following two equations

$$\phi(0, x) = x \tag{6.2}$$

$$\phi(t, \phi(s, x)) = \phi(t + s, x). \tag{6.3}$$

It is well known that subgroups $\mathcal{T}$ of $\mathbb{R}$ are either dense in $\mathbb{R}$ or isomorphic to the integers. In the first case, the time is called continuous, in the latter case, discrete.

6.2.1 Continuous Time Dynamical Systems

Since flows obtained by initial value problems (IVP) of the form (6.1) satisfy equations (6.2) and (6.3), they correspond to specific continuous time and space dynamical systems. Although not all continuous time and space dynamical systems can be put in a form of a differential equation, IVPs of the form (6.1) are sufficiently general to cover a very wide class of such systems. In particular, if ϕ is continuously differentiable, then $y' = f(y)$, with $f(y) = \frac{d}{dt}\phi(t, y)\Big|_{t=0}$, describes the dynamical system.

6.2.2 Discrete Time Dynamical Systems

For discrete time systems, we can assume without loss of generality that $\mathcal{T}$ is the integers. The analog of of Initial Value Problem (6.1) for discrete time systems is a recurrence equation of type

$$y_{t+1} = f(y_t), \qquad y_0 = x. \tag{6.4}$$

Chapter 7

Dynamic Undecidability

Chapter 8

Static vs Dynamic Undecidability

The previous results are *static undecidability* results. They don't really say things about the hardness of

- simulating dynamical systems;
- verifying dynamical systems.

8.1 A provocative point of view

As observed in [Asarin, 1995] and in [Ruohonen, 1997], it is relatively simple but not very informative to get undecidability results with continuous time dynamical systems, if f encodes a undecidable problem.

To illustrate this, we consider the following example taken from [Ruohonen, 1997]. In this paper, Ruohonen discusses the event detection problem: given a differential equation $y' = f(t, y)$, with initial value $y(0)$, decide if a given condition $g_j(t, y(t), y'(t)) = 0$, $j = 1, \cdots, k$ happens at some time t in a given interval I.

Given the Turing machine $\mathscr{M}$, the sequence $f_0, f_1, \cdots$ of rationals defined by

$$f_n = \begin{cases} 2^{-m} & \text{if} \quad \mathscr{M} \text{ stops in } m \text{ steps on input } n \\ 0 & \text{if} \quad \mathscr{M} \text{ does not stop on input } n \end{cases}$$

is not a computable sequence of rationals, but is a computable sequence of reals.

Now, the detection of the event $y(t) = 0$ for the ordinary differential equation $y' = 0$, given n, and the initial value $y(0) = f_n$, is undecidable over any interval containing 0, because $f_n = 0$ is undecidable.

A further modification can be obtained as follows [Ruohonen, 1997].

Consider the smooth function

$$g(x) = f_{\lfloor x+1/2 \rfloor} e^{-\tan^2 \pi x},$$

which is computable on $[0,\infty)$. The detection of the event $y_1(t) = 0$ for the ODE

$$\begin{cases} y_1' &= g(y_2) - 1 \\ y_2' &= 0 \end{cases}$$

given an initial value $y_1(0) = 1$, $y_2(0) = n$, where n is a nonnegative integer is then undecidable on $[0,1]$.

As put forth in [Asarin, 1995] undecidability results given by recursive analysis are somehow built similarly.

8.2 Dynamic undecidability

To be able to discuss in more detail computability of differential equations, we will focus on dynamical systems that encode the transitions of a Turing machine instead of the result of the whole computation simulation[1]. Typically, we start with some (simple) computable injective function which encodes any configuration of a Turing machine M as a point in $\mathbb{R}^n$. Let x be the encoding of the initial configuration of $\mathcal{M}$. Then, we look for a function $f : E \subset \mathbb{R}^{n+1} \to \mathbb{R}^n$ such that

- Discrete Time Case: the solution of $y^{t+1} = f(y,t)$ with $y(0) = x$, is such that x_t is the encoding of the configuration of $\mathcal{M}$ after t steps.
- Continuous time Case: the solution of $y'(t) = f(y,t)$, with $y(0) = x$, at time $T \in N$ is the encoding of the configuration of $\mathcal{M}$ after T steps.

We will see, in the remainder of this section, that f can be restricted to have low dimension, to be smooth or even analytic, or to be defined on a compact domain.

Instead of stating that the property above is a Turing machine simulation, we can address it as a reachability result. Given the IVP defined by f and x, and any region $A \subset \mathbb{R}^n$, we are interested in deciding if there is a $t \geq 0$ such $y(t) \in A$, i.e., if the flow starting in x crosses A. It is clear that if f simulates a Turing machine in the previous sense, then reachability for that system is undecidable (just consider A as encoding the halting configurations of $\mathcal{M}$). So, reachability is another way to address the computability of ODEs and a negative result is often a byproduct of the simulation of Turing machines. Similarly, undecidability of event detection follows from Turing simulation results.

[1] This is called dynamic undecidability in [Ruohonen, 1993].

Chapter 9

Some Dynamic Undecidability Results: Using a Discrete Time

9.1 Some models

9.1.1 The PAM Model

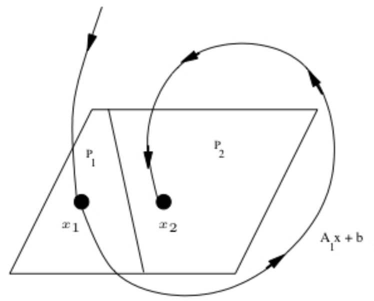
Here is a simple toy model.

Definition 9.1 (PAM [Asarin and Maler, 1998]) *A Piecewise Affine Map (PAM) is a discrete time dynamical system $\mathcal{H}$, defined by $\mathbf{x}_{t+1} = f(\mathbf{x}_t)$ on $X \subset \mathbb{R}^d$, where $f : X \to \mathbb{R}^d$ is piecewise affine: that is to say,*

$$f(\mathbf{x}) = f_i(\mathbf{x}) \text{ for } \mathbf{x} \in P_i,\ i = 1, \dots, n$$

where f_i is some affine function with rational coefficients, and the P_i constitutes a partition of X into finitely many rational convex polyhedra.

Recall that a convex polyhedra is the convex hull of a finite number of points. A rational convex polyhedra is the convex hull of a finite number of points with rational coordinates.



9.1.2 The PCD Model

We are going to discuss the Piecewise Constant Derivative (PCD) model that has been introduced by Eugene Asarin, Oded Maler and Amir Pnueli in [Asarin et al., 1995], as a simple model for hybrid systems. It has later on been discussed in several papers such as [Asarin and Bouajjani, 2001, Asarin and Maler, 1998, Bournez, 1999].

A hybrid system is a system that combines continuous evolutions with discrete transitions. Such models appear as soon as one tries to model some systems where a discrete system, such as a computer, evolves in a continuous environment: See e.g. [Antsaklis, 2000].

From a theoretical computer science point of view, one interest of the hybrid systems models, is that they generalize both discrete time transition systems and continuous time dynamical systems.

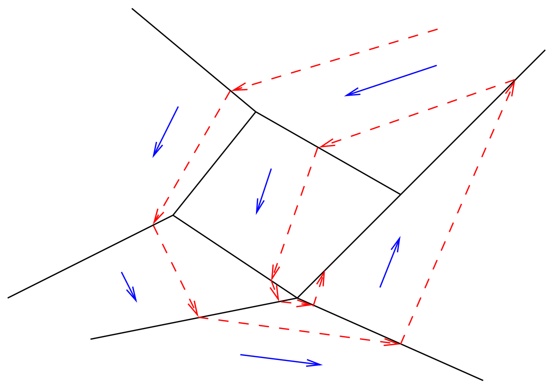
Definition 9.2 (PCD System [Asarin and Maler, 1998]) *A (rational) piecewise-constant derivative (PCD) system is a continuous time dynamical system $\mathcal{H}$, defined by differential equation $\dot{\mathbf{x}} = f(\mathbf{x})$ on $X \subset \mathbb{R}^d$, where $f : X \to \mathbb{R}^d$, can be represented by the formula*

$$f(\mathbf{x}) = \mathbf{c}_i \text{ for } \mathbf{x} \in P_i,\ i = 1, \ldots, n$$

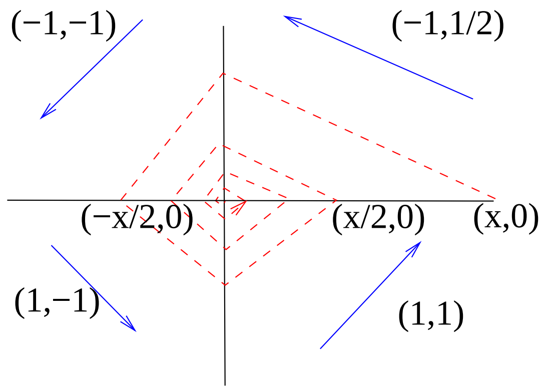
where $c_i \in \mathbb{Q}^d$, and the P_i constitutes a finitely many partition of X into rational convex polyhedra.

A trajectory of $\mathcal{H}$ starting from some $\mathbf{x}_0 \in X$, is a solution of the differential equation $\dot{\mathbf{x}} = f(\mathbf{x})$ with initial condition $\mathbf{x}(0) = \mathbf{x}_0$: that is a continuous function $\phi : \mathbb{R}^+ \to X$ such that $\phi(0) = \mathbf{x}_0$, and for every t, $f(\phi(t))$ is equal to the *right derivative* of $\phi(t)$.

In other words, a PCD system consists of partitioning the space into convex polyhedral sets ("regions"), and assigning a constant derivative $\mathbf{c}$ ("slope") to all the points sharing the same region. The trajectories of such systems are broken lines, with the breakpoints occurring on the boundaries of the regions [Asarin et al., 1995]: see the following figure.

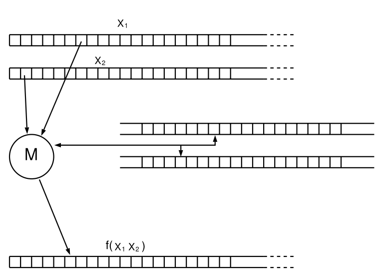


Here is an example of a trajectory of a PCD system:



9.2 The most fundamental model: Turing Machines

Turing machines can also be considered as particular discrete time dynamical systems.



A Turing machine is indeed a discrete time dynamical system $(\Gamma, \vdash)$, where

- $\Gamma = Q \times \Sigma^* \times \mathbb{Z}$ corresponds to configurations (a configuration (q, w, z) is given by some internal state $q \in Q$ of the machine, some position $z \in \mathbb{Z}$ of the head of the machine, and the content $w \in \Sigma^*$, that can be seen as a word over the alphabet Σ of the machine, of the tape).

- $\vdash$ is the "next configuration relation": it relates a configuration to its direct successor (when the machine is deterministic, or to its direct successors when the machine is non-deterministic).

9.3 Some Facts

9.3.1 A Key Decision Problem

The reachability problem is the following decision problem.

- Given

  1. a system $\mathcal{H} = (X, f)$,
  2. some $A \subset X$,
  3. some $B \subset X$,

- determine whether there is a trajectory starting from A ($x(0) \in A$) that reaches B ($x(t) \in B$ for some t).

9.3.2 Some Undecidability Results

- For all 4 models,

  – Reachability is undecidable;
  – Reachability is recursively enumerable;
  – Reachability is Σ_1^0-complete: any r.e. set can be reduced to reachability of a system S.

9.4 Proof method

The idea is to simulate 2-counters (Minsky) machines, or Turing machines.

9.4.1 The involved notion of simulation

Consider some machine M: M can be a Turing machine, a pushdown automaton, or a counter machine. M corresponds to a particular discrete time dynamical system $(\Gamma, \vdash)$.

By a discrete time dynamical system

A PAM simulates M if there is a piecewise affine function $f : I \to I$, where $I \subset \mathbb{R}^d$, and a f-stable $D \subset I$ and a bijective function $\phi : \Gamma \to D$ such that

$$\vdash = \phi^{-1} \circ f \circ \phi.$$

Intuitively, this means that in order to apply T, one can encode the configuration with ϕ, apply f, and then decode the result with ϕ^{-1}.

By a continuous time dynamical system

To a continuous time dynamical system (X, f), one can associate its *stroboscopic map*: this is the discrete time dynamical system (X, g), where $g(x)$ is the solution at time 1 of $x' = f(x)$ with $x(0)) = x$. That is to say, what is obtained by considering the system at discrete time.

We can say that a continuous time dynamical system simulates M if its stroboscopic map simulates M.

9.4.2 Counter machines

We recall that a k-counter machine has k counters: a counter C takes values in $\mathbb{N}$ with operations $C++$ (incrementation), $C--$ (decrementation), test $C > 0$.

A program is then made of these very basic instructions.

For example,

q_1: $D++$; goto q_2

q_2: $C--$; goto q_3

q_3: if $C > 0$ then goto q_2 else goto q_1

is a program.

Recall that reachability is undecidable (and Σ_1^0-complete) for 2-counters (Minsky) machines.

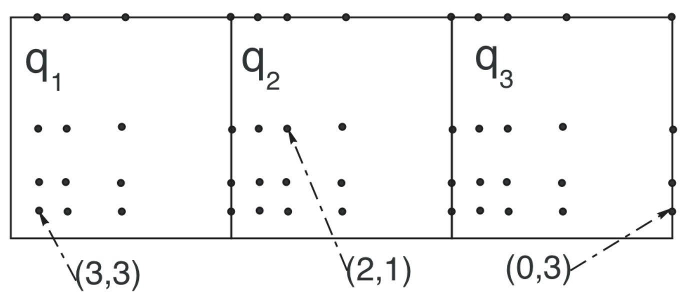
The idea to simulate a counter is to represent the fact that $C = n$ in the counter machine by the fact that some variable x is such that $x = 2^{-n}$.

Basically, this can be represented graphically by:

One then uses the following correspondance:

Counter	PAM
State Space $\mathbb{N}$	State Space $[0,1]$
State $C = n$	State $x = 2^{-n}$
$C{+}{+}$	$x := x/2$
$C{-}{-}$	$x := 2x$
$C > 0?$	$x < 0.75?$

To represent a Minsky machine, one then encodes the two counters, and its states into two reals using the following idea:



Minsky Machine	PAM
State Space $\{q_1, q_2, \cdots, q_k\} \times \mathbb{N} \times \mathbb{N}$	State Space $[1, k+1] \times [0,1]$
State (q_i, m, n)	State $x = i + 2^{-m}$, $y = 2^{-n}$

If one prefers, the function ϕ of subsection 9.4.1 is the function that maps $(q, n_1, n_2) \in Q \times \mathbb{N} \times \mathbb{N}$ to $(q + 2^{-n_1}, 2^{-n_2})$.

On the previous example, one just need to consider the following PAM (the value of the piecewise affine function can be defined in any arbitrary way outside the above sets/definition).

Minsky Machine	PAM
State Space $\{q_1, \cdots, q_k\} \times \mathbb{N} \times \mathbb{N}$	State Space $[1, k+1] \times [0,1]$
State (q_i, m, n)	$x = i + 2^{-m}$, $y = 2^{-n}$
q_1: $D{+}{+}$; goto q_2	$\begin{cases} x := x+1 \\ y := y/2 \end{cases}$ if $1 < x \le 2$
q_2: $C{-}{-}$; goto q_3	$\begin{cases} x := 2(x-2)+3 \\ y := y \end{cases}$ if $2 < x \le 3$
q_3: if $C > 0$ then goto q_2 else q_1	$\begin{cases} x := x-1 \\ y := y \end{cases}$ if $3 < x < 4$ $\begin{cases} x := x-2 \\ y := y \end{cases}$ if $x = 4$

9.4.3 From Minsky to Turing Machines

This is even possible to do a real time simulation of a 2-stacks (that is to say a Turing) machines.

A stack $S = s_1 s_2 \cdots$ over alphabet $\{0,1,2,\cdots,k-1\}$, where s_i is the top of the stack, can be encoded in several ways:

1. Idea 1:

   by
$$r(S) = \sum_{i=1}^{\infty} \frac{s_i}{k^i}$$

2. Improved idea $1'$:
$$r(S) = \sum_{i=1}^{\infty} \frac{2s_i+1}{(2k)^i}$$

For both encoding, stack operations have then arithmetic counterparts:
For the coding corresponding to Idea 1:

$$S' = PUSH(v,S) \text{ if } r(S') = (r(s)+v)/k$$

$$(S',v) = POP(S) \text{ if } r(S') = kr(S) - v$$

As a Turing machine can be considered as a 2-stacks machine: We can encode easily a Turing machine using a PAM $(\mathbb{R}^2, f)$.

The interest of "Improved Idea $1'$" is that it allows to state that we can even assume the piecewise affine map f to be

1. a *continuous* function.

2. and one can even consider domain $[0,1]^2$, instead of $\mathbb{R}^2$.

Theorem 9.1 (Theorem 3.1 of [Koiran et al., 1994]) *An arbitrary Turing machine can be simulated in linear time by a* continuous *piecewise linear function* $f : [0,1]^2 \to [0,1]^2$.

Proof: Any Turing machine can be considered as a particular 2-stacks automaton, i.e. as a discrete time dynamical system $M = (Q \times \Sigma^* \times \Sigma^*, \vdash)$: (q,γ_1,γ_2) corresponds to internal state q, and to stacks γ_1 and γ_2 seen as words over the alphabet Σ. In order to simplify the description, we suppose wlog in what follows that $Q = \{1,3\}^{p_1} \times \{1,3\}^{p_2}$ (you can assume $p_2 = 0$ but this form is more symmetric) and that $\Sigma = \{1,3\}$.

Each configuration (q,γ_1,γ_2) of M is encoded in the radix-4 expansion of a point (x_1,x_2) of $[0,1]^2$ as follows: if $q = (q_{1,1}, q_{1,2}, \ldots, q_{1,p_1}, q_{2,1}, q_{2,2}, \ldots, q_{2,p_2}) \in Q = \{1,3\}^{p_1} \times \{1,3\}^{p_2}$ and $\gamma_i = s_{i,1}, s_{i,2}, \ldots, s_{i,n}, \ldots$, then

$$x_i = \sum_{j=1}^{p_i} \frac{q_{i,j}}{4^j} + \sum_{j=1}^{\infty} \frac{s_{i,j}}{4^{p_i+j}}$$

We will denote $\overline{abc}$ the real number with radix-4 expansion abc.
Let $I_{1,l_1} \times I_{2,l_2}$ be all the sets defined by:

- $I_{i,l_i} = [l_i, l_i + 1/4^{p_i+p}]$ and $l_i = \overline{0.q_{i,1}q_{i,2},\dots,q_{i,p_i},s_{i,1}}$
- or $I_{i,l_i} = \{l_i\}$ and $l_i = \overline{0.q_{i,1}q_{i,2},\dots,q_{i,p_i}}$

for any $s_{i,1}$ and $q_{i,j}$ elements of $\{1,3\}$.

The stack is nonempty in the first case, and empty in the second one. In what follows, we will not make any more this distinction, and we will suppose, in the case of an empty stack, that $s_{i,1}, s_{i,2},\dots,s_{i,p} = 0$.

Assume that $(x_1, x_2) \in I_{1,l_1} \times I_{2,l_2}$ encodes the configuration $(q, a_1\gamma_1, a_2\gamma_2)$ of M at time t, where $a_1, a_2 \in \Sigma$, $\gamma_1, \gamma_2 \in \Sigma^*$ and

$$q = (q_1, q_2) = (q_{1,1},\dots,q_{1,p_1}, q_{2,1},\dots,q_{2,p_2}) \in Q.$$

Call $\Delta x_i = x_i - l_i$, for $i \in \{1,2\}$.

On $I_{1,l_1} \times I_{2,l_2}$, we define f such that $f(x_1, x_2) = (x'_1, x'_2)$ with

$$x'_i = \overline{0.q'_{i,1},\dots,q'_{i,p_i}} + \Delta x'_i$$

where

$$q' = (q'_1, q'_2) = (q'_{1,1}, q'_{1,2}\dots, q'_{1,p_1}, q'_{2,1}, q'_{2,2},\dots, q'_{2,p_2})$$

is the next internal state of M (fully determined by the current state q and the top-of-stack letters a_1 and a_2), and $\Delta x'_i$ defined by:

- $\Delta x'_i = 4\Delta x_i$ if stack i is popped,
- $\Delta x'_i = \frac{s_{i,1}}{4^{p_i+1}} + \Delta x_i$ if stack i is unchanged
- $\Delta x'_i = \frac{a_i}{4^{p_i+1}} + \frac{s_{i,1}}{4^{p_i+2}} + \frac{\Delta x_i}{4}$ if a_i is pushed on stack i

It can be checked that, in any case, f is built such that $f(x_1, x_2)$ encodes configuration M at time $t+1$ whenever (x_1, x_2) encodes configuration M at time t.

f is piecewise linear, as it is defined as linear on each of the products $I_{1,l_1} \times I_{2,l_2}$.

In order to complete the proof, we have to define f outside

$$C = \bigcup_{l_1,l_2} I_{1,l_1} \times I_{2,l_2},$$

to the whole of $[0,1]^2$: this extension cannot interfere with the simulation of M since only points of C are used in a computation. There are continuous piecewise linear extensions of f since the distance between two distinct products is greater than 0. As a matter of fact, the suppremum distance is bounded below by $min(1/4^{p_1+1}, 1/4^{p_2+1}$.
□

Remark 9.1 *Observe that the proofs shows that one need very simple piecewise affine functions to be able to simulate a Turing machine: basically, we only need to be able to do additions, multiplications by* 4 *and divisions by* 4*, on appropriate pieces.*

Remark 9.2 *Observe that instead of extending the function outside C piecewise linearly, we can even extend it in a $\mathcal{C}^\infty$ way. Hence, an arbitrary Turing machine can be simulated in linear time by a $mathcalC^\infty$ function $f : [0,1]^2 \to [0,1]^2$.*

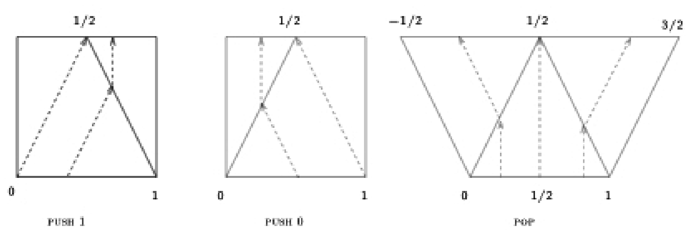
It is believe that however this is not possible using an analytic function over a compact domain.

9.4.4 Using PCDs

Suppose we want to go further and simulate a Turing Machine with PCDs.

The first step is to see that one can do basic operations with PCDS.

For the encoding corresponding to "Idea 1", assuming that the alphabet is $\Sigma = \{0,1\}$, one just need to consider the following basic PCDs.

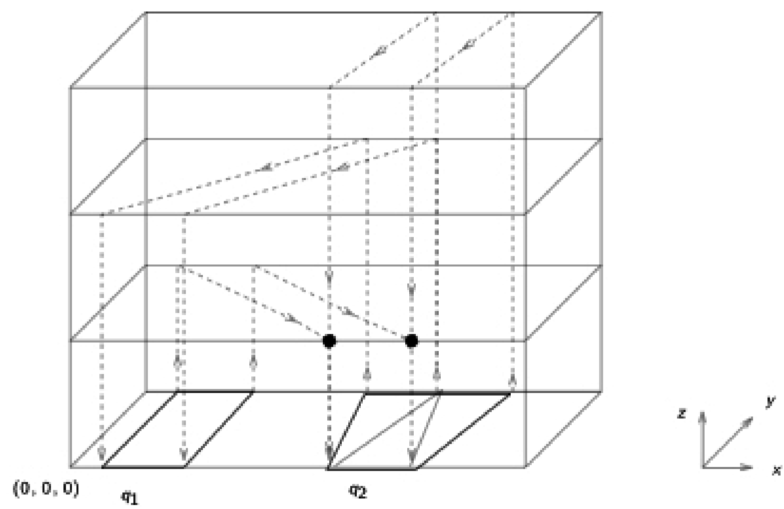


Of course, this is easy to generalize the idea to build basic blocks for the encoding to "Idea 1′": basically, one just need to do multiplications and divisions by 4 instead of 2, which can be done using the same principles.

With these blocks, this is easy to get a PCD that simulates any Push-Down automaton with a PCD.

For example, for the Push-Down automaton

q_1: $S := PUSH(1,S)$; goto q_2
q_2: $(v,S) := POP(S)$; if $v = 1$ then goto q_2 else q_1,
we just need to build a PCD like this:



We get:

Theorem 9.2 (Asarin-Maler-Pnueli 94) *Every Turing machine can be simulated by a* 4*-dimensional PCD system.*

Playing a little bit with the construction trying to reduce the dimension, one can be easily convinced that dimension 3 is enough:

Theorem 9.3 (Asarin-Maler-Pnueli 94) *Every Turing machine can be simulated by a* 3*-dimensional PCD system.*

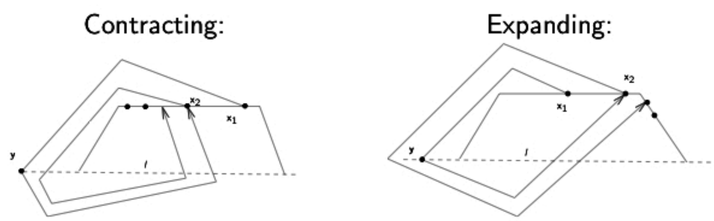
However, this is not possible in dimension 2:

Theorem 9.4 (Asarin-Maler-Pnueli 94) *But not by a* 2*-dimensional PCD system.*

Indeed:

Theorem 9.5 (Asarin-Maler-Pnueli 94) *Reachability for planar* 2*-dimensional PCD systems is decidable.*

The main ingredient of the proof: Jordan's theorem: All repetitive behaviors are either contracting or expanding spirals:



9.5 Extensions

Turing machines can be embedded into analog space discrete time systems with low dimensional systems with other simple dynamics: [Moore, 1990], [Ruohonen, 1993], [Branicky, 1995], [Ruohonen, 1997] consider general dynamical systems, [Koiran et al., 1994] piecewise affine maps, [Siegelmann and Sontag, 1995] sigmoidal neural nets, [Siegelmann and Sontag, 1995], closed form analytic maps, which can be extended to be robust [Graça et al., 2005a], and [Kurganskyy and Potapov, 2005] one dimensional very restricted piecewise defined maps.

Chapter 10

Some Dynamic Undecidability Results: Using a Continuous Time with Smooth Dynamics

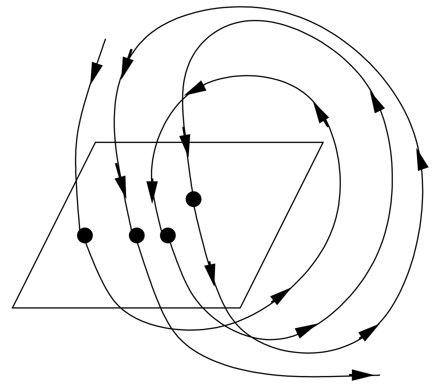
The embedding of Turing machines in continuous dynamical systems is often realized in two steps. Turing machines are first embedded into analog space discrete time systems, and then the obtained systems are in turn embedded into analog space and time systems.

We saw the first step in previous chapter.

For the second step, the most common technique is to build a continuous time and space system whose discretization corresponds to the embedded analog space discrete time system.

We saw an example in previous chapter with PCDs. However, in PCD the dynamic is non-smooth: it is non-continuous. We discuss here what can be said for smooth dynamics (at least continuous).

In a general setting, there are several classical ways to discretize a continuous time and space system: One way is to use a virtual stroboscope: the flow $x_t = \phi(t, x)$, when t is restricted to integers, defines the trajectories of a discrete time dynamical system. Another possibility is through a PoincarÃ© section: the sequence x_t of the intersections of trajectories with, for example, a hypersurface can provide the flow of a discrete time dynamical system. See [Hirsch et al., 2003].



The opposite operation, called *suspension*, is usually achieved by extending and smoothing equations, and usually requires higher dimensional systems. This explains why Turing machines are simulated by three-dimensional smooth continuous time systems in [Moore, 1990], [Moore, 1991], [Branicky, 1995] or by three-dimensional piecewise constant differential equations in [Asarin et al., 1995], while they are known to be simulated in discrete time by only two-dimensional piecewise affine maps in [Koiran et al., 1994]. It is known that two-dimensional piecewise constant differential equations cannot[1] simulate arbitrary Turing machines [Asarin et al., 1995], while the question whether one-dimensional piecewise affine maps can simulate arbitrary Turing machines is open. Other simulations of Turing machines by continuous time dynamical systems include the robust simulation with polynomial ODEs in [Graça et al., 2005a], [Graça et al., 2007]. This result is an improved version of the simulation of Turing machines with real recursive functions in [Campagnolo et al., 2000a], where it is shown that smooth but non-analytic classes of real recursive functions are closed under iteration. Notice that while the solution of a polynomial ODE is computable on its maximal interval of existence, the simulation result shows that the reachability problem is undecidable for polynomial ODEs.

10.1 Discussion

The key technique in embedding the time evolution of a Turing machine in a flow is to use "continuous clocks" as in [Branicky, 1995].[2]

The idea is to start from the function $f:\mathbb{R}\to\mathbb{R}$, preserving the integers, and build the ordinary differential equation over $\mathbb{R}^3$

$$\begin{aligned} y_1' &= c(f(r(y_2)) - y_1)^3\theta(\sin(2\pi y_3)) \\ y_2' &= c(r(y_1) - y_2)^3\theta(-\sin(2\pi y_3)) \\ y_3' &= 1. \end{aligned}$$

[1] See also already mentioned generalizations of this result in [Ceraens and Viksna, 1996] and [Asarin et al., 2001].

[2] Branicky attributes the idea of a two phase computation to [Brockett, 1989] and [Brockett, 1991]. A similar trick is actually present in [Ruohonen, 1993]. We will actually not follow [Branicky, 1995] but its presentation in [Campagnolo, 2001].

Here $r(x)$ is a rounding-like function that has value n whenever $x \in [n-1/4, n+1/4]$ for some integer n, and $\theta(x)$ is 0 for $x \leq 0$, $\exp(-1/x)$ for $x > 0$, and c is some suitable constant.

The variable $y_3 = t$ is the time variable. Suppose $y_1(0) = y_2(0) = x \in \mathbb{N}$. For $t \in [0, 1/2]$, $y_2' = 0$, and hence y_2 is kept fixed to x. Now, if $f(x) = x$, then y_1 will be kept to x. If $f(x) \neq x$, then $y_1(t)$ will approach $f(x)$ on this time interval, and from the computations in [Campagnolo, 2001], if a large enough number is chosen for c we can be sure that $|y_1(1/2) - f(x)| \leq 1/4$. Consequently, we will have $r(y_1(1/2)) = f(x)$. Now, for $t \in [1/2, 1]$, roles are inverted: $y_1' = 0$, and hence y_1 is kept fixed to the value $f(x)$. On that interval, y_2 approaches $f(x)$, and $r(y_2(1)) = f(x)$. The equation has a similar behavior for all subsequent intervals of the form $[n, n+1/2]$ and $[n+1/2, n+1]$. Hence, at all integer time t, $f^{[t]}(x) = r(y_1(t))$.[3] [Loff et al., 2007] proposes a similar construction that returns $f^{[\lfloor t \rfloor]}(x)$ for all $t \in \mathbb{R}$.

In other words, the construction above transforms a function over $\mathbb{R}$ into a higher dimensional ordinary differential equation that simulates its iterations. To do so, $\theta(\sin(2\pi y_3))$ is used as a kind of clock. Therefore, the construction is essentially "hybrid" since it combines smooth dynamics with non-differentiable, or at least non-analytic clocks to simulate the discrete dynamics of a Turing machine. Even if the flow is smooth (i.e. in C^∞) with respect to time, the orbit does not admit a tangent at every point since y_1 and y_2 are alternatively constant. Arguably, one can overcome this limitation by restricting Turing machine simulations to analytic flows and maps. While it was shown that analytic maps over unbounded domains are able to simulate the transition function of any Turing machine in [Koiran and Moore, 1999], only recently it was shown that Turing machines can be simulated with analytic flows over unbounded domains in [Graça et al., 2005a]. It would be desirable to extend the result to compact domains. However, it is conjectured in [Moore, 1998] that this is not possible, i.e. that no analytic map on a compact finite-dimensional space can simulate a Turing machine through a reasonable input and output encoding.

10.2 Some Dynamic Undecidability Results

We review some dynamic undecidability results obtained in literature (non-exhaustive list).

- [Moore90]: simulation of Turing machine with a $\mathscr{C}^\infty$ discrete-time dynamic over $\mathbb{R}^2$.
- [Ruohonen93]: simulation of a n-counter machine.
- [Asarin-Maler-Pnueli95]: simulation of a Turing machine with a PCD-system over $\mathbb{R}^3$.
- [Branicky95]: simulation of a Turing machine with hybrid systems.
- [Siegelmann95]: simulation of an extended automata with a mirror system.

[3] $f^{[t]}(x)$ denotes de tth iteration of f on x.

- [GraÃ§a-Campagnolo-Buescu2005]: simulation of a Turing machine with a GPAC.
- …

Chapter 11

Space and Time Contraction for Continuous Time Systems

11.1 Considering Dynamical Systems as Language Recognizers

Dynamical systems can be considered as recognizers of languages: let Σ denote alphabet $\{0,1\}$. Σ^* denotes words over this alphabet.

Two (very classical) encodings of words into real numbers will play some important role in what follows:

- ν_X is the function that maps Σ^* to $[0,1]$ as follows: word $w = w_1 \dots w_n \in \{0,1\}^*$ is mapped to $\nu_X(w) = \sum_{i=1}^{n} \frac{(2w_i+1)}{4^i}$.
- $\nu_{\mathbb{N}}$ is the function that maps Σ^* to $\mathbb{N}$ as follows: word $w = w_1 \dots w_n \in \{0,1\}^*$ is mapped to $\nu_{\mathbb{N}}(w) = \sum_{i=1}^{n} (2w_i+1)4^i$.

We can now define.

Definition 11.1 (Dynamical Systems as Language Recognizers) *Let $\mathcal{H}$ be a continuous time or discrete time dynamical system over space X. We will consider two cases: the case $X = [-1,1]^d$ (compact case), or $X = \mathbb{R}^d$ (unrestricted case). Consider $\nu = \nu_X$ for the compact case, $\nu = \nu_{\mathbb{N}}$ for the unrestricted case. Let V_{accept} be the set of $\mathbf{x} \in X$ with $\|\mathbf{x}\| \leq 1/4$. Let $V_{compute}$ be the set of $\mathbf{x} \in X$ with $\|\mathbf{x}\| \geq 1/2$. (or take V_{accept} and $V_{compute}$ to any two disjoint subsets corresponding to a polyhedron with rational coefficients, at a strictly positive distance one from the other).*

We will say that $\mathcal{H}$ computes (or semi-recognize) some language $L \subset \Sigma^*$, over alphabet $\Sigma = \{0,1\}$, if the following holds: for all $w \in \Sigma^*$, $w \in L$ iff the trajectory of $\mathcal{H}$ starting from $(\nu(w), 0, \cdots, 0, 1)$ reaches V_{accept}.

For robustness reasons, we assume that, for any $w \not\in L$, the corresponding trajectory stay forever in $V_{compute}$.

Given some notion of time associated to trajectories, we will say that L is recognized in time T, if furthermore when the trajectory reaches V_{accept}, trajectory has a time bounded above by T. It is said accepted in time $f : \mathbb{N} \to \mathbb{N}$ if furthermore $T \leq f(|w|)$, for all w, where $|w|$ stands for the length of w.

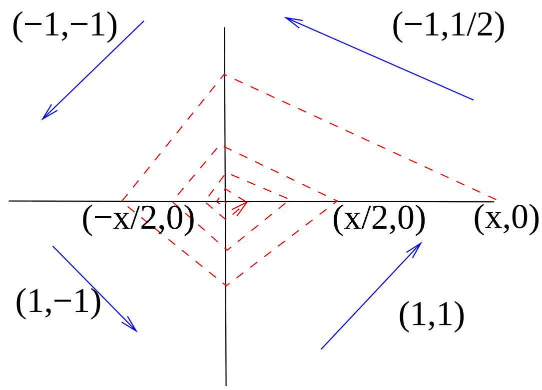
In this chapter, we consider continuous-time dynamical systems, and talk about computability issues. In next chapter, we will focus on complexity.

11.2 Time Contraction for PCD systems

11.2.1 Zeno's Paradox

We come back to PCD systems.

Consider the following PCD:



It takes a time $5x * 1/2$ to go from $(x,0)$ to $(x/2,0)$ that is to say to make a turn of the spiral.

Second turn is made in time $5x * 1/4$. Third in times $5x * 1/8$. And so on.

Considering that

$$5/2(x + x/2 + x/4 + \ldots) = 5x$$

is finite, one can consider that in time $5x$ the trajectory has reached $(0,0$, the limit of the spiral. This happens in finite time, but requires a transfinite number of crossing of regions.

This is called *Zeno's paradox*: to a continuous finite time can correspond a transfinite number of discrete steps.

11.2.2 Using Zeno's Paradox

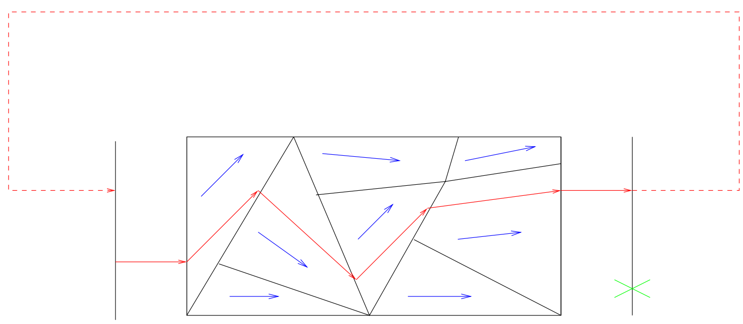
Actually, using the fact that one can simulate a Turing machine M using a PCD system in dimension 3, and this idea, it is possible to build a PCD system of dimension

4 that decides whether M terminates or not, i.e. that solves the halting problem of M.

The idea of the construction is the following.

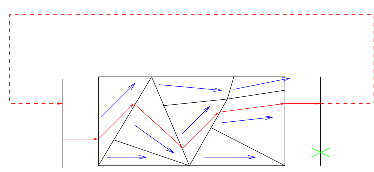
Recall that one can build a PCD system in dimension 3 that simulates a Turing machine M.

Let it correspond to the following (very abstract) picture:



Suppose that you divide all dimensions by 2, but keep speeds unchanged.

You get the following PCD:

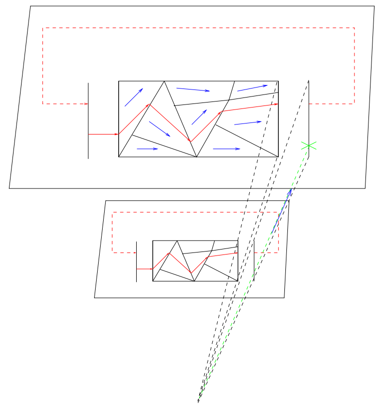


The point is that if it takes a time T for the first one to make a turn, it will make a time $T/2$ for the second to do exactly the same turn.

Of course, instead of dividing everything by 2, you can divide by 4, by 8 and so on.

The trick is that if one put all of this in a 4-dimension space, say x, y, z, u, by putting the first at $u = 1$, the second at $u = 1/2$, the third at $u = 1/4$ and so one, one gets a "pyramid" that corresponds indeed to a PCD.

Graphically:



In the first one, we may assume that there is a particular point that corresponds to the accepting state of Turing machine M.

In the pyramid, it will correspond to an open segment. Assume that we fix that the speed on this segment goes upward.

Assume that we add some regions that maps u to $u/2$ (and all variables divided by 2 also), and that set the system that these regions are crossed at each turn: we will get a PCD that simulates one step of Turing machine M in time 1, another step in time $1/2$, another step in time $1/4$ and so one. As $1+1/2+1/4+1/8+\cdots = 2$, in time 2 either we will reach the accepting segment. Otherwise, we will reach the summit of the pyramid.

In other words, if M accepts we will reach the accepting point at $u = 1$. If M rejects, we will reach the point $(0,0,0,0)$. That is to say, we decide whether M accepts or not in finite time !!

Details can be found in [Asarin and Maler, 1998].

11.2.3 What can be computed?

Hierarchies of undecidable problems

We recall the following definition:

Definition 11.2 (Arithmetical hierarchy [Rogers Jr., 1987, Odifreddi, 1992]) *The classes $\Sigma_k, \Pi_k, \Delta_k$, for $k \in \mathbb{N}$, are defined inductively by:*

- *Σ_0 is the class of the languages that are recursive;*
- *For $k \geq 1$, Σ_k is the class of the languages that are recursively enumerable in a set in Σ_{k-1} (that is semi-recognized by a Turing machine with an oracle in Σ_{k-1});*
- *For $k \in \mathbb{N}$, Π_k is defined as the class of languages whose complement are in Σ_k, and Δ_k is defined as $\Delta_k = \Pi_k \cap \Sigma_k$.*

Several characterizations of the sets of the arithmetical hierarchy are known: see [Odifreddi, 1992, Rogers Jr., 1987]. In particular assume a first order formula F, over some recursive predicates, characterizing the elements of a set $S \subset \mathbb{N}$, is given. Then S is in the arithmetical hierarchy and the Tarski-Kuratowski algorithm on formula F returns a level of the arithmetical hierarchy containing S: see [Odifreddi, 1992, Rogers Jr., 1987] for the full details.

The hyper-arithmetical hierarchy is an extension of the arithmetical hierarchy to constructive ordinal numbers. It consists of the classes of languages $\Sigma_1, \Sigma_2, \dots, \Sigma_k, \dots, \Sigma_\omega, \Sigma_{\omega+1}, \Sigma_{\omega+2}, \dots, \Sigma_{\omega 2}, \Sigma_{\omega 2+1}, \dots, \Sigma_{\omega^2}, \dots$ indexed by the constructive ordinal numbers. It is a strict hierarchy and it satisfies the strict inclusions $\Sigma_\alpha \subset \Sigma_\beta$ whenever $\alpha < \beta$. It can be related to the analytical hierarchy by $\Delta_1^1 = \cup_\beta \Sigma_\beta$: see [Rogers Jr., 1987].

The idea of the construction of this hierarchy is the following:

- Σ_1 is defined as the class of the recursively enumerable sets: that is to say Σ_1 is the class of the languages that are semi-recognized by a Turing machine.
- When k is a constructive ordinal and when the class Σ_k is defined, Σ_{k+1} is defined as the class of the languages that are recursively enumerable in a set in Σ_k: that is to say Σ_{k+1} is the class of the languages that are semi-recognized by some oracle Turing machine whose oracle is a language in Σ_k.
- When k is a constructive limit ordinal, $k = lim\, k_i$, and when the classes $(\Sigma_{k_i})_{i \in \mathbb{N}}$ are defined, Σ_k is defined as the class of the languages that are recursively enumerable in some fixed diagonalization of classes $(\Sigma_{k_i})_i$.

Summary:

If you prefer:

- Σ_1 = Recursively enumerable sets.
- ...
- Σ_{k+1} = Sets recursively enumerable in a set in Σ_k.
- ...
- Σ_ω = Sets recursively enumerable in a diagonalisation of $\Sigma_{\gamma<\omega}$
- $\Sigma_{\omega+1}$ = Sets recursively enumerable in a set in Σ_ω
- ...
- $\Sigma_{\alpha=\lim\gamma}$ = Sets recursively enumerable in a diagonalisation of $\Sigma_{\gamma<\alpha}$.
- $\Sigma_{\alpha+1}$ = Sets recursively enumerable in a set of Σ_α
- ...

It is then possible to relate the computational power of PCD systems in finite continuous time to the hyperarithmetical hierarchy ([Asarin and Maler, 1998, Bournez, 1999]).

Dimension	Languages semi-recognized
2	$< \Sigma_1$
3	Σ_1
4	Σ_2
5	Σ_ω
6	$\Sigma_{\omega+1}$
7	Σ_{ω^2}
8	Σ_{ω^2+1}
...	...
2p+1	$\Sigma_{\omega^{p-1}}$
2p+2	$\Sigma_{\omega^{p-1}+1}$

In particular, any set definable by some arithmetical formula is decided in dimension 5 !!!

Chapter 12

The General Purpose Analog Computers. Differential Analyzers

In 1941, Claude Shannon introduced in [Shannon, 1941] the GPAC model as a model for the Differential Analyzer [Bush, 1931], on which he worked as an operator.

12.1 Differential Analysers

Differential Analysers are mechanical (and later on electronics) continuous time analog machines.

The Differential Analyzer was used from the 1930s to the early 60s to solve numerical problems. For example, differential equations were used to solve ballistics problems. These devices were first built with mechanical components and later evolved to electronic versions.

One of the MIT Differential Analyser

First differential analyzers were mechanical. Electronic versions were used from late 40s until 70s.

Analog paradigm is selling some modern differential analyzers. I have one *Analog Paradigm Model-1* in my office at Ecole Polytechnique.

Underlying principles of the Differential Analysers can be attributed to Lord Kelvin 1876. First ever built machine was built under the supervision of V. Bush 1931 at MIT. Applications were from gunfire control up to aircraft design. They were intensively used during U.S. war effort.

12.2 The GPAC model

GPAC stands for *General Purpose Analog Computer*.

The GPAC was originally introduced by Shannon in [Shannon, 1941], and further refined in [Pour-El, 1974, Lipshitz and Rubel, 1987, Graça and Costa, 2003, Graça, 2004].

Basically, a GPAC is any circuit that can be build from the 4 basic units of Figure 12.1, that is to say from basic units realizing constants, additions, multiplications and integrations, all of them working over analog real quantities (that were corresponding to angles in the mechanical Differential Analysers, and later on to voltage in the electronic versions).

Actually, not all kinds of interconnections must be allowed since this may lead to undesirable behavior (*e.g.* non-unique outputs. For further details, refer to [Graça and Costa, 2003]).

In other words, a GPAC may be seen as a circuit built of interconnected black boxes, whose behavior is given by Figure 12.1, where inputs are functions of an independent variable called the *time* (in an electronic Differential Analyzer, inputs usually correspond to electronic voltages). These black boxes add or multiply two inputs, generate a constant, or solve a particular kind of Initial Value Problem defined with an Ordinary Differential Equation (ODE for short).

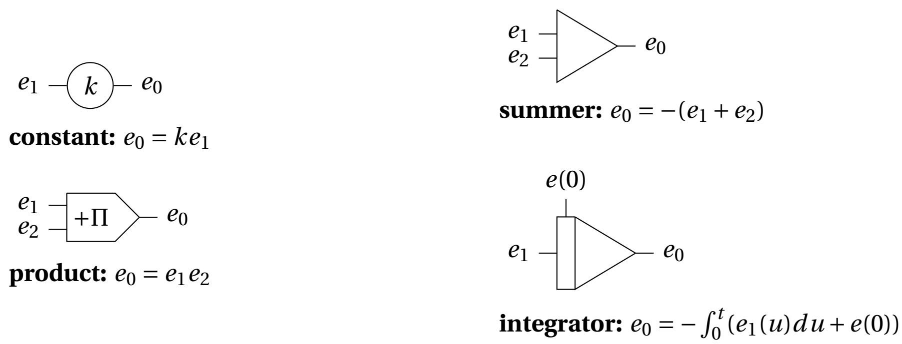


Figure 12.1: Circuit presentation of the GPAC: a circuit built from basic units. Presentation of the 4 types of units: constant, adder, multiplier, and integrator.

Figures 12.3 illustrates for example how the sine function can generated using two integrators, with suitable initial state, as being the solution of ordinary differential equation

$$\begin{cases} y'(t) = z(t) \\ z'(t) = -y(t) \end{cases}$$

with suitable initial conditions.

The original GPAC model introduced by Shannon has the feature that it works in *real time*: for example if the input t is updated in the GPAC circuit of Figure 12.3, then the output is immediately updated for the corresponding value of t.

Shannon, in his original paper, already mentioned that the GPAC generates polynomials, the exponential function, the usual trigonometric functions, their inverses, and their composition. More generally, Shannon claimed that all functions generated by a GPAC are differentially algebraic in the sense of the following definition.

Definition 12.1 *A unary function y is differentially algebraic (d.a.) on the interval I if there exists an $n \in \mathbb{N}$ and a nonzero polynomial p with real coefficients such that*

$$p\left(t, y, y', \ldots, y^{(n)}\right) = 0, \quad \text{on } I. \tag{12.1}$$

As a corollary, and noting that the Gamma function $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ is not d.a. [Rubel, 1989], we get that

Proposition 12.1 *The Gamma function cannot be generated by a GPAC.*

Another famous example of not d.a. function is Riemann's Zeta function $\zeta(x) = \sum_{k=0}^{\infty} \frac{1}{k^x}$ (proof of non d.a. by Hilbert).

If we have in mind that these functions are known to be computable under the computable analysis framework [Pour-El and Richards, 1989], the previous result has long been interpreted as evidence that the GPAC is a somewhat weaker model than computable analysis.

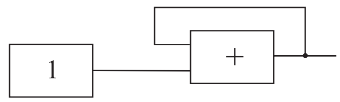


Figure 1: A circuit that admits no solutions as outputs.

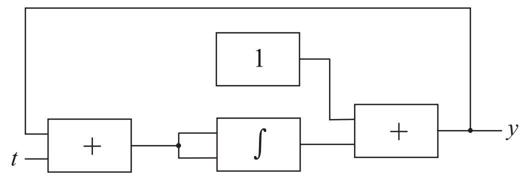


Figure 2: A circuit that admits two distinct solutions as outputs.

Figure 12.2: Problematic circuits: (I apologize: the representation of basic blocks differ from other figures as the current images are taken from some other source)
.

However, Shannon's proof relating functions generated by GPACs with d.a. functions was incomplete (as pointed out and partially corrected in [Pour-El, 1974, Lipshitz and Rubel, 1987]). Actually, as pointed out in [Graça and Costa, 2003], the original GPAC model suffers from several robustness problems.

12.3 GPAC and polynomomial Initial Value Problems

However, for the more robust class of GPACs defined in [Graça and Costa, 2003] by restricting the possible layout of a GPAC, the following stronger property holds:

Proposition 12.2 *A scalar function $f : \mathbb{R} \to \mathbb{R}$ is generated by a GPAC iff it is a component of the solution of a system*

$$y' = p(y, t), \tag{12.2}$$

where p is a vector of polynomials. A function $f : \mathbb{R} \to \mathbb{R}^k$ is generated by a GPAC iff all of its components are.

Basically, the idea of the proof is just to introduce a variable for each output of a basic unit, and write the corresponding ordinary differential equation (ODE), and observe that it can be written as an ODE with a polynomial right hand side.

For a concrete example of Proposition 12.2, see Figure 12.3. From now on, we will mostly talk about GPACs as being systems of ODEs of the type (12.2).

We say that a function $f : \mathbb{R} \to \mathbb{R}$ is generable (by a GPAC) if and only if it corresponds to some component of a solution of such a polynomial initial value problem (12.2).

The discussion on how to go from univariate to multivariate functions, that is to say from functions $f : \mathbb{R} \to \mathbb{R}^m$ to functions $f : \mathbb{R}^n \to \mathbb{R}^m$ is briefly discussed in

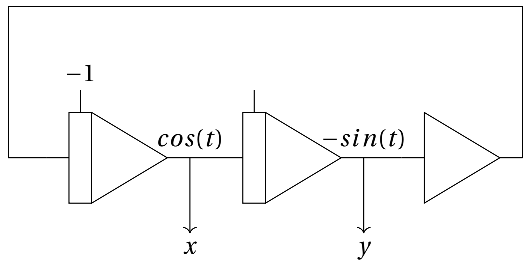


Figure 12.3: Example of GPAC circuit: computing sine and cosine with two variables

[Shannon, 1941], but no clear definitions and results for this case have been stated or proved previously, up to our knowledge. This is the purpose of the following part of the course. Another objective is to introduce basic measures of the resources used by a GPAC (in particular on the growth of functions), which might be used in the future to establish complexity results for functions generated with GPACs.

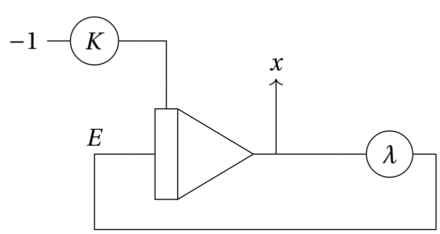
We will introduce in another course formally the notion of *generable* functions which are solutions of a polynomial initial-value problem (PIVP), and generalize this notion to several input variables. We will prove that this class enjoys a number of stability and robustness properties.

12.4 Programming with the GPAC

We here provide some examples of programs.

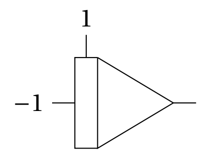
12.4.1 Exponential

Exponential: $E(t) = Kexp(-\lambda t)$

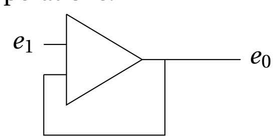


12.4.2 Linear operations

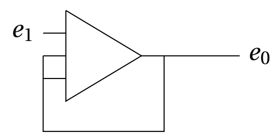
Linear function: $x(t) = t$



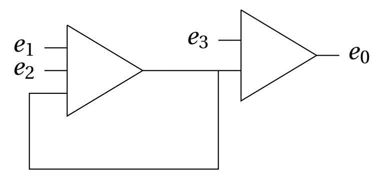
Linear operations:



$$e_0 = -\frac{e_1}{2}$$



$$e_0 = -\frac{e_1}{3}$$

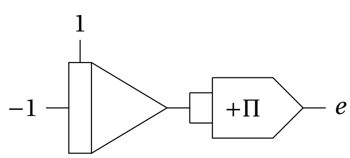


$$e_0 = \frac{e_1 + e_2}{2} - e_3$$

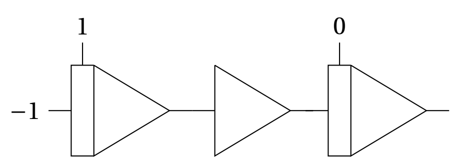
12.4.3 Polynomials

Parabola: $x(t) = (-1+t)^2$

Solution 1: With a product:

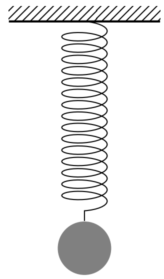


Solution 2: with integrators:



12.4.4 Damped Spring

Damped spring



mass m

spring constant k

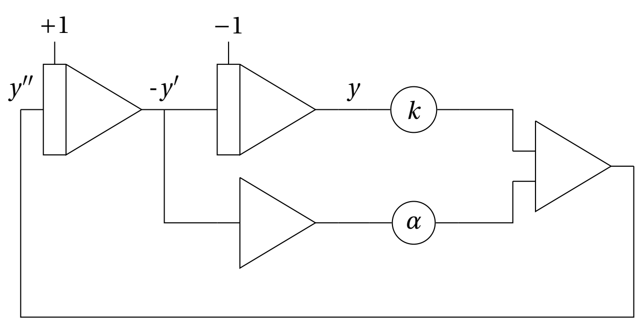
$$my'' + \alpha y' + ky = 0$$

Hence

$$y'' = -\frac{\alpha y' + ky}{m}$$

Damped spring: $y'' = -\alpha y' + ky$:

$m = 1$.



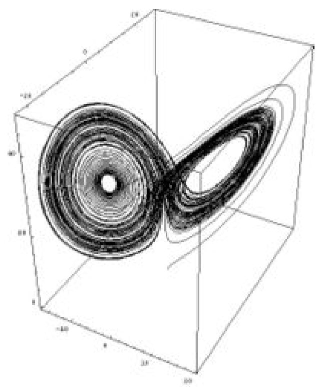
Example:

- $k = 0.8$

- $\alpha = 0.2$

12.4.5 Lorenz's attractor

Lorenz's attractor:



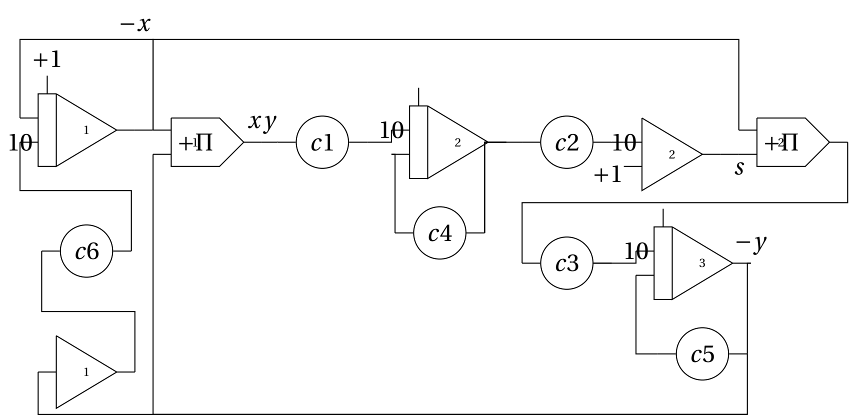
$$\begin{cases} x' &= \sigma(y-x) \\ y' &= \rho x - y - xz \\ z' &= xz - \beta z \end{cases}$$

$$\sigma = 10, \beta = 8/3, \rho = 28$$

Rescaling:

$$\begin{cases} x &= \int 1.8y - x dt + C \\ s &= 1 - 2.678z \\ y &= \int 1.5556xs - 0.1y dt \\ z &= \int 1.5xy - 0.2667z dt \end{cases}$$

The program:



$c1 = 0.15$, $c2 = 0.268$, $c3 = 0.1536$, $c4 = 0.2667$, $c5 = 0.1$, $c6 = 0.18$

Chapter 13

(GPAC) Generable Functions

13.1 Notations

In this document, $\mathbb{R}$ denotes the real numbers, $\mathbb{R}_+ = [0,+\infty)$ the nonnegative real numbers, $\mathbb{N} = \{0,1,2,\ldots\}$ the natural numbers, $\mathbb{Z}$ the integers, $[\![a,b]\!] = \{a,a+1,\ldots,b\}$ the integers between a and b, $\mathbb{Q}$ the rational numbers, $\mathbb{R}_P$ the polynomial time computable real numbers [Ko, 1991], $\mathbb{R}_G$ the smallest generable field (see Section 13.7). $M_{n,d}(\mathbb{K})$ denotes the set of $n \times d$ matrices over the ring $\mathbb{K}$. For any set X, $\mathscr{P}(X)$ denotes the powerset of X and $\#X$ the cardinal of X. For any function f, $\operatorname{dom} f$ is the domain of f, $f^{[n]}$ the n^{th} iterate of f, $f{\restriction}_X$ the restriction of f to X, $J_f(x)$ denotes the Jacobian matrix of f at x. For any vector $y \in \mathbb{R}^n$ and $e \leqslant n$, $y_{1..e} = (y_1,\ldots,y_e)$ denotes the first e components of y and $\|y\| = \max(|y_1|,\ldots,|y_n|)$ denotes the infinity norm. For any $x_0 \in \mathbb{R}^n$ and $r > 0$, $B_r(x_0) = \{x : \|x - x_0\|_2 < r\}$ denotes the open of radius r and center p for the euclidean norm. Given a (multivariate) polynomial p, $\deg(p)$ denotes its degree and Σp the sum of the absolute value of its coefficients. We denote by $\mathbb{K}[\mathbb{R}^d]$ the set of polynomial functions in d variables with coefficients in $\mathbb{K}$. Given a vector of polynomial $p = (p_1,\ldots,p_k)$, which we simply refer to as a polynomial, $\deg(p) = \max(\deg(p_1),\ldots,\deg(p_k))$ and $\Sigma p = \max(\Sigma p_1,\ldots,\Sigma p_k)$. We denote by $\mathbb{K}^k[\mathbb{R}^d]$ the set of vectors of polynomial functions in d variables of size k with coefficients in $\mathbb{K}$. In this article, we write poly to denote an unspecified polynomial. For any $x \in \mathbb{R}$, $\operatorname{sgn}(x)$ denotes the sign of x, $\lfloor x \rfloor$ the integer part of x, $\operatorname{int}_k(x) = \max(0,\min(k,\lfloor x \rfloor))$, $\lfloor x \rceil$ the nearest integer (undefined for $n + \frac{1}{2}$).

13.2 Generable functions

In this section, we will define a notion of function generated by a PIVP. From previous discussions, they correspond to functions generated by the General Purpose Analog Computers of Claude Shannon [Shannon, 1941];

This class of functions is closed by a number of natural operations such as arithmetic operators or composition. In particular, we will see that those functions are always analytic. The major property of this class is the stability by ODE solving: if

f is *generable* and y satisfies $y' = f(y)$ then y is generable. This means that we can design differential systems where the right-hand side contains much more general functions than polynomials, and this system can be rewritten to use polynomials only.

Several of the results here are extensions to the multidimensional case of results established in [Graça, 2007]. Moreover, a noticeable difference is that here we are also talking about complexity, whereas [Graça, 2007] is often not precise about the growth of functions as only motivated by computability theory.

In this section, $\mathbb{K}$ will always refer to a real field, for example $\mathbb{K} = \mathbb{Q}$. The basic definitions work for any such field but the main results will require some assumptions on $\mathbb{K}$. These assumptions will be formalized in Definition 13.2 and detailed in Section 13.7.

13.2.1 Unidimensional case

We start with the definition of generable functions from $\mathbb{R}$ to $\mathbb{R}^n$. Those are defined as the solution of some polynomial IVP (PIVP) with an additional boundedness constraint. This will be of course key to talk about complexity theory for the GPAC, since if no constraint is put on the growth of functions, it is easy to see that arbitrary growing functions can be generated by a GPAC (or, equivalently, by a PIVP), such as the $t \mapsto \exp(\exp(\ldots\exp(t)))$ function. Indeed consider the following system

$$\begin{cases} y_1(0) = 1 \\ y_2(0) = 1 \\ \quad\cdots \\ y_n(0) = 1 \end{cases} \qquad \begin{cases} y_1'(t) = y_1(t) \\ y_2(t) = y_1(t)y_2(t) \\ \quad\cdots \\ y_d'(t) = y_1(t)\cdots y_n(t) \end{cases}$$

This system has the form (**??**) and can be solved explicitly. It has the following solution:

$$y_1(t) = e^t \qquad y_{n+1}(t) = e^{y_n(t)-1} \qquad y_d(t) = e^{e^{\cdot^{\cdot^{\cdot^{e^{e^t}-1}}}}-1}$$

Hence, although previous papers about the GPAC studied computability, like [Shannon, 1941], [Pour-El, 1974], [Graça and Costa, 2003] or [Graça, 2004], they said nothing about complexity. And as the previous example shows, the output of a GPAC can have an arbitrarily high growth and thus arbitrarily high complexity. Hence, to distinguish between reasonable GPACs, it is natural to bound the growth of the outputs of a GPAC and use those bounds as a complexity measure. Moreover, as we have shown in [**?**], we can compute (in the Computable Analysis setting [Brattka et al., 2008]) the solution of a PIVP in time polynomial in the growth bound of the PIVP. This motivates the following definition (in what follows, $\mathbb{K}[\mathbb{R}^n]$ denotes polynomial functions with n variables and with coefficients in $\mathbb{K}$, where variables live in $\mathbb{R}^n$ and[1] $\mathbb{R}_+ = [0, +\infty[$):

[1] We write $[a,b]$ (respectively: $]a,b]$, $[a,b[$, $]a,b[$) for closed (resp. semi-closed, open) interval.

Definition 13.1 (Generable function) *Let* $\mathtt{sp}:\mathbb{R}_+\to\mathbb{R}_+$ *be a nondecreasing function and* $f:\mathbb{R}\to\mathbb{R}^m$. *We say that* $f\in\mathrm{GVAL}_{\mathbb{K}}[\mathtt{sp}]$ *if and only if there exists* $n\geqslant m$, $y_0\in\mathbb{K}^n$ *and* $p\in\mathbb{K}^n[\mathbb{R}^n]$ *such that there is a (unique)* $y:\mathbb{R}\to\mathbb{R}^n$ *satisfying for all time* $t\in\mathbb{R}$:

- $y'(t)=p(y(t))$ *and* $y(0)=y_0$ ▶ *y satisfies a differential equation*
- $f(t)=y_{1..m}(t)=(y_1(t),\dots,y_m(t))$ ▶ *f is a component of y*
- $\|y(t)\|\leqslant\mathtt{sp}(|t|)$ ▶ *y is bounded by* sp

The set of all generable functions is denoted by $\mathrm{GVAL}_{\mathbb{K}}=\bigcup_{\mathtt{sp}:\mathbb{R}\to\mathbb{R}_+}\mathrm{GVAL}_{\mathbb{K}}[\mathtt{sp}]$. *When this is not ambiguous, we do not specify the field* $\mathbb{K}$ *and write* $\mathrm{GVAL}[\mathtt{sp}]$ *or simply* GVAL. *We will also write* $\mathrm{GVAL}[\mathrm{poly}]$ *(or* $\mathrm{GVAL}_{\mathbb{K}}[\mathrm{poly}]$*) as a synonym of* $\mathrm{GVAL}[\mathtt{sp}]$ *(respectively:* $\mathrm{GVAL}_{\mathbb{K}}[\mathtt{sp}]$*) for some polynomial* sp *(see coming Remark 13.9).*

Remark 13.1 (Uniqueness) *The uniqueness of y in Definition 13.1 is a consequence of the Cauchy-Lipschitz theorem. Indeed a polynomial is a locally Lipschitz function.*

Remark 13.2 (Regularity) *As a consequence of the Cauchy-Lipschitz theorem, the solution y in Definition 13.1 is at least* C^∞. *It can be seen that it is in fact real analytic, as it is the case for analytic differential equations in general [Arnold, 1978].*

Remark 13.3 (Multidimensional output) *It should be noted that although Definition 13.1 defines generable functions with output in* $\mathbb{R}^m$, *it is completely equivalent to say that f is generable if and only if each of its component is (*i.e. f_i *is generable for every i); and restrict the previous definition to functions from* $\mathbb{R}$ *to* $\mathbb{R}$ *only. Also note that if y is the solution from Definition 13.1, then obviously y is generable.*

Although this might not be obvious at first glance, this class contains polynomials, and contains many elementary functions such as the exponential function, as well as the trigonometric functions. Intuitively, all functions in this class can be computed efficiently by classical machines, where sp measures some "hardness" in computing the function. We took care to choose the constants such as the initial time and value, and the coefficients of the polynomial in $\mathbb{K}$. The idea is to prevent any uncomputability from arising by the choice of uncomputable real numbers in the constants.

Example 13.1 (Polynomials are generable) *Let p in* $\mathbb{Q}(\pi)[\mathbb{R}]$. *For example* $p(x)=x^7-14x^3+\pi^2$. *We will show that* $p\in\mathrm{GVAL}_{\mathbb{K}}[\mathtt{sp}]$ *where* $\mathtt{sp}(x)=x^7+14x^3+\pi^2$. *We need to rewrite p with a polynomial differential equation: we immediately get that* $p(0)=\pi^2$ *and* $p'(x)=7x^6-42x^2$. *However, we cannot express* $p'(x)$ *as a* *polynomial of $p(x)$ only: we need access to x. This can be done by introducing a new variable $v(x)$ such that $v(x) = x$. Indeed, $v'(x) = 1$ and $v(0) = 0$. Finally we get:*

$$\begin{cases} p(0) = \pi^2 \\ p'(x) = 7v(x)^6 - 42v(x)^2 \end{cases} \qquad \begin{cases} v(0) = 0 \\ v'(x) = 1 \end{cases}$$

Formally, we define $y(x) = (p(x), x)$ and show that $y(0) = (\pi^2, 0) \in \mathbb{K}^2$ and $y'(x) = p(y(x))$ where $p_1(a,b) = 7b^6 - 42b^2$ and $p_2(a,b) = 1$. Also note that the coefficients are clearly in $\mathbb{Q}(\pi)$). We also need to check that sp *is a bound on $\|y(x)\|$ (for $x \geq 0$):*

$$\|y(x)\| = \max(|x|, |x^7 - 14x^3 + \pi^2|) \leq \mathtt{sp}(x)$$

This shows that $p \in \mathrm{GVAL}_{\mathbb{K}}[\mathtt{sp}]$ and can be generalized to show that any polynomial in one variable is generable.

Example 13.2 (Some generable elementary functions) *We will check that* $\exp \in \mathrm{GVAL}_{\mathbb{Q}}[\exp]$ *and* $\sin, \cos, \tanh \in \mathrm{GVAL}_{\mathbb{Q}}[x \mapsto 1]$. *We will also check that* $\arctan \in \mathrm{GVAL}_{\mathbb{Q}}[x \mapsto \max(x, \frac{\pi}{2})]$.

- *A characterization of the exponential function is the following:* $\exp(0) = 1$ *and* $\exp' = \exp$. *Since* $\|\exp\| = \exp$, *it is immediate that* $\exp \in \mathrm{GVAL}_{\mathbb{Q}}[\exp]$. *The exponential function might be the simplest generable function.*

- *The sine and cosine functions are related by their derivatives since* $\sin' = \cos$ *and* $\cos' = -\sin$. *Also* $\sin(0) = 0$ *and* $\cos(0) = 1$, *and* $\|(\sin(x), \cos(x))\| \leqslant 1$, *we get that* $\sin, \cos \in \mathrm{GVAL}_{\mathbb{Q}}[x \mapsto 1]$ *with the same system.*

- *The hyperbolic tangent function will be very useful in this paper. Is it known to satisfy the very simple polynomial differential equation* $\tanh' = 1 - \tanh^2$. *Since* $\tanh(0) = 0$ *and* $|\tanh(x)| \leqslant 1$, *this shows that* $\tanh \in \mathrm{GVAL}_{\mathbb{Q}}[x \mapsto 1]$.

- *Another very useful function will be the arctangent function. A possible definition of the arctangent is the unique function satisfying* $\arctan(0) = 0$ *and* $\arctan'(x) = \frac{1}{1+x^2}$. *Unfortunately this is neither a polynomial in* $\arctan(x)$ *nor in x. A common trick is to introduce a new variable $z(x) = \frac{1}{1+x^2}$ so that* $\arctan'(x) = z(x)$, *in the hope that z satisfies a PIVP. This is the case since $z(0) = 1$ and $z'(x) = \frac{-2x}{(1+x^2)^2} = -2xz(x)^2$ which is a polynomial in z and x. We introduce a new variable for x as we did in the previous examples. Finally, define $y(x) = (\arctan(x), \frac{1}{1+x^2}, x)$ and check that $y(0) = (0,1,0)$ and $y'(x) = (y_2(x), -2y_3(x)y_2(x)^2, 1)$. The $\frac{\pi}{2}$ bound on* arctan *is a textbook property, and the bound on the other variables is immediate.*

Not only the class of generable functions contains many classical and useful functions, but it is also closed under many operations. We will see that the sum, difference, product and composition of generable functions are still generable.

The issue of constants

Before moving on to the properties of this class, we need to mention the easily overlooked issue about constants, best illustrated as an example.

Example 13.3 (The issue of constants) *Let $\mathbb{K}$ be a field, containing at least the rational numbers. Assume that generable functions are closed under composition, that is for any two $f,g \in \mathrm{GVAL}_{\mathbb{K}}$ we have $f \circ g \in \mathrm{GVAL}_{\mathbb{K}}$. Let $\alpha \in \mathbb{K}$ and $g = x \mapsto \alpha$. Then for any $(f : \mathbb{R} \to \mathbb{R}) \in \mathrm{GVAL}_{\mathbb{K}}$, $f \circ g \in \mathrm{GVAL}_{\mathbb{K}}$. Using Definition 13.1, we get that $f(g(0)) \in \mathbb{K}$ which means $f(\alpha) \in \mathbb{K}$ for any $\alpha \in \mathbb{K}$. In other words, $\mathbb{K}$ must satisfy the following property:*

$$f(\mathbb{K}) \subseteq \mathbb{K} \qquad \forall f \in \mathrm{GVAL}_{\mathbb{K}}$$

This property does not hold for general fields.

The example above outlines the need for a stronger hypothesis on $\mathbb{K}$ if we want to be able to compose functions. Motivated by this example, we introduce the following notion of *generable field*.

Definition 13.2 (Generable field) *A field $\mathbb{K}$ is* generable *if and only if $\mathbb{Q} \subseteq \mathbb{K}$ and for any $\alpha \in \mathbb{K}$ and $(f : \mathbb{R} \to \mathbb{R}) \in \mathrm{GVAL}_{\mathbb{K}}$, we have $f(\alpha) \in \mathbb{K}$.*

⚠ From now on, we will assume that $\mathbb{K}$ is a generable field. See Section 13.7 for more details on this assumption.

Example 13.4 (Usual constants are generable) *In this paper, we will use again and again that some well-known constants belong to any generable field. We detail the proof for π and e:*

- *It is well-known that $\frac{\pi}{4} = \arctan(1)$. We saw in Example 13.2 that $\arctan \in \mathrm{GVAL}_{\mathbb{Q}}$ and since $1 \in \mathbb{K}$ we get that $\frac{\pi}{4} \in \mathbb{K}$ because $\mathbb{K}$ is a generable field. We conclude that $\pi \in \mathbb{K}$ because $\mathbb{K}$ is a field and $4 \in \mathbb{K}$.*
- *By definition, $e = \exp(1)$ and $\exp \in \mathrm{GVAL}_{\mathbb{Q}}$, so $e \in \mathbb{K}$ because $\mathbb{K}$ is a generable field and $1 \in \mathbb{K}$.*

Robustness of the class

Lemma 13.1 (Arithmetic on generable functions) *Let $f \in \mathrm{GVAL}[\mathrm{sp}]$ and $g \in \mathrm{GVAL}[\overline{\mathrm{sp}}]$.*

- $f+g, f-g \in \mathrm{GVAL}[\mathrm{sp}+\overline{\mathrm{sp}}]$
- $fg \in \mathrm{GVAL}[\max(\mathrm{sp},\overline{\mathrm{sp}},\mathrm{sp}\,\overline{\mathrm{sp}})]$
- $\frac{1}{f} \in \mathrm{GVAL}[\max(\mathrm{sp},\mathrm{sp}')]$ *where* $\mathrm{sp}'(t) = \frac{1}{|f(t)|}$, *if f never cancels*
- $f \circ g \in \mathrm{GVAL}[\max(\overline{\mathrm{sp}},\mathrm{sp}\circ\overline{\mathrm{sp}})]$

Note that the first three items only require that $\mathbb{K}$ is a field, whereas the last item also requires $\mathbb{K}$ to be a generable field.

Proof: Assume that $f:\mathbb{R}\to\mathbb{R}^m$ and $g:\mathbb{R}\to\mathbb{R}^\ell$. We will make a detailed proof of the product and composition cases, since the sum and difference are much simpler. The intuition follows from basic differential calculus and the chain rule: $(fg)' = f'g+fg'$ and $(f\circ g)' = g'(f'\circ g)$. Note that $\ell = 1$ for the composition to make sense and $\ell = m$ for the product to make sense (componentwise). The only difficulty in this proof is technical: the differential equation may include more variables than just the ones computing f and g. This requires a bit of notation to stay formal. Apply Definition 13.1 to f and g to get $p,\overline{p},y_0,\overline{y}_0$. Consider the following systems:

$$\begin{cases} y(0)= y_0\\ y'(t)= p(y(t))\\ \overline{y}(0)=\overline{y}_0\\ \overline{y}'(t)=\overline{p}(\overline{y}(t))\end{cases}\qquad \begin{cases} z_i(0)= y_{0,i}\overline{y}_{0,i}\\ z_i'(t)= p_i(y(t))\bar{y}_i(t)+y_i(t)\overline{p}_i(\bar{y}(t))\\ u_i(0)= f_i(\overline{y}_{0,1})\\ u_i'(t)=\overline{p}_i(\overline{y}(t))p(u(t))\end{cases}\qquad i\in[\![1,m]\!]$$

Those systems are clearly polynomial. By construction, u and z exist over $\mathbb{R}$ since $z_i(t)=y_i(t)\bar{y}_i(t)$ satisfies the differential equation over $\mathbb{R}$ (indeed y and $\bar{y}$ exist over $\mathbb{R}$). Similarly, $u_i(t)=y_i(\bar{y}(t))$ exists over $\mathbb{R}$ and satisfies the equation. Remember that by definition, for any $i\in[\![1,m]\!]$ and $j\in[\![1,\ell]\!]$, $f_i(t)=y_i(t)$ and $g_j(t)=z_j(t)$. Consequently, $z_i(t)=f_i(t)g_i(t)$ and $u_i(t)=f_i(g_1(t))$.

Also by definition, $\left\|y(t)\right\|\leqslant \mathtt{sp}(\mathtt{t})$ and $\left\|\overline{y}(t)\right\|\leqslant\overline{\mathtt{sp}}(t)$. It follows that $|z_i(t)|\leqslant |y_i(t)||\overline{y}_i(t)|\leqslant \mathtt{sp}(t)\overline{\mathtt{sp}}(t)$, and similarly we have $|u_i(t)|\leqslant|f_i(g_1(t))|\leqslant\mathtt{sp}(g_1(t))\leqslant \mathtt{sp}(\overline{\mathtt{sp}}(t))$.

The case of $\frac{1}{g}$ is very similar: define $g=\frac{1}{f}$ then $g'=-f'g^2$. The only difference is that we don't have an a priori bound on g except $\frac{1}{|f|}$, and we must assume that f is never zero for g to be defined over $\mathbb{R}$.

Finally, a very important note about constants and coefficients which appear in those systems. It is clear that $y_{0,i}\overline{y}_{0,i}\in\mathbb{K}$ because $\mathbb{K}$ is a field. Similarly, for $\frac{1}{f}$ we have $\frac{1}{f(0)}=\frac{1}{y_{0,1}}\in\mathbb{K}$. However, there is no reason in general for $f_i(\overline{y}_{0,1})$ to belong to $\mathbb{K}$, and this is where we need the assumption that $\mathbb{K}$ is generable. ☐

13.2.2 Multidimensional case

We introduced generable functions as a special kind of function from $\mathbb{R}$ to $\mathbb{R}^n$. We saw that this class nicely contains polynomials, however it comes with two defects which prevents other interesting functions from being generable:

- The domain of definition is $\mathbb{R}$: this is very strong, since other "easy" targets such as tan, log or even $x\mapsto\frac{1}{x}$ cannot be defined, despite satisfying polynomial differential equations.

- The domain of definition is one-dimensional: it would be useful to define generable functions in several variables, like multivariate polynomials.

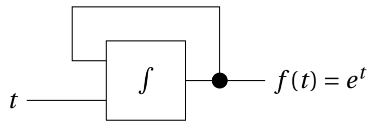


Figure 13.1: Simple GPAC

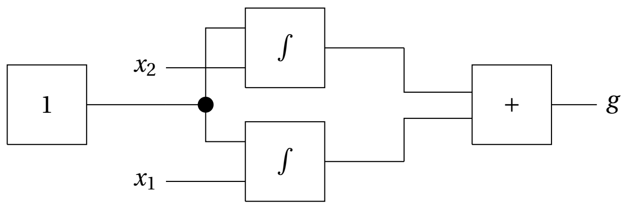


Figure 13.2: GPAC with two inputs

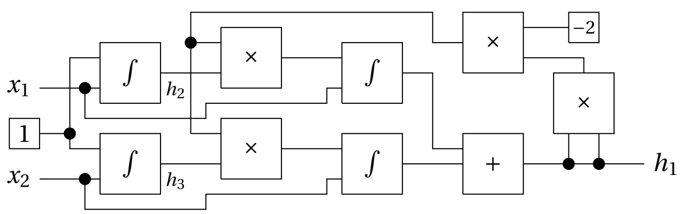


Figure 13.3: A more involved multidimensional GPAC

The first issue can be dealt with by adding restrictions on the domain where the differential equation holds, and by shifting the initial condition (0 might not belong to the domain). Overcoming the second problem is less obvious.

About motivation of definitions

The examples below give two intuitions before introducing the formal definition. The first example draws inspiration from multivariate calculus and differential form theory. The second example focuses on GPAC composition. As we will see, both examples highlight the same properties of multidimensional generable functions.

Example 13.5 (Multidimensional GPAC) *The history and motivation for the GPAC have been described above. The GPAC is the starting point for the definition of generable functions. It crucially relies on the integrator unit to build interesting circuits. In modern terms, the integration is often done implicitly with respect to time, as shown in Figure 13.1 where the corresponding equation is* $f(t) = \int f$*, or* $f' = f$*. Notice that the circuit has a single "floating input" which is* t *and is only*

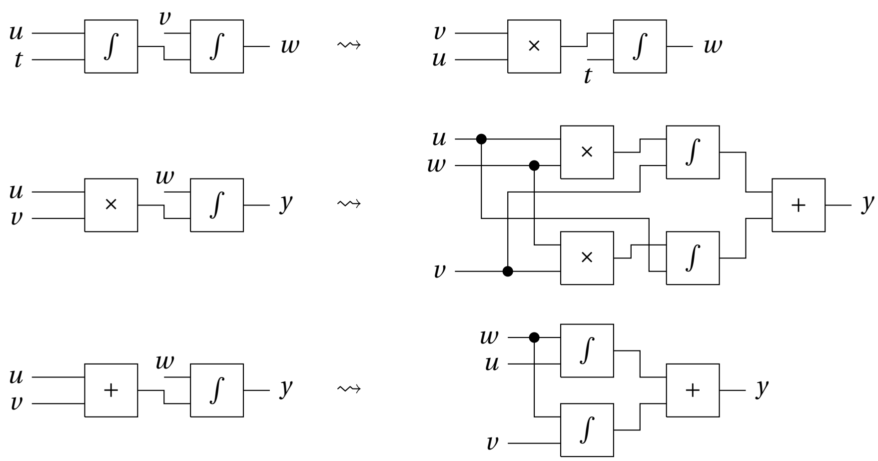


Figure 13.4: GPAC rewriting

used in the “derivative port” of the integrator. What would be the meaning of a circuit with several such inputs, as shown in Figure 13.2 ? Formally writing the system and differentiating gives:

$$g = \int 1 dx_1 + \int 1 dx_2 = x_1 + x_2$$
$$dg = dx_1 + dx_2$$

Figure 13.3 gives a more interesting example to better grasp the features of these GPAC. Using the same “trick” as before we get:

$$\begin{aligned} h_2 &= \textstyle\int 1 dx_1 & dh_2 &= dx_1 \\ h_3 &= \textstyle\int 1 dx_2 & dh_3 &= dx_2 \\ h_1 &= \textstyle\int -2h_1^2 h_2 dx_1 + \int -2h_1^2 h_3 dx_2 & dh_1 &= -2h_1^2 h_2 dx_1 - 2h_1^2 h_3 dx_2 \end{aligned}$$

It is now apparent that the computed function h satisfies a special property because $dh_1(x) = p_1(h_1, h_2, h_3)dx_1 + p_2(h_1, h_2, h_3)dx_2$ where p_1 and p_2 are polynomials. In other words, $dh_1 = p(h) \cdot dx$ where $h = (h_1, h_2, h_3)$, $x = (x_1, x_2)$ and $p = (p_1, p_2)$ is a polynomial vector. We obtain similar equations for h_2 and h_3. Finally, $dh = q(h)dx$ where $q(h)$ is the polynomial matrix given by:

$$q(h) = \begin{pmatrix} -2h_1^2 h_2 & -2h_1^2 h_3 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

This can be equivalently stated as $J_h = q(h)$. This is a generalization of PIVP to polynomial partial differential equations.

To complete this example, note that it can be solved exactly and $h_1(x_1, x_2) = \frac{1}{x_1^2+x_2^2}$ which is defined over $\mathbb{R}^2 \setminus \{(0,0)\}$.

Example 13.6 (GPAC composition) *Another way to look at Figure 13.3 and Figure 13.2 is to imagine that $x_1 = X_1(t)$ and $x_2 = X_2(t)$ are functions of the time (produced by other GPACs), and rewrite the system in the time domain with $h = H(t)$:*

$$\begin{aligned} H_2'(t) &= X_1'(t) \\ H_3'(t) &= X_2'(t) \\ H_1'(t) &= -2H_1(t)^2H_2(t)X_1'(t) - 2H_1(t)^2H_3(t)X_2'(t) \end{aligned}$$

We obtain a system similar to the unidimensional PIVP: for a given choice of X we have $H'(t) = q(H(t))X'(t)$ where $q(h)$ is the polynomial matrix given by:

$$q(h) = \begin{pmatrix} -2h_1^2h_2 & -2h_1^2h_3 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

*Note that this is the same polynomial matrix as in the previous example. The relationship between the time domain H and the original h is simply given by $H(t) = h(x(t))$. This approach has a natural interpretation on the GPAC circuit in terms of circuit rewriting. Assume that x_1 and x_2 are the outputs of two GPACs (with input t), i.e. $x_1 = x_1(t)$ and $x_2 = x_2(t)$. Then x_1, x_2 are given by the first two components of a polynomial ODE (**??**), i.e. $x_1(t) = y_1(t)$ and $x_2(t) = y_2(t)$. Moreover one has $x_1'(t) = p_1(y), x_2'(t) = p_2(y)$. That means that the output $H(t) = (H_1(t), H_2(t), H_3(t))$ of the GPAC of Figure 13.3 satisfies*

$$H'(t) = q(H(t))X'(t) = q(H(t))(p_1(y), p_2(y))$$

and therefore consists of the first three components of the polynomial ODE given by

$$\begin{aligned} H' &= q(H(t))(p_1(y), p_2(y)) \\ y' &= p(y) \end{aligned}$$

Thus, if x_1 and x_2 are the outputs of the some GPACs, depending on one input t, and if we connect the outputs of these two GPACs to the inputs of the two-dimensional GPAC of Figure 13.3, we obtain a one-input GPAC computing $H(t)$, where t is the input. Note that in a normal GPAC, the time t is the only valid input of the derivative port of the integrator, so we need to rewrite integrators which violate this rule. This can be done by rewriting the ODE defining $H(t)$ into a polynomial ODE as done above, and then by implementing a GPAC which computes the solution of this ODE such that the time t is the only valid input *of the derivative port of each integrator (this is trivial to implement). This procedure always stops in finite time. Moreover it always works as long as $q(\cdot)$ is a matrix consisting of polynomials.*

Formal definitions

These considerations lead to state that the following generalization is clearly the one we want:

Definition 13.3 (Generable function) *Let $d, \ell \in \mathbb{N}$, I an open and connected subset of $\mathbb{R}^d$, $\mathtt{sp} : \mathbb{R}_+ \to \mathbb{R}_+$ a nondecreasing function and $f : I \to \mathbb{R}^\ell$. We say that $f \in \mathrm{GVAL}_{\mathbb{K}}[\mathtt{sp}]$ if and only if there exists $n \geqslant \ell$, $p \in M_{n,d}(\mathbb{K})[\mathbb{R}^n]$, $x_0 \in (\mathbb{K}^d \cap I)$, $y_0 \in \mathbb{K}^n$ and $y : I \to \mathbb{R}^n$ satisfying for all $x \in I$:*

- $y(x_0) = y_0$ *and* $J_y(x) = p(y(x))$ *(i.e.* $\partial_j y_i(x) = p_{ij}(y(x))$*)* ▶ *y satisfies a differential equation*
- $f(x) = y_{1..\ell}(x)$ ▶ *f is a component of y*
- $\|y(x)\| \leqslant \mathtt{sp}(\|x\|)$ ▶ *y is bounded by* $\mathtt{sp}$

Remark 13.4 (Uniqueness) *The uniqueness of y in Definition 13.3 can be seen in two different ways: by uniqueness of the unidimensional case and by analyticity. Note that the existence of y (and thus the domain of definition) is a hypothesis of the definition.*

Consider $x \in I$ and γ a smooth curve[a] from x_0 to x with values in I and consider $z(t) = y(\gamma(t))$ for $t \in [0,1]$. It can be seen that $z'(t) = J_y(\gamma(t))\gamma'(t) = p(y(\gamma(t))\gamma'(t) = p(z(t))\gamma'(t)$, $z(0) = y(x_0) = y_0$ and $z(1) = y(x)$. The initial value problem $z(0) = y_0$ and $z'(t) = p(z(t))\gamma'(t)$ satisfies the hypothesis of the Cauchy-Lipschitz theorem and as such admits a unique solution. Since this IVP is independent of y, the value of $z(1)$ is unique and must be equal to $y(x)$, for any solution y and any x. This implies that y must be unique.

Alternatively, use Proposition 13.4 to conclude that any solution must be analytic. Assume that there are two solutions y and z. Then all partial derivatives at any order at the initial point x_0 are equal because they only depend on y_0. Thus y and z have the same partial derivatives at all order and must be equal on a small open ball around y_0. A classical argument of finite covering with open balls then extends this argument to any point of the interior of domain of definition that is connected to y_0. Since the domain of definition is assumed to be open and connected, this concludes to the equality of y and z.

[a] see Remark 13.6

Remark 13.5 (Regularity) *In the euclidean space $\mathbb{R}^n$, C^k smoothness is equivalent to the smoothness of the order k partial derivatives. Consequently, the equation $J_y = p(y)$ on the open set I immediately proves that y is C^∞. Propo-*

sition 13.4 shows that y is in fact real analytic.

Remark 13.6 (Domain of definition) *Definition 13.3 requires the domain of definition of f to be connected, otherwise it would not make sense. Indeed, we can only define the value of f at point u if there exists a path from x_0 to u in the domain of f. It could seem, at first sight, that the domain being "only" connected may be too weak to work with. This is not the case, because in the euclidean space $\mathbb{R}^d$,* open *connected subsets are always smoothly arc connected, that is any two points can be connected using a smooth C^1 (and even C^∞) arc. Proposition 13.5 extends this idea to generable arcs, with a very useful corollary.*

Remark 13.7 (Multidimensional output) *Remark 13.3 also applies to this definition: $f :\subseteq \mathbb{R}^d \to \mathbb{R}^n$ is generable if and only if each of its component is generable (*i.e. *f_i is generable for all i).*

Remark 13.8 (Definition consistency) *It should be clear that Definition 13.3 and Definition 13.1 are consistent. More precisely, in the case of unidimensional function ($d = 1$) with domain of definition $I = \mathbb{R}$, both definitions are exactly the same since $J_y = y'$ and $M_{n,1}(\mathbb{R}) = \mathbb{R}^n$.*

The following example focuses on the second issue mentioned at the beginning of the section, namely the domain of definition.

Example 13.7 (Inverse and logarithm functions) *We illustrate that the choice of the domain of definition makes important differences in the nature of the function.*

- *Let $0 < \varepsilon < 1$ and define $f_\varepsilon : x \in]\varepsilon,\infty[\mapsto \frac{1}{x}$. It can be seen that $f_\varepsilon'(x) = -f_\varepsilon(x)^2$ and $f_\varepsilon(1) = 1$. Furthermore, $|f_\varepsilon(x)| \leqslant \frac{1}{\varepsilon}$ thus $f_\varepsilon \in \mathrm{GVAL}[\alpha \mapsto \frac{1}{\varepsilon}]$. So in particular, $f_\varepsilon \in \mathrm{GVAL}[\mathrm{poly}]$ for any $\varepsilon > 0$. Something interesting arises when $\varepsilon \to 0$: define $f_0(x) = x \in (0,\infty) \mapsto \frac{1}{x}$. Then f_0 is still generable and $|f_0(x)| \leqslant \frac{1}{|x|}$. Thus $f_0 \in \mathrm{GVAL}[\alpha \mapsto \frac{1}{\alpha}]$ but $f_0 \notin \mathrm{GVAL}[\mathrm{poly}]$. Note that strictly speaking, $f_0 \in \mathrm{GVAL}[\mathtt{sp}]$ where $\mathtt{sp}(\alpha) = \frac{1}{\alpha}$ and $\mathtt{sp}(0) = 0$ because the bound function needs to be defined over $\mathbb{R}_+$.*
- *A similar phenomenon occurs with the logarithm: define $g_\varepsilon : x \in (\varepsilon,\infty) \mapsto \ln(x)$. Then $g_\varepsilon'(x) = f_\varepsilon(x)$ and $g_\varepsilon(1) = 0$. Furthermore, $|g_\varepsilon(x)| \leqslant \max(|x|, |\ln\varepsilon|)$. Thus $g_\varepsilon \in \mathrm{GVAL}[\alpha \mapsto \max(\alpha, |\ln\varepsilon|, \frac{1}{\varepsilon})]$, and in particular $g_\varepsilon \in \mathrm{GVAL}[\mathrm{poly}]$ for any $\varepsilon > 0$. Similarly, $g_0 : x \in]0,\infty[\mapsto \ln(x)$ is generable but does not belong to $\mathrm{GVAL}[\mathrm{poly}]$.*

Example 13.8 (Classical non-generable functions) *While many of the usual real functions are known to be generated by a GPAC, a notable exception is Euler's Gamma function* $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ *function or Riemann's Zeta function* $\zeta(x) = \sum_{k=0}^\infty \frac{1}{k^x}$ *[Shannon, 1941], [Pour-El and Richards, 1989]. Furthermore, Riemann's Zeta function (over, for example,* $[2,\infty)$*) is an example of real-analytic, polynomially-bounded that is not in* GVAL[poly].

Example 13.9 (Generable functions not in GVAL[poly]) *We have seen that Riemann's Zeta function* ζ *is an example of a function not in* GVAL[poly] *due to the fact that it is not generable. An example of a generable function not belonging to* GVAL[poly] *is the exponential* e^x *because, while it is generable, its derivative is not bounded by another polynomial. Note that it is quite possible to have bounded generable functions which do not belong to* GVAL[poly]. *An example is the function given by* $f(x) = \sin(e^x)$ *which is generable and bounded, but its derivative* $f'(x) = e^x \cos(e^x)$ *is not bounded by any polynomial.*

The previous examples show that $\text{GVAL}_{\mathbb{K}}[\mathtt{sp}]$ can be used to define a proper hierarchy of generable functions. Adapting the examples given in Example 13.9 one can show for instance that

$$\text{GVAL}[\text{poly}] \subsetneq \text{GVAL}[e^x] \subsetneq \text{GVAL}[e^{e^x}] \subsetneq \dots$$

In particular these examples show the following result.

Theorem 13.1 (Existence of noncollapsing classes) $\text{GVAL}[\text{poly}] \subsetneq \text{GVAL}$.

13.3 Stability properties

In this section, the major results will the be stability of multidimensional generable functions under arithmetical operators, composition and ODE solving. Note that some of the results use properties on $\mathbb{K}$ which can be found in Section 13.7.1.

Lemma 13.2 (Arithmetic on generable functions) *Let* $d, \ell, n, m \in \mathbb{N}$, $\text{sp}, \overline{\text{sp}} : \mathbb{R} \to \mathbb{R}_+$, $f :\subseteq \mathbb{R}^d \to \mathbb{R}^n \in \text{GVAL}[\text{sp}]$ *and* $g :\subseteq \mathbb{R}^\ell \to \mathbb{R}^m \in \text{GVAL}[\overline{\text{sp}}]$. *Then:*

- $f + g, f - g \in \text{GVAL}[\text{sp} + \overline{\text{sp}}]$ *over* $\text{dom} f \cap \text{dom} g$ *if* $d = \ell$ *and* $n = m$
- $fg \in \text{GVAL}[\max(\text{sp}, \overline{\text{sp}}, \text{sp}\,\overline{\text{sp}})]$ *if* $d = \ell$ *and* $n = m$
- $f \circ g \in \text{GVAL}[\max(\overline{\text{sp}}, \text{sp} \circ \overline{\text{sp}})]$ *if* $m = d$ *and* $g(\text{dom} g) \subseteq \text{dom} f$

Proof: We focus on the case of the composition, the other cases are very similar.

Apply Definition 13.3 to f and g to respectively get $l, \bar{l} \in \mathbb{N}$, $p \in M_{l,d}(\mathbb{K})[\mathbb{R}^l]$, $\bar{p} \in M_{\bar{l},\ell}(\mathbb{K})[\mathbb{R}^{\bar{l}}]$, $x_0 \in \text{dom} f \cap \mathbb{K}^d$, $\bar{x}_0 \in \text{dom} g \cap \mathbb{K}^\ell$, $y_0 \in \mathbb{K}^l$, $\bar{y}_0 \in \mathbb{K}^{\bar{l}}$, $y : \text{dom} f \to \mathbb{R}^l$

and $\bar{y} : \operatorname{dom} g \to \mathbb{R}^{\bar{l}}$. Define $h = y \circ g$, then $J_h = J_y(g)J_g = p(h)\bar{p}_{1..m}(\bar{y})$ and $h(\bar{x}_0) = y(\bar{y}_0) \in \mathbb{K}^l$ by Corollary 13.2. In other words $(\bar{y}, h)$ satisfy:

$$\begin{cases} \bar{y}(\bar{x}_0) = y_0 \in \mathbb{K}^{\bar{l}} \\ h(\bar{x}_0) = y(\bar{y}_0) \in \mathbb{K}^l \end{cases} \qquad \begin{cases} \bar{y}' = \bar{p}(\bar{y}) \\ h' = p(h)\bar{p}_{1..m}(\bar{y}) \end{cases}$$

This shows that $f \circ g = z_{1..m} \in \text{GVAL}$. Furthermore,

$$\begin{aligned} \left\|(\bar{y}(x), h(x))\right\| &\leqslant \max(\left\|\bar{y}(x)\right\|, \left\|y(g(x))\right\|) \\ &\leqslant \max(\overline{\mathrm{sp}}(\|x\|), \mathrm{sp}(\left\|g(x)\right\|)) \\ &\leqslant \max(\overline{\mathrm{sp}}(\|x\|), \mathrm{sp}(\overline{\mathrm{sp}}(\|x\|))). \end{aligned}$$

□

Our main result is that the solution to an ODE whose right hand-side is generable, and possibly depends on an external and C^1 control, may be rewritten as a GPAC. A corollary of this result is that the solution to a generable ODE is generable.

Proposition 13.1 (Generable ODE rewriting) *Let $d, n \in \mathbb{N}$, $I \subseteq \mathbb{R}^n$, $X \subseteq \mathbb{R}^d$, $\mathrm{sp} : \mathbb{R}_+ \to \mathbb{R}_+$ and $(f : I \times X \to \mathbb{R}^n) \in \text{GVAL}_{\mathbb{K}}[\mathrm{sp}]$. Define $\overline{\mathrm{sp}} = \max(\mathrm{id}, \mathrm{sp})$. Then there exists $m \in \mathbb{N}$, $(g : I \times X \to \mathbb{R}^m) \in \text{GVAL}_{\mathbb{K}}[\overline{\mathrm{sp}}]$ and $p \in \mathbb{K}^m[\mathbb{R}^m \times \mathbb{R}^d]$ such that for any interval J, $t_0 \in \mathbb{K} \cap J$, $y_0 \in \mathbb{K}^n \cap J$, $y \in C^1(J, I)$ and $x \in C^1(J, X)$, if y satisfies:*

$$\begin{cases} y(t_0) = y_0 \\ y'(t) = f(y(t), x(t)) \end{cases} \qquad \forall t \in J$$

then there exists $z \in C^1(J, \mathbb{R}^m)$ such that:

$$\begin{cases} z(t_0) = g(y_0, x(t_0)) \\ z'(t) = p(z(t), x'(t)) \end{cases} \qquad \begin{cases} y(t) = z_{1..d}(t) \\ \|z(t)\| \leqslant \overline{\mathrm{sp}}(\max(\left\|y(t)\right\|, \|x(t)\|)) \end{cases} \qquad \forall t \in J$$

Proof: Apply Definition 13.3 to f get $m \in \mathbb{N}$, $p \in M_{m,n+d}(\mathbb{K})[\mathbb{R}^m]$, $f_0 \in \operatorname{dom} f \cap \mathbb{K}^d$, $w_0 \in \mathbb{K}^m$ and $w : \operatorname{dom} f \to \mathbb{R}^m$ such that $w(f_0) = w_0$, $J_{w(v)} = p(w(v))$, $\|w(v)\| \leqslant \mathrm{sp}(\|v\|)$ and $w_{1..n}(v) = f(v)$ for all $v \in \operatorname{dom} f$. Define $u(t) = w(y(t), x(t))$, then:

$$\begin{aligned} u'(t) &= J_w(y(t), x(t))(y'(t), x'(t)) \\ &= p(w(y(t), x(t)))(f(y(t), x(t)), x'(t)) \\ &= p(u(t))(u_{1..n}(t), x'(t)) \\ &= q(u(t), x'(t)) \end{aligned}$$

where $q \in \mathbb{K}^m[\mathbb{R}^{m+d}]$ and $u(t_0) = w(y(t_0)) = w(y_0, x(t_0))$. Note that w itself is a generable function and more precisely $w \in \text{GVAL}_{\mathbb{K}}[\text{poly}]$ by definition. Finally, note that $y'(t) = u_{1..d}(t)$ so that we get for all $t \in J$:

$$\begin{cases} y(t_0) = y_0 \\ y'(t) = u_{1..d}(t) \end{cases} \qquad \begin{cases} u(t_0) = w(y_0, x(t_0)) \\ u'(t) = q(u(t), x'(t)) \end{cases}$$

Define $z(t) = (y(t), u(t))$, then $z(t_0) = (y_0, w(y_0, x(t_0))) = g(y_0, x(t_0))$ where $y_0 \in \mathbb{K}^n$ and $w \in \mathrm{GVAL}_{\mathbb{K}}[\mathrm{sp}]$ so $g \in \mathrm{GVAL}_{\mathbb{K}}[\overline{\mathrm{sp}}]$. And clearly $z'(t) = r(z(t), x'(t))$ where $r \in \mathbb{K}^{n+m}[\mathbb{R}^{n+m}]$. Finally, $\|z(t)\| = \max(\|y(t)\|, \|w(y(t), x(t))\|) \leqslant \max(\|y(t)\|, \mathrm{sp}(\max(\|y(t)\|, \|x(t)\|)) \leqslant \overline{\mathrm{sp}}(\max(\|y(t)\|, \|x(t)\|))$. □

A simplified version of this lemma shows that generable functions are closed under ODE solving.

Corollary 13.1 (Generable functions are closed under ODE) *Let* $d \in \mathbb{N}$, $J \subseteq \mathbb{R}$ *an interval,* $\mathrm{sp}, \overline{\mathrm{sp}} : \mathbb{R}_+ \to \mathbb{R}_+$, $f :\subseteq \mathbb{R}^d \to \mathbb{R}^d$ *in* $\mathrm{GVAL}[\mathrm{sp}]$, $t_0 \in \mathbb{K} \cap J$ *and* $y_0 \in \mathbb{K}^d \cap \operatorname{dom} f$. *Assume there exists* $y : J \to \operatorname{dom} f$ *satisfying for all* $t \in J$:

$$\begin{cases} y(t_0) = y_0 \\ y'(t) = f(y(t)) \end{cases} \qquad \|y(t)\| \leqslant \overline{\mathrm{sp}}(t)$$

Then $y \in \mathrm{GVAL}[\max(\overline{\mathrm{sp}}, \mathrm{sp} \circ \overline{\mathrm{sp}})]$ *and is unique.*

Remark 13.9 (Polynomially bounded generable functions) *In light of the stability properties above, the class of* polynomially bounded *generable functions,*

$$\mathrm{GVAL}[\mathrm{poly}] = \bigcup_{k=1}^{\infty} \mathrm{GVAL}[\alpha \mapsto k\alpha^k]$$

is particularly interesting because it is stable by operations: addition, multiplication, composition and ODE solving (provided the solution is polynomially bounded). Notice that $\mathrm{GVAL}[\mathrm{poly}]$ *is not simply the intersection of* GVAL *with the set of functions bounded by a polynomial, as shown in Example 13.9.*

Our last result is simple but very useful. Generable functions are continuous and continuously differentiable, so locally Lipschitz continuous. We can give a precise expression for the modulus of continuity in the case where the domain of definition is simple enough.

Proposition 13.2 (Modulus of continuity) *Let* $\mathrm{sp} : \mathbb{R}_+ \to \mathbb{R}_+$, $f \in \mathrm{GVAL}[\mathrm{sp}]$. *There exists* $q \in \mathbb{K}[\mathbb{R}]$ *such that for any* $x_1, x_2 \in \operatorname{dom} f$, *if* $[x_1, x_2] \subseteq \operatorname{dom} f$ *then* $\|f(x_1) - f(x_2)\| \leqslant \|x_1 - x_2\| \, q(\mathrm{sp}(\max(\|x_1\|, \|x_2\|)))$. *In particular, if* $\operatorname{dom} f$ *is convex then* f *has a polynomial modulus of continuity.*

Proof: Apply Definition 13.3 to get d, ℓ, n, p, x_0, y_0 and y. Let $k = \deg(p)$. Recall that for a matrix, the subordinate norm is given by $|||M||| = \max_i \sum_j |M_{ij}|$. Then:

$$\begin{aligned}
\|f(x_1)-f(x_2)\| = \left\|\int_{x_1}^{x_2} J_{y_{1..\ell}}(x)dx\right\| &= \left\|\int_0^1 J_{y_{1..\ell}}((1-\alpha)x_1+\alpha x_2)(x_2-x_1)d\alpha\right\| \\
&\leqslant \int_0^1 |||J_{y_{1..\ell}}((1-\alpha)x_1+\alpha x_2)|||\cdot\|x_2-x_1\|\,d\alpha \\
&\leqslant \|x_2-x_1\|\int_0^1 \max_{i\in[\![1,\ell]\!]}\sum_{j=1}^{d}|p_{ij}(y((1-\alpha)x_1+\alpha x_2))|d\alpha \\
&\leqslant \|x_2-x_1\|\int_0^1 \max_{i\in[\![1,\ell]\!]}\sum_{j=1}^{d}\Sigma p\max(1,\left\|y((1-\alpha)x_1+\alpha x_2)\right\|)^k)d\alpha \\
&\leqslant \|x_2-x_1\|\int_0^1 \max_{i\in[\![1,\ell]\!]} d\Sigma p\max(1,\mathtt{sp}(\|(1-\alpha)x_1+\alpha x_2\|))^k d\alpha \\
&\leqslant \|x_2-x_1\|\int_0^1 d\Sigma p\max(1,\mathtt{sp}(\max(\|x_1\|,\|x_2\|)))^k d\alpha \\
&\leqslant \|x_2-x_1\|\,d\Sigma p\max(1,\mathtt{sp}(\max(\|x_1\|,\|x_2\|)))^k
\end{aligned}$$

□

13.4 Analyticity of generable functions

It is a well-known result that the solution of a PIVP $y' = p(y)$ (and more generally, of an analytic differential equation $y' = f(y)$ where f is analytic) is real analytic on its domain of definition. In the previous section we defined a generalized notion of generable function satisfying $J_y = p(y)$ which analyticity is less immediate. In this section we go through the proof in detail, which of course subsumes the result for PIVP.

We recall a well-known characterization of analytic functions. It is indeed much easier to show that a function is infinitely differentiable and of controlled growth, rather than showing the convergence of the Taylor series.

Proposition 13.3 (Characterization of analytic functions) *Let $f \in C^\infty(U)$ for some open subset U of $\mathbb{R}^m$. Then f is analytic on U if and only if, for each $u \in U$, there are an open ball V, with $u \in V \subseteq U$, and constants $C > 0$ and $R > 0$ such that the derivatives of f satisfy*

$$|\partial_\alpha f(x)| \leqslant C\frac{\alpha!}{R^{|\alpha|}} \qquad x \in V, \alpha \in \mathbb{N}^m$$

Proof:See proposition 2.2.10 of [**?**]. □

In order to use this result, we show that the derivatives of generable functions at a point x do not grow faster than the described bound. We use a generalization of Faà di Bruno formula for the derivatives of a composition.

Theorem 13.2 (Generalised Faà di Bruno's formula) *Let $f : X \subseteq \mathbb{R}^d \to Y \subseteq \mathbb{R}^n$ and $g : Y \to \mathbb{R}$ where X, Y are open sets and f, g are sufficiently smooth functions[a]. Let $\alpha \in \mathbb{N}^d$ and $x \in X$, then*

$$\partial_\alpha (g \circ f)(x) = \alpha! \sum_{(s,\beta,\lambda)\in\mathscr{D}_\alpha} \partial_\lambda g(f(x)) \prod_{k=1}^{s} \frac{1}{\lambda_k!} \left(\frac{1}{\beta_k!}\partial_{\beta_k} f(x)\right)^{\lambda_k}$$

where ∂_λ means $\partial_{\sum_{u=1}^s \lambda_u}$ and where $\mathscr{D}_\alpha$ is the list of decompositions *of α. A multi-index $\alpha \in \mathbb{N}^d$ is decomposed into $s \in \mathbb{N}$ parts $\beta_1, \dots, \beta_s \in \mathbb{N}^d$ with multiplicies $\lambda_1, \dots, \lambda_s \in \mathbb{N}^n$ respectively if $|\lambda_i| > 0$ for all i, all the β_i are distincts from each other and from* 0, *and $\alpha = |\lambda_1|\beta_1 + \cdots + |\lambda_s|\beta_s$. Note that β and λ are multi-indices of multi-indices: $\beta \in \left(\mathbb{N}^d\right)^s$ and $\lambda \in \left(\mathbb{N}^d\right)^s$.*

[a] More precisely, for the formula to hold for α, all the derivatives which appear in the right-hand side must exist and be continuous

Proof: See [**?**] or [**?**]. □

We have seen that one-dimensional GPAC generable functions are analytic. We now extend this result to the multidimensional case.

Proposition 13.4 (Generable implies analytic) *If $f \in$ GVAL then f is real-analytic on* dom f.

Proof: Let $\mathtt{sp} : \mathbb{R} \to \mathbb{R}_+$, $p \in M_{n,d}[\mathbb{R}^n]$ and $y : \mathbb{R}^n \to \mathbb{R}^n$ from Definition 13.3. It is sufficient to prove that y is analytic on $D = \operatorname{dom} f$ to get the result. Let $i \in [\![1, n]\!]$, and $j \in [\![1, d]\!]$, since $J_y = p(y)$ then $\partial_j y_i(x) = p_{ij}(y(x))$ and p_{ij} is a polynomial vector so clearly C^∞. By Remark 13.5, y is also C^∞ so we can apply Theorem 13.2 for any $x \in D$, $\alpha \in \mathbb{N}^d$ and get

$$\partial_\alpha(\partial_j y_i)(x) = \partial_\alpha(p_{ij} \circ y)(x) = \alpha! \sum_{(s,\beta,\lambda)\in\mathscr{D}_\alpha} \partial_\lambda p_{ij}(y(x)) \prod_{k=1}^{s} \frac{1}{\lambda_k!} \left(\frac{1}{\beta_k!}\partial_{\beta_k} y(x)\right)^{\lambda_k}$$

Define $B_\alpha(x) = \frac{1}{\alpha!}\left\|\partial_\alpha y(x)\right\|$, and denote by $\alpha + j$ the multi-index λ such that $\lambda_j = \alpha_j + 1$ and $\lambda_k = \alpha_k$ for $k \neq j$. Define $C(y(x)) = \max_{i,j,\lambda}(|\partial_\lambda p_{ij}(y(x))|)$ and note that it is well-defined because $\partial_\lambda p_{ij}$ is zero whenever $|\lambda| > \deg(p_{ij})$. Define $\mathscr{D}'_\alpha = \{(s,\beta,\lambda) \in \mathscr{D}_\alpha \,|\, |\lambda| \leqslant \deg(p)\}$. The equations becomes:

$$\begin{aligned}
|\partial_\alpha(\partial_j y_i)(x)| &\leqslant \alpha! \sum_{(s,\beta,\lambda)\in\mathscr{D}_\alpha} |\partial_\lambda p_{ij}(y(x))| \prod_{k=1}^{s} \frac{1}{\lambda_k!} \left|\frac{1}{\beta_k!}\partial_{\beta_k} y(x)\right|^{\lambda_k} \\
&\leqslant \alpha! C(y(x)) \sum_{(s,\beta,\lambda)\in\mathscr{D}'_\alpha} \prod_{k=1}^{s} \frac{1}{\lambda_k!} B_{\beta_k}(x)^{|\lambda_k|}.
\end{aligned}$$

Note that the right-hand side of the expression does not depend on i. We are going to show by induction that $B_\alpha(x) \leqslant \left(\frac{C(y(x))}{R}\right)^{|\alpha|}$ for some choice of R. The initialization for $|\alpha| = 1$ is trivial because $\alpha! = 1$ and $B_\alpha(x) = \left\|\partial_\alpha y(x)\right\| \leqslant C(y(x))$ so we only need $R \leqslant 1$. The induction step is as follows:

$$
\begin{aligned}
B_{\alpha+j}(x) &\leqslant C(y(x)) \sum_{(s,\beta,\lambda)\in\mathscr{D}'_\alpha} \prod_{k=1}^{s} \frac{1}{\lambda_k!} B_{\beta_k}(x)^{|\lambda_k|} \\
&\leqslant C(y(x)) \sum_{(s,\beta,\lambda)\in\mathscr{D}'_\alpha} \prod_{k=1}^{s} \frac{1}{\lambda_k!} \left(\frac{C(y(x))}{R}\right)^{|\beta_k||\lambda_k|} \\
&\leqslant C(y(x)) \sum_{(s,\beta,\lambda)\in\mathscr{D}'_\alpha} \frac{1}{\lambda!} \left(\frac{C(y(x))}{R}\right)^{\sum_{u=1}^{s}|\beta_k||\lambda_k|} \\
&\leqslant C(y(x)) \left(\frac{C(y(x))}{R}\right)^{|\alpha|} \sum_{(s,\beta,\lambda)\in\mathscr{D}'_\alpha} \frac{1}{\lambda!} \\
&\leqslant C(y(x)) \left(\frac{C(y(x))}{R}\right)^{|\alpha|} \#\mathscr{D}'_\alpha.
\end{aligned}
$$

Evaluating the exact cardinal of $\mathscr{D}'_\alpha$ is complicated but we only need a good enough bound to get on with it. First notice that for any $(s,\beta,\lambda) \in \mathscr{D}'_\alpha$, we have $|\lambda| \leqslant \deg(p)$ by definition, and since each $|\lambda_i| > 0$, necessarily $s \leqslant \deg(p)$. This means that there is a finite number, denote it by A, of (s,λ) in $\mathscr{D}'_\alpha$. For a given λ, we must have $\alpha = \sum_{i=1}^{s} |\lambda_i|\beta_i$ which implies that $|\beta_{ij}| \leqslant |\alpha|$ and so there at most $(1+|\alpha|)^{ns}$ choices for β, and since $s \leqslant \deg(p)$, $\#\mathscr{D}'_\alpha \leqslant A(1+|\alpha|)^b$ where b and A are constants. Choose $R \leqslant 1$ such that $R^{|\alpha|} \geqslant A(1+|\alpha|)^b$ for all α to get the claimed bound on $B_\alpha(x)$.

To conclude with Proposition 13.3, consider $x \in D$. Let V be an open ball of D containing x. Let $M = \sup_{u\in V} C(y(x))$, it is finite because C is bounded by a polynomial, $\left\|y(x)\right\| \leqslant \mathtt{sp}(x)$ and V is an open ball (thus included in a compact set). Finally we get:

$$
\left\|\partial_\alpha y(x)\right\| \leqslant \alpha! \left(\frac{M}{R}\right)^{|\alpha|}
$$

□

13.5 Generable zoo

In this section, we introduce a number of generable functions. Since a GPAC (PIVP) only generates analytic functions, it cannot generate discontinuous functions like the sign. However these functions can be arbitrarily approximated by GPACs, as we show in this section, where we present a "zoo" of such approximating functions. This zoo illustrates the wide range of generable functions. Some of the functions selected in this "zoo" were chosen to approximate noncontinuous functions traditionally used in computer programs like the absolute value or the sign function. Other functions were selected due to their usefulness for potential applications, like simulating Turing machines with a GPAC, using a bounded amount of resources, which we intend to explore in an incoming paper.

We note that the approximation of a discontinuous functions by a GPAC generable function is uniform, since we provide the GPAC with a parameter which sets the maximum allowed error of the approximation. The use of different values of the parameter by the same GPAC allows to dynamically change the quality of the approximation, without making any other change on the GPAC. The table below gives a list of the functions and their purpose.

We use the term "dead zone" to refer to interval(s) where the generable function does not compute the expected function (but still has controlled behavior). We use the term "high" to mean that the function is close to x (an input) within $e^{-\mu}$ where μ is another input. Conversely, the use the term "low" to mean that it is close to 0 within $e^{-\mu}$. And "X" means something in between. Finally "integral" means that function is of the form ϕx and the integral of ϕ (on some interval) is between 1 and a constant.

We conclude this section by giving a large class of functions that can be uniformly approximated by (polynomially bounded) generable functions, except on a small number of dead zones (typically at discontinuity points) that can be made arbitrary small, see Section 13.6.

Generable Zoo

Name	Notation	Comment
Sign	$\mathrm{sg}(x,\mu,\lambda)$	Compute the sign of x with error $e^{-\mu}$ and dead zone in $[-\lambda^{-1},\lambda^{-1}]$. See 13.4
Floor	$\mathrm{ip}_1(x,\mu,\lambda)$	Compute $\mathrm{int}_1(x)$ with error $e^{-\mu}$ and dead zone in $[-\lambda^{-1},\lambda^{-1}]$. See 13.5
Abs	$\mathrm{abs}(x,\mu,\lambda)$	Compute $\lvert x\rvert$ with error with error $e^{-\mu}$ and dead zone in $[-\lambda^{-1},\lambda^{-1}]$. See 13.7
Max	$\mathrm{mx}(x,y,\mu,\lambda)$	Compute $\max(x,y)$ and $\lVert x\rVert$ with error $e^{-\mu}$ and dead zone for $x-y\in[-\lambda^{-1},\lambda^{-1}]$. See 13.8
Norm	$\mathrm{norm}\,\delta(x,\mu,\lambda)$	Compute $\lVert x\rVert$ with error δ. See 13.9
Round	$\mathrm{rnd}(x,\mu,\lambda)$	Compute $\lfloor x\rceil$ with error $e^{-\mu}$ and dead zones in $[n-\frac{1}{2}+\lambda^{-1},n+\frac{1}{2}-\lambda^{-1}]$ for all $n\in\mathbb{Z}$. See 13.6
Low-X-High	$\mathrm{lxh}_{[a,b]}(t,\mu,x)$	Compute 0 when $t\in]-\infty,a]$ and x when $t\in[b,\infty[$ with error $e^{-\mu}$ and a dead zone in $[a,b]$. See 13.10
High-X-Low	$\mathrm{hxl}_{[a,b]}(t,\mu,x)$	Compute x when $t\in]-\infty,a]$ and 0 when $t\in[b,\infty[$ with error $e^{-\mu}$ and a dead zone in $[a,b]$. See 13.10

13.5.1 Sign and rounding

We begin with a small result on the hyperbolic tangent function, which will be used to build several generable functions of interest.

Lemma 13.3 (Bounds on tanh**)** $1-\mathrm{sgn}(t)\tanh(t)\leqslant e^{-|t|}$ *for all* $t\in\mathbb{R}$.

Proof: The case of $t = 0$ is trivial. Assume that $t \geqslant 0$ and observe that $1-\tanh(t) = 1-\frac{1-e^{-2t}}{1+e^{-2t}} = \frac{2e^{-2t}}{1+e^{-2t}} = e^{-t}\frac{2e^{-t}}{1+e^{-2t}}$. Define $f(t) = \frac{2e^{-t}}{1+e^{-2t}}$ and check that $f'(t) = \frac{2e^{-t}(e^{-2t}-1)}{(1+e^{-2t})^2} \leqslant 0$ for $t \geqslant 0$. Thus f is a non-increasing function and $f(0) = 1$ which concludes.

If $t < 0$ then note that $1-\operatorname{sgn}(t)\tanh(t) = 1-\operatorname{sgn}(-t)\tanh(-t)$ so we can apply the result to $-t \geqslant 0$ to conclude. □

The simplest generable function of interest uses the hyperbolic tangent to approximate the sign function. On top of the sign function, we can build an approximation of the floor function. See Figure 13.5 for a graphical representation.

Definition 13.4 (Sign function) *For any $x, \mu, \lambda \in \mathbb{R}$ define*

$$\operatorname{sg}(x,\mu,\lambda) = \tanh(x\mu\lambda)$$

Lemma 13.4 (Sign) $\operatorname{sg} \in \mathrm{GVAL}[\mathrm{poly}]$ *and for any $x \in \mathbb{R}$ and $\lambda, \mu \geqslant 0$,*

$$|\operatorname{sgn}(x) - \operatorname{sg}(x,\mu,\lambda)| \leqslant e^{-|x|\lambda\mu} \leqslant 1$$

In particular, sg *is non-decreasing in x and if $|x| \geqslant \lambda^{-1}$ then*

$$|\operatorname{sgn}(x) - \operatorname{sg}(x,\mu,\lambda)| \leqslant e^{-\mu}$$

Proof: Note that $\operatorname{sg} = \tanh \circ f$ where $f(x,\mu,\lambda) = x\mu\lambda$. We saw in Example 13.2 that $\tanh \in \mathrm{GVAL}[t \mapsto 1]$. By Lemma 13.2, $f \in \mathrm{GVAL}[\alpha \mapsto \max(1,\alpha^3)]$. Thus $\operatorname{sg} \in \mathrm{GVAL}[\alpha \mapsto \max(1,\alpha^3)]$.

Use Lemma 13.3 and the fact that tanh is an odd function to get the first bound. The second bound derives easily from the first. Finally, sg is a non-decreasing function because tanh is an increasing function. □

Definition 13.5 (Floor function) *For any $x, \mu, \lambda \in \mathbb{R}$ define*

$$\operatorname{ip}_1(x,\mu,\lambda) = \frac{1+\operatorname{sg}(x-1,\mu,\lambda)}{2}$$

Lemma 13.5 (Floor) $\operatorname{ip}_1 \in \mathrm{GVAL}[\mathrm{poly}]$ *and for any $x \in \mathbb{R}$ and $\mu, \lambda \geqslant 0$,*

$$|\operatorname{int}_1(x) - \operatorname{ip}_1(x,\mu,\lambda)| \leqslant \frac{e^{-|x-1|\lambda\mu}}{2} \leqslant \frac{1}{2}$$

where $\operatorname{int}_1(x) = 0$ if $x < 1$ and 1 if $x \geqslant 1$. In particular ip_1 is non-decreasing in x and if $|1-x| \geqslant \lambda^{-1}$ then

$$|\operatorname{int}_1(x) - \operatorname{ip}_1(x,\mu,\lambda)| < e^{-\mu}$$

We will now see how to build a very precise approximation of the rounding function. Of course rounding is not a continuous operation so we need a small deadzone around the discontinuity points.

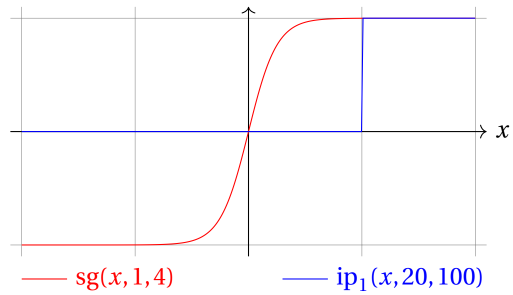


Figure 13.5: Graph of sg and ip_1.

Definition 13.6 (Round function) *For any* $x \in \mathbb{R}$, $\lambda \geqslant 2$ *and* $\mu \geqslant 0$, *define*

$$\mathrm{rnd}(x,\mu,\lambda) = x - \frac{1}{\pi}\arctan(\mathrm{cltan}(\pi x,\mu,\lambda))$$

$$\mathrm{cltan}(\theta,\mu,\lambda) = \frac{\sin(\theta)}{\sqrt{\mathrm{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}}\mathrm{sg}(\cos\theta,\mu+3\lambda,2\lambda)$$

$$\mathrm{nz}(x,\mu,\lambda) = x + \frac{2}{\lambda}\mathrm{ip}_1\left(1-x+\frac{3}{4\lambda},\mu+1,4\lambda\right)$$

Lemma 13.6 (Round) *For any* $n \in \mathbb{Z}$, $\lambda \geqslant 2$, $\mu \geqslant 0$, *we have* $|\mathrm{rnd}(x,\mu,\lambda) - n| \leqslant \frac{1}{2}$ *for all* $x \in \left[n-\frac{1}{2}, n+\frac{1}{2}\right]$ *and* $|\mathrm{rnd}(x,\mu,\lambda)-n| \leqslant e^{-\mu}$ *for all* $x \in \left[n-\frac{1}{2}+\frac{1}{\lambda}, n+\frac{1}{2}-\frac{1}{\lambda}\right]$. *Furthermore* $\mathrm{rnd} \in \mathrm{GVAL}[\mathrm{poly}]$.

Proof: Let's start with the intuition first: consider $f(x) = x - \frac{1}{\pi}\arctan(\tan(\pi x))$. It is an exact rounding function: if $x = n+\delta$ with $n \in \mathbb{N}$ and $\delta \in]\frac{-1}{2},\frac{1}{2}[$ then $\tan(\pi x) = \tan(\pi\delta)$ and since $\delta\pi \in]\frac{-\pi}{2},\frac{\pi}{2}[$, $f(x) = x - \delta = n$. The problem is that it is undefined on all points of the form $n + \frac{1}{2}$ because of the tangent function.

The idea is to replace $\tan(\pi x)$ by some "clamped" tangent cltan which will be like $\tan(\pi x)$ around integer points and stay bounded when close to $x = n + \frac{1}{2}$ instead of exploding. To do so, we use the fact that $\tan\theta = \frac{\sin\theta}{\cos\theta}$ but this formula is problematic because we cannot prevent the cosine from being zero, without loosing the sign of the expression (the cosine could never change sign). Thus the idea is to remove the sign from the cosine, and restore it, so that $\tan\theta = \mathrm{sgn}(\cos\theta)\frac{\sin\theta}{|\cos\theta|}$. And now we can replace $|\cos(\theta)|$ by $\sqrt{\mathrm{nz}(\cos^2\theta)}$, where $\mathrm{nz}(x)$ is mostly x except near 0 where is lower-bounded by some small constant (so it is never zero). The sign of cosine can be computed using our approximate sign function sg.

Formally, we begin with nz and show that:

- $\mathrm{nz} \in \mathrm{GVAL}[\mathrm{poly}]$

- nz is an increasing function of x

- For $x \geqslant \frac{1}{\lambda}$, $|\operatorname{nz}(x,\mu,\lambda) - x| \leqslant e^{-\mu}$

- For $x \geqslant 0$, $\operatorname{nz}(x,\mu,\lambda) \geqslant \frac{1}{2\lambda}$

The first point is a consequence of $\operatorname{ip}_1 \in \text{GVAL}[\text{poly}]$ from Corollary 13.5. The second point comes from Corollary 13.5: if $x \geqslant \frac{1}{\lambda}$, then $1 - x + \frac{3}{4\lambda} \leqslant 1 - \frac{1}{4\lambda}$, thus $|\operatorname{nz}(x,\mu,\lambda) - x| \leqslant \frac{2}{\lambda}e^{-\mu-1} \leqslant e^{-\mu}$ since $\lambda \geqslant 2$. To show the last point, first apply Corollary 13.5: if $x \leqslant \frac{1}{2\lambda}$, then $1 - x + \frac{3}{4\lambda} \geqslant 1 + \frac{1}{4\lambda}$, thus $|\operatorname{nz}(x,\mu,\lambda) - x - \frac{2}{\lambda}| \leqslant \frac{2}{\lambda}e^{-\mu-1}$ Thus $\operatorname{nz}(x,\mu,\lambda) \geqslant \frac{2}{\lambda}(1 - e^{-\mu-1}) + x \geqslant \frac{1}{\lambda}$ since $1 - e^{-\mu-1} \leqslant \frac{1}{2}$ and $x \geqslant 0$. And for $x \geqslant \frac{1}{2\lambda}$, by Corollary 13.5 we get that $\operatorname{nz}(x,\mu,\lambda) \geqslant x \geqslant \frac{1}{2\lambda}$ which shows the last point.

Then we show that:

- $\operatorname{cltan} \in \text{GVAL}[\text{poly}]$, is π-periodic and is an odd function.

- For $\theta \in \left[-\frac{\pi}{2} + \frac{1}{\lambda}, \frac{\pi}{2} - \frac{1}{\lambda}\right]$, $|\operatorname{cltan}(\theta,\mu,\lambda) - \tan(\theta)| \leqslant e^{-\mu}$

First apply the above results to get that $\operatorname{nz}(\cos^2\theta, \mu + 16\lambda^3, 4\lambda^2) \geqslant \frac{1}{8\lambda^2}$. It follows that $\operatorname{cltan}(\theta,\mu,\lambda) \leqslant \frac{1}{\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}} \leqslant \sqrt{8}\lambda$, which is a polynomial in λ. Since $\sin, \cos, \operatorname{sg}, \operatorname{nz} \in \text{GVAL}[\text{poly}]$, it follows that $\operatorname{clan} \in \text{GVAL}[\text{poly}]$. The periodicity comes from the properties of sine and cosine, and the fact that sg is an odd function. It is an odd function for similar reasons. To show the second point, since it is periodic and odd, we can assume that $\theta \in \left[0, \frac{\pi}{2} - \frac{1}{\lambda}\right]$. For such a θ, we have that $\frac{\pi}{2} - \theta \geqslant \frac{1}{\lambda}$, thus $\cos(\theta) \geqslant \sin(\frac{\pi}{2} - \theta) \geqslant \frac{1}{2\lambda}$ (use that $\sin(u) \geqslant \frac{u}{2}$ for $0 \leqslant u \leqslant \frac{\pi}{2}$). By Lemma 13.4 we get that $|\operatorname{sg}(\cos\theta, \mu + 3\lambda, 2\lambda) - 1| \leqslant e^{-\mu-3\lambda}$. Also $\cos^2\theta \geqslant \frac{1}{4\lambda^2}$ thus by the above results we get that $|\operatorname{nz}(\cos^2\theta, \mu + 16\lambda^3, 4\lambda^2) - \cos^2\theta| \leqslant e^{-\mu}$. Using the fact that $|\frac{\sqrt{a}-\sqrt{b}}{\sqrt{a}}| \leqslant |a - b|$ for any $a > 0$ and $b \in \mathbb{R}$, we get that $\left|\frac{\sqrt{\operatorname{nz}(\cos^2\theta,\mu,4\lambda^2)} - |\cos\theta|}{\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,2\lambda)}}\right| \leqslant |\operatorname{nz}(\cos^2\theta, \mu + 16\lambda^3, 4\lambda^2) - \cos^2\theta| \leqslant \sqrt{8}\lambda e^{-\mu-16\lambda^3}$. Putting everything together, using that $\cos\theta \geqslant$

$\frac{1}{2\lambda}$ and $\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,2\lambda)\geqslant 8\lambda^2$, we get that

$$\begin{aligned}
|\operatorname{cltan}(\theta,\mu,\lambda)-\tan\theta| &= \left|\frac{\sin(\theta)\operatorname{sg}(\cos\theta,\mu+3\lambda,2\lambda)}{\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}}-\frac{\sin\theta}{\cos\theta}\right|\\
&\leqslant \left|\frac{\sin(\theta)(\operatorname{sg}(\cos\theta,\mu+3\lambda,2\lambda)-\operatorname{sgn}(\cos\theta)}{\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}}\right|\\
&\quad+\left|\frac{\sin(\theta)\operatorname{sgn}(\cos\theta)}{\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}}-\frac{\sin\theta}{\cos\theta}\right|\\
&\leqslant \frac{|\operatorname{sg}(\cos\theta,\mu+3\lambda,2\lambda)-\operatorname{sgn}(\cos\theta)|}{\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}}\\
&\quad+\left|\frac{1}{\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}}-\frac{1}{|\cos\theta|}\right|\\
&\leqslant \sqrt{8}\lambda e^{-\mu-3\lambda}+\frac{|\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}-|\cos\theta||}{|\cos\theta|\|\sqrt{\operatorname{nz}(\cos^2\theta,\mu+16\lambda^3,4\lambda^2)}}\\
&\leqslant \sqrt{8}\lambda e^{-\mu-3\lambda}+2\lambda\cdot\sqrt{8}\lambda\cdot\sqrt{8}\lambda e^{-\mu-16\lambda^3}\\
&\leqslant 3\lambda e^{-\mu-3\lambda}+16\lambda^3 e^{-\mu-16\lambda^3}\\
&\leqslant e^{-\mu}
\end{aligned}$$

because $xe^{-x}\leqslant\frac{1}{2}$ for any $x\geqslant 0$.

Let $n\in\mathbb{N}$ and $x=n+\delta\in[n-\frac{1}{2},n+\frac{1}{2}]$. Since cltan is π-periodic, $\operatorname{rnd}(x,\mu,\lambda)=n+\delta-\frac{1}{\pi}\arctan(\operatorname{cltan}(\pi\delta,\mu,\lambda))$. Furthermore $\pi\delta\in[-\frac{\pi}{2},\frac{\pi}{2}]$ so $\cos(\pi\delta)\geqslant 0$ and $\operatorname{sgn}(\sin(\pi\delta))=\operatorname{sgn}(\delta)$. Consequently, $\operatorname{sg}(\cos(\pi\delta),\mu+3\lambda,2\lambda)\in[0,1]$ by definition of sg and $\sqrt{\operatorname{nz}(\cos^2(\pi\delta),\mu+16\lambda^3,4\lambda^2)}>\sqrt{\cos^2(\pi\delta)}$ because $\operatorname{ip}_1>0$. Consequently, we get that $|\operatorname{cltan}(\pi\delta,\mu,\lambda)|\leqslant\frac{|\sin(\pi\delta)|}{\cos(\pi\delta)}$ and $\operatorname{sgn}(\operatorname{cltan}(\pi\delta,\mu,\lambda))=\operatorname{sgn}(\delta)$. Finally, we can write $\frac{1}{\pi}\arctan(\operatorname{cltan}(\pi\delta,\mu,\lambda))=\alpha$ with $|\alpha|\leqslant|\frac{1}{\pi}\arctan(\tan(\pi\delta))|\leqslant|\delta|$ and $\operatorname{sgn}(\alpha)=\operatorname{sgn}(\delta)$ which shows that $|\operatorname{rnd}(x,\mu,\lambda)-n|\leqslant\delta\leqslant\frac{1}{2}$.

Finally we can show the result about rnd: since cltan and tan are in GVAL[poly], then rnd $\in$ GVAL[poly]. Now consider $x\in\left[n-\frac{1}{2}+\frac{1}{\lambda},n+\frac{1}{2}-\frac{1}{\lambda}\right]$, and let $\theta=\pi x-\pi n$. Then $\theta\in\left[-\frac{\pi}{2}+\frac{\pi}{\lambda},\frac{\pi}{2}-\frac{\pi}{\lambda}\right]\subseteq\left[-\frac{\pi}{2}+\frac{1}{\lambda},\frac{\pi}{2}-\frac{1}{\lambda}\right]$, and since cltan is periodic, then $\operatorname{rnd}(x,\mu,\lambda)=n+\frac{\theta}{\pi}-\frac{1}{\pi}\arctan(\operatorname{cltan}(\theta,\mu,\lambda)$. Finally, using the results about cltan yields: $|\operatorname{rnd}(x,\mu,\lambda)-n|=\frac{1}{\pi}|\theta-\arctan(\operatorname{cltan}(\theta,\mu,\lambda)|=\frac{1}{\pi}|\arctan(\tan(\theta))-\arctan(\operatorname{cltan}(\theta,\mu,\lambda)|\leqslant\frac{1}{\pi}|\tan(\theta)-\operatorname{cltan}(\theta,\mu,\lambda)|\leqslant\frac{e^{-\mu}}{\pi}\leqslant e^{-\mu}$ since arctan is a 1-Lipschitz function. □

13.5.2 Absolute value, maximum and norm

A very common operation is to compute the absolute value of a number. Of course this operation is not generable because it is not even differentiable. However, a good enough approximation can be built. In particular, this approximation has several key features: it is non-negative and it is an over-approximation. We can then use it to build an approximation of the max function and the infinite norm.

Definition 13.7 (Absolute value function) *For any $x \in \mathbb{R}$ and $\mu, \lambda > 0$ define:*

$$\operatorname{abs}(x,\mu,\lambda) = \frac{1}{1+\lambda\mu}\ln(2\cosh((1+\lambda\mu)x))$$

Lemma 13.7 (Absolute value) *For any $x \in \mathbb{R}$ and $\mu, \lambda > 0$ we have*

$$|x| \leqslant \operatorname{abs}(x,\mu,\lambda) \leqslant |x| + \min\left(\frac{1}{1+\lambda\mu}, e^{-|x|\lambda\mu}\right).$$

So in particular, if $|x| \geqslant \lambda^{-1}$ then $|x| \leqslant \operatorname{abs}(x,\mu,\lambda) \leqslant |x| + e^{-\mu}$. Furthermore $\operatorname{abs} \in \mathrm{GVAL}[\mathrm{poly}]$ *and is an even function.*

Proof: Since cosh is an even function, we immediately get that abs is even. Let $x \geqslant 0$ and $\mu, \lambda > 0$. Since $2\cosh(u) \geqslant e^u$, it trivially follows that $\operatorname{abs}(x,\mu,\lambda) \geqslant \frac{1}{1+\lambda\mu}(1+\lambda\mu)x \geqslant x$. Also $\ln(2\cosh(u)) = \ln(e^u(1+e^{-2u})) = u + \ln(1+e^{-2u}) \leqslant u + e^{-2u}$ so it follows that $\operatorname{abs}(x,\mu,\lambda) \leqslant x + \frac{1}{1+\lambda\mu}e^{-2(1+\lambda\mu)x} \leqslant x + e^{-x\lambda\mu}$. Furthermore, $\frac{\partial \operatorname{abs}}{\partial x}(x,\mu,\lambda) = \tanh((1+\lambda\mu)x)$ which shows that $x \mapsto \operatorname{abs}(x,\mu,\lambda) - x$ is decreasing and positive over $[0,+\infty[$ and thus has its maximum $\operatorname{abs}(0,\mu,\lambda) = \frac{1}{1+\mu\lambda}$ attained at 0. Since $\left(\ln(2\cosh(u))\right)' = \tanh(u)$, $\tanh \in \mathrm{GVAL}[\mathrm{poly}]$ and $\ln(2\cosh(u))$ is bounded by $|u|+1$, we get that $\left(u \mapsto \ln(2\cosh(u))\right) \in \mathrm{GVAL}[\mathrm{poly}]$ by applying Corollary 13.1. It follows that $\operatorname{abs} \in \mathrm{GVAL}[\mathrm{poly}]$ using the usual lemmas. □

Definition 13.8 (Max/Min function) *For any $x, y \in \mathbb{R}$ and $\mu, \lambda > 0$ define:*

$$\operatorname{mx}(x,y,\mu,\lambda) = \frac{y + x + \operatorname{abs}(y-x,\mu,\lambda)}{2} \qquad \operatorname{mn}(x,y,\mu,\lambda) = x + y - \operatorname{mx}(x,y,\mu,\lambda).$$

For any $x \in \mathbb{R}^n$ and $\delta \in]0,1]$ define:

$$\operatorname{mx}_\delta(x) = \operatorname{mx}(x_1, \operatorname{mx}(\ldots, \operatorname{mx}(x_{n-1}, x_n, 1, (n\delta)^{-1})\ldots)).$$

Lemma 13.8 (Max/Min function) *For any $x, y \in \mathbb{R}$ and $\lambda, \mu > 0$ we have:*

$$\max(x,y) \leqslant \operatorname{mx}(x,y,\mu,\lambda) \leqslant \max(x,y) + \min\left(\frac{1}{1+\lambda\mu}, e^{-|x-y|\lambda\mu}\right)$$

and

$$\min(x,y) - \min\left(\frac{1}{1+\lambda\mu}, e^{-|x-y|\lambda\mu}\right) \leqslant \operatorname{mn}(x,y,\mu,\lambda) \leqslant \min(x,y)$$

So in particular, if $|x-y| \geqslant \lambda^{-1}$ then $\max(x,y) \leqslant \operatorname{mx}(x,y,\mu,\lambda) \leqslant \max(x,y) + e^{-\mu}$ *and* $\min(x,y) - e^{-\mu} \leqslant \operatorname{mn}(x,y,\mu,\lambda) \leqslant \min(x,y)$. *Furthermore* $\operatorname{mx}, \operatorname{mn} \in \mathrm{GVAL}[\mathrm{poly}]$. *For any $x \in \mathbb{R}^n$ and $\delta \in]0,1]$ we have:*

$$\max(x_1,\ldots,x_n) \leqslant \operatorname{mx}_\delta(x) \leqslant \max(x_1,\ldots,x_n) + \delta$$

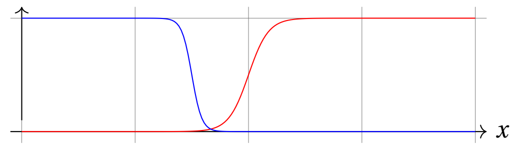


Figure 13.6: Graph of $\mathrm{lxh}_{[1,3]}$ and $\mathrm{hxl}_{[1,2]}$

Furthermore $\mathrm{mx}_\delta \in \mathrm{GVAL}[\mathrm{poly}]$.

Proof: By Lemma 13.7, $|y-x| \leqslant \mathrm{abs}(y-x,\mu,\lambda) \leqslant |y-x| + \min\left(\frac{1}{1+\lambda\mu}, e^{-|x-y|\lambda\mu}\right)$ and the result follows because $\max(x,y) = \frac{y+x+|y-x|}{2}$. The result on mn follows the one on mx. Finally mx, mn $\in \mathrm{GVAL}[\mathrm{poly}]$ from Lemma 13.2.

Observe that $\max(x) \leqslant \mathrm{mx}_\delta(x)$ is trivial by definition. The other inequality is a simple calculus based on $\max(x,y,\mu,\lambda) \leqslant \max(x,y) + \frac{1}{1+\mu\lambda}$:

$$\mathrm{mx}_\delta(x) \leqslant \max(x) + n\frac{1}{1+(n\delta)^{-1}} \leqslant \max(x) + \delta.$$

Note that strictly speaking, for $\mathrm{mx}_\delta \in \mathrm{GVAL}_{\mathbb{K}}[\mathrm{poly}]$ we need that $\delta \in \mathbb{K}$ or use a smaller δ' in $\mathbb{K}$ which is always possible. □

Definition 13.9 (Norm function) *For any* $x \in \mathbb{R}^n$ *and* $\delta \in]0,1]$ *define:*

$$\mathrm{norm}\infty,\delta(x) = \mathrm{mx}_{\delta/2}(\mathrm{abs}_{\delta/2}(x_1),\ldots,\mathrm{abs}_{\delta/2}(x_n))$$

where $\mathrm{abs}_\delta(x) = \mathrm{mx}_\delta(x,-x)$.

Lemma 13.9 (Norm function) *For any* $x \in \mathbb{R}^n$ *and* $\delta \in]0,1]$ *we have:*

$$\|x\| \leqslant \mathrm{norm}\infty,\delta(x) \leqslant \|x\| + \delta$$

Furthermore $\mathrm{norm}\infty,\delta \in \mathrm{GVAL}[\mathrm{poly}]$.

Proof: Apply Lemma 13.7 and Lemma 13.8. □

13.5.3 Switching functions

An important construct in digital computation is the "if ... then ... else ..." construct, which allows us to switch between two different behaviours. Again, this cannot be done exactly with a GPAC since GPACs cannot generate discrete functions and we need something which acts like a select function, which can pick between two values depending on how a third value compares to a threshold. The problem is that this operation is not continuous, and thus not generable. But such a select function can be approximated by a GPAC. As a good first step, we build so-called "low-X-high" and "high-X-low" functions which act as a switch between 0 (low) and a value (high). Around the threshold will be an small uncertainty zone (X) where the exact value cannot be predicted. See Figure 13.6 for a graphical representation.

Definition 13.10 ("low-X-high" and "high-X-low") *Let* $I=[a,b]$ *with* $b>a$, $t\in\mathbb{R}$, $\mu\in\mathbb{R}$, $x\in\mathbb{R}$, $\nu=\mu+\ln(1+x^2)$, $\delta=\frac{b-a}{2}$ *and define:*

$$\operatorname{lxh}_I(t,\mu,x)=\operatorname{ip}_1\left(t-\frac{a+b}{2}+1,\nu,\frac{1}{\delta}\right)x$$

$$\operatorname{hxl}_I(t,\mu,x)=\operatorname{ip}_1\left(\frac{a+b}{2}-t+1,\nu,\frac{1}{\delta}\right)x$$

Lemma 13.10 ("low-X-high" and "high-X-low") *Let* $I=[a,b]$, $\mu\in\mathbb{R}_+$, *then* $\forall t,x\in\mathbb{R}$*:*

- $\exists\phi_1,\phi_2$ *such that* $\operatorname{lxh}_I(t,\mu,x)=\phi_1(t,\mu,x)x$ *and* $\operatorname{hxl}_I(t,\mu,x)=\phi_2(t,\mu,x)x$
- *if* $t\leqslant a, |\operatorname{lxh}_I(t,\mu,x)|\leqslant e^{-\mu}$ *and* $|x-\operatorname{hxl}_I(t,\mu,x)|\leqslant e^{-\mu}$
- *if* $t\geqslant b, |x-\operatorname{lxh}_I(t,\mu,x)|\leqslant e^{-\mu}$ *and* $|\operatorname{hxl}_I(t,\mu,x)|\leqslant e^{-\mu}$
- *in all cases,* $|\operatorname{lxh}_I(t,\mu,x)|\leqslant|x|$ *and* $|\operatorname{hxl}_I(t,\mu,x)|\leqslant|x|$

Furthermore, $\operatorname{lxh}_I,\operatorname{hxl}_I\in\mathrm{GVAL}[\mathrm{poly}]$.

Proof: By symmetry, we only prove it for lxh. This is a direct consequence of Corollary 13.5 and the fact that $|x|\leqslant e^{\ln(1+x^2)}$. Indeed if $t\leqslant a$ then $t-\frac{a+b}{2}+1\leqslant 1-\delta$ thus $|\operatorname{lxh}_I(t,\nu,x)|\leqslant|x|e^{-\nu}\leqslant e^{-\mu}$. Similarly if $t\geqslant b$ then $t-\frac{a+b}{2}+1\geqslant 1+\delta$ and we get a similar result. Apply Lemma 13.2 multiple times to see that they are belong to GVAL[poly]. ☐

13.6 GPAC approximation

The examples of the previous section all share an interesting common pattern, which we formalise with the definition below. In this section, $\mathbb{K}$ can be any generable field[2].

Definition 13.11 (GPAC approximation) *Let* I *be an open and connected subset of* $\mathbb{R}^m$, $\Gamma\subseteq I$ *a subset of* I *of* exceptions *and* $f:I\to\mathbb{R}^m$. *We say that* f *is* GPAC-approximable over I but Γ *if there exists* $g\in\mathrm{GVAL}_{\mathbb{K}}[\mathrm{poly}]$ *such that for any* $x\in I$ *and* $\mu,\lambda>0$ *we have*

$$\left\|f(x)-g(x,\mu,\lambda)\right\|\leqslant e^{-\mu}\quad\text{if}\quad d(x,\Gamma)\geqslant\lambda^{-1},$$

[2] See Section 13.7 for more details.

where $d(x,\Gamma)$ denotes the distance between x and Γ (for the infinite norm).

The set Γ of points where the approximation fails will typically be discrete, finite or even empty. If Γ is empty, we do not mention it and say f is GPAC-approximable. Intuively, g provides an *effective*, uniform and arbitrary good approximation of f, except on a set that can be made "arbitrary small". We cannot quantify how small the set of exception is in general, since the definition allows for pathological cases such as $\Gamma = I$ or $\Gamma = I \cap \mathbb{Q}^m$. However, in case where Γ is discrete, a condition met by all examples in this paper, for any compact set K, the measure of exception set $\{d(x,\Gamma) \leqslant \lambda^{-1}\} \cap K$ converges to 0 as λ tends to infinity.

Note that our notion of approximation is not really related to classical approximation theory, by a sequence of functions for example. Indeed, in the definition, the same function g is used for all μ and λ, which creates a lot of constraints since g is generable, i.e. it satisfies a polynomial partial differential equation. Informally, one can think of g as a "template" with parameters μ and λ that we can tweak to get closer and closer to f but the shape itself of the template is fixed once and for all.

It appears that there is an interesting trade-off between the bound sp on the norm of g (i.e. $g \in \text{GVAL}[\texttt{sp}]$) and the quality of the approximation. Indeed, if sp is chosen to be a polynomial, we can seemingly achieve an exponential error bound ($e^{-\mu}$) but only an inverse distance from Γ ($1/\lambda$) for interesting functions. For simplicity, we only consider polynomially bounded generable functions is this definition.

Note that the definition does not mandate that f be continuous and indeed it needs not be. For example, Lemma 13.6 proves that the rounding function is GPAC-approximable over $\mathbb{R}$ but $\frac{1}{2} + \mathbb{Z}$. More generally, the discontinuity points will always belong to Γ.

In this section, we give several examples of classes of functions that can be approximated as described above.

Lemma 13.11 (Basic approximable functions) *Any generable function is approximable on its domain of definition. If f and g are GPAC-approximable over X but Γ_f and Γ_g respectively, then $f \pm g$ and fg are GPAC-approximable over X but $\Gamma_f \cup \Gamma_g$.*

Proof: Any generable function trivially satisfies the definition using itself as an approximation. If f is approximated by F and g by G then for any $\mu, \lambda > 0$ and $x \in X$ such that $d(x, \Gamma_f \cup \Gamma_g) \geqslant \lambda^{-1}$:

$$\left\| f(x) + g(x) - F(x,\mu+1,\lambda) - G(x,\mu+1,\lambda) \right\| \leqslant 2e^{-\mu-1} \leqslant e^{-\mu}.$$

Thus $(x,\mu,\lambda) \mapsto F(x,\mu+1,\lambda) + G(x,\mu+1,\lambda)$ approximate $f+g$ over X but $\Gamma_f \cup \Gamma_g$.

The case of the multiplication is similar but slightly more involved. Define for any $x \in X$ and $\mu, \lambda > 0$:

$$H(x,\mu,\lambda) =$$

$$\underbrace{F\big(x,\mu+2+\text{norm}\infty,1(G(x,1,\lambda)),\lambda\big)}_{:=\tilde{f}(x,\mu,\lambda)}\underbrace{G\big(x,\mu+3+\text{norm}\infty,1(F(x,1,\lambda)),\lambda\big)}_{:=\tilde{g}(x,\mu,\lambda)}.$$

It will be useful to recall that $\|x\| \leqslant \operatorname{norm}_{\infty,1}(x)$ thanks to Lemma 13.9. Let $\mu, \lambda > 0$ and $x \in X$ such that $d(x, \Gamma_f \cup \Gamma_g) \geqslant \lambda^{-1}$. Note that since we have $\left\|f(x) - F(x,1,\lambda)\right\| \leqslant e^{-1}$ then $\|F(x,1,\lambda)\| \geqslant \left\|f(x)\right\| - 1$. Similarly, $\left\|\tilde{g}(x,\mu,\lambda) - G(x,1,\lambda)\right\| \leqslant e^{-1} + e^{-\mu}$ thus $\|G(x,1,\lambda)\| \geqslant \left\|\tilde{g}(x,\mu,\lambda)\right\| - 2$. Finally check that $x \mapsto xe^{-x}$ is globally bounded by 1. Thus we have:

$$\begin{aligned}
\left\|f(x)g(x) - H(x,\mu,\lambda)\right\| &\leqslant \left\|f(x)\right\| \left\|g(x) - \tilde{g}(x,\mu,\lambda)\right\| \\
&\quad + \left\|f(x) - \tilde{f}(x,\mu,\lambda)\right\| \left\|\tilde{g}(x,\mu,\lambda)\right\| \\
&\leqslant \left\|f(x)\right\| e^{-\mu-2-\operatorname{norm}_{\infty,1}(F(x,1,\lambda))} \\
&\quad + e^{-\mu-3-\operatorname{norm}_{\infty,1}(G(x,1,\lambda))} \left\|\tilde{g}(x,\mu,\lambda)\right\| \\
&\leqslant \left\|f(x)\right\| e^{-\mu-1-\left\|f(x)\right\|} + e^{-\mu-1-\left\|\tilde{g}(x,\mu,\lambda)\right\|} \left\|\tilde{g}(x,\mu,\lambda)\right\| \\
&\leqslant 2e^{-\mu-1} \leqslant e^{-\mu}.
\end{aligned}$$

This shows that H approximates fg over x but $\Gamma_f \cup \Gamma_g$. The fact that $H \in \mathrm{GVAL}[\mathrm{poly}]$ follows from the hypothesis on F and G and Lemma 13.2. □

Theorem 13.3 (Piecewise approximability) *Let $-\infty \leqslant a_0 < a_1 < \ldots < a_{k+1} \leqslant +\infty$ and $f :]a_0, a_{k+1}[\to \mathbb{R}$. Assume that for each $i \in \{0, \ldots, k\}$, f is GPAC-approximable over $]a_i, a_{i+1}[$ but Γ_i. Further assume that all finite a_i belong to $\mathbb{K}$. Then f is GPAC-approximable over $]a_0, a_{k+1}[$ but $\{a_1, \ldots, a_k\} \cup \bigcup_{i=0}^{k} \Gamma_i$.*

Proof: Without loss of generality, we can assume that f is defined over $\mathbb{R}$. Indeed if f is only defined over $[a,b]$, $[a,+\infty[$ or $]-\infty,b]$, we can add an extra infinite interval over which f is constantly equal to 0. The resulting g for this extended f satisfies the definition over the original domain of definition of f.

We now assume that $a_0 = -\infty$ and $a_{k+1} = +\infty$. Let $\tilde{f}_i \in \mathrm{GVAL}[\mathrm{poly}]$ be the GPAC-approximation of f over $]a_i, a_{i+1}[$ but Γ_i, for $i \in \{0, \ldots, k\}$. There is a subtle issue at this point: a priori $\tilde{f}_i$ is only defined over $]a_i, a_{i+1}[\times]0,+\infty[^2$. We will show that $\tilde{f}_i$ can be assumed to be defined over $\mathbb{R}\times]0,+\infty[^2$ and we defer of proof of this fact to end of this proof. Define for any $x \in \mathbb{R}$, $\mu \geqslant 0$ and $\lambda > 0$:

$$g(x,\mu,\lambda) = \tilde{f}_0(x,\nu,\lambda) + \sum_{i=1}^{k} \mathrm{lxh}_{[-1,1]}\left((x-a_i)\lambda, \nu, \tilde{f}_i(x,\nu,\lambda) - \tilde{f}_{i-1}(x,\nu,\lambda)\right)$$

where $\nu = \mu + k + 1$. First note that $g \in \mathrm{GVAL}_{\mathbb{K}}[\mathrm{poly}]$ because it is a finite sum of generable functions in $\mathrm{GVAL}[\mathrm{poly}]$, and the endpoints of the intervals belong to $\mathbb{K}$. Define $\Gamma = \{a_1, \ldots, a_k\} \cup \bigcup_{i=0}^{k} \Gamma_i$. Let $\mu, \lambda > 0$ and $x \in \mathbb{R}$ be such that $d(x,\Gamma) \geqslant \lambda^{-1}$. It follows that $a_i + \lambda^{-1} \leqslant x \leqslant a_{i+1} - \lambda^{-1}$ for some $i \in \{0, \ldots, k\}$. Let $j \in \{0, \ldots, k\}$ and apply Lemma 13.10 to get that $|\mathrm{lxh}_{[-1,1]}((x-a_j)\lambda, \nu, X)| \leqslant e^{-\nu}$ if $j \geqslant i+1$ and $|\mathrm{lxh}_{[-1,1]}((x-$

$a_j)\lambda,\nu,X)-X|\leqslant e^{-\nu}$ if $j\leqslant i$. It follows that:

$$\begin{aligned}\left|g(x,\mu,\lambda)-f(x)\right| &\leqslant \left|g(x,\mu,\lambda)-\tilde{f}_i(x,\nu,\lambda)\right|+e^{-\nu}\\ &=\left|g(x,\mu,\lambda)-\tilde{f}_0(x,\nu,\lambda)-\sum_{j=1}^{i}\left(\tilde{f}_i(x,\nu,\lambda)-\tilde{f}_{i-1}(x,\nu,\lambda)\right)\right|\\ &\leqslant \sum_{j=1}^{i}\Big(\mathrm{lxh}_{[-1,1]}\left((x-a_i)\lambda,\nu,\tilde{f}_i(x,\nu,\lambda)-\tilde{f}_{i-1}(x,\nu,\lambda)\right)\\ &\qquad -\left(\tilde{f}_i(x,\nu,\lambda)-\tilde{f}_{i-1}(x,\nu,\lambda)\right)\Big)+e^{-\nu}\\ &\leqslant (k+1)e^{-\nu}\leqslant e^{-\mu}.\end{aligned}$$

This concludes the proof that f is approximate by g over $\mathbb{R}$ but Γ. It remains to show that, indeed, each $\tilde{f}_i$ can be assumed to be defined over $\mathbb{R}$. We show this in full-generality for intervals.

Let $f:]a,b[\to\mathbb{R}$ and $\tilde{f}:]a,b[\times]0,+\infty[^2$ a GPAC-approximation of f. Let $\mathtt{sp}$ be a polynomial such that $\tilde{f}\in\mathrm{GVAL}[\mathtt{sp}]$. Apply Proposition 13.2 to $\tilde{f}$ to get a polynomial q. Recall that q acts as a modulus of continuity:

$$\left|\tilde{f}(x,\mu,\lambda)-f(y,\mu,\lambda)\right|\leqslant|x-y|q(\mathtt{sp}(\max(|x|,|y|,\mu,\lambda)))$$

for any $x,y\in]a,b[$ and $\mu,\lambda>0$. Let $p\in\mathbb{K}[\mathbb{R}]$ be a nondecreasing polynomial such that $p(x)\geqslant q(\mathtt{sp}(x))$ for all $x\geqslant 0$. Define for any $x\in\mathbb{R}$ and $\mu,\lambda>0$:

$$\mathrm{clamp}(x,\mu,\lambda)=\mathrm{mx}(a+\theta^{-1},\mathrm{mn}(x,b-\theta^{-1},\mu+1,\theta),\mu+1,\theta)$$

where $\delta=b-a$ and $\theta=2\lambda+(2\delta)^{-1}$. Observe that clamp satisfies three key properties:

- $\mathrm{clamp}(x,\mu,\lambda)\in]a,b[$ for all $x\in\mathbb{R}$ and $\mu,\lambda>0$: indeed, by Lemma 13.8, $\mathrm{clamp}(x,\mu,\lambda)\geqslant a+\theta^{-1}>a$. On the other hand, $\mathrm{clamp}(x,\mu,\lambda)\leqslant\max(a+\theta^{-1},\mathrm{mn}(x,b,\mu+1,\theta))+\frac{1}{1+(1+\mu)\theta}$ but $\mathrm{mn}(\mathrm{mn}(x,b-\theta^{-1},\mu+1,\theta))\leqslant b-\theta^{-1}$ so $\mathrm{clamp}(x,\mu,\lambda)\leqslant\max(a+\theta^{-1},b-\theta^{-1})+\frac{1}{1+(1+\mu)\theta}$. Note that $\theta>(2\delta)^{-1}$ so $a+\theta^{-1}<b-\theta^{-1}$. Consequently $\mathrm{clamp}(x,\mu,\lambda)\leqslant b-\theta^{-1}+\frac{1}{1+(1+\mu)\theta}<b$.
- if $a+\lambda^{-1}\leqslant x\leqslant b-\lambda^{-1}$ then $|\mathrm{clamp}(x,\mu,\lambda)-x|\leqslant e^{-\mu}$: if $a+\lambda^{-1}\leqslant x$ then $x-(a+\theta^{-1})-\theta^{-1}\geqslant\lambda^{-1}-2\theta^{-1}\geqslant 0$ so $|\mathrm{clamp}(x,\mu,\lambda)-\mathrm{mn}(x,b-\theta^{-1},\mu+1,\theta)|\leqslant e^{-\mu-1}$. Similarly, $x\leqslant b-\lambda^{-1}$ implies that $x\leqslant(b-\theta^{-1})-\theta^{-1}$ so $|\mathrm{mn}(x,b-\theta^{-1},\mu+1,\theta)-x|\leqslant e^{-\mu-1}$. It follows that $|\mathrm{clamp}(x,\mu,\lambda)-x|\leqslant 2e^{-\mu-1}\leqslant e^{-\mu}$.
- $\mathrm{clamp}\in\mathrm{GVAL}[\mathrm{poly}]$: use Lemma 13.8 and the usual arithmetic lemmas. Note that it works because $\lambda\mapsto(2\lambda+(2\delta)^{-1})^{-1}$ belongs to GVAL[poly] for any fixed δ.

We can now use clamp to make sure the argument of $\tilde{f}$ is always within the domain of definition $]a,b[$, and make sure that it is a good enough approximation using the modulus of continuity. Define for any $x\in\mathbb{R}$ and $\mu,\lambda>0$:

$$\tilde{F}(x,\mu,\lambda)=\tilde{f}(\mathrm{clamp}(x,\mu+1+p(1+\mathrm{norm}_{\infty,1}(x,\mu,\lambda)),\lambda),\mu+1,\lambda)$$

Clearly $\tilde{F} \in \mathrm{GVAL}[\mathrm{poly}]$. Let $\mu, \lambda > 0$ and $x \in]a, b[$ such that $d(x, \Gamma \cup \{a, b\}) \geqslant \lambda^{-1}$. It follows from the results above that:

$$\begin{aligned}
\left|f(x) - \tilde{F}(x,\mu,\lambda)\right| &\leqslant \left|f(x) - \tilde{f}(x,\mu+1,\lambda)\right| + \left|\tilde{F}(x,\mu,\lambda) - \tilde{f}(x,\mu+1,\lambda)\right| \\
&\leqslant e^{-\mu-1} + \left|x - \mathrm{clamp}(x, \mu+1+p(1+\mathrm{norm}\infty,1(x,\mu,\lambda)),\lambda)\right| \\
&\quad \times p(\max(|x|, \\
&\left|\mathrm{clamp}(x,\mu+1+p(1+\mathrm{norm}\infty,1(x,\mu,\lambda)),\lambda)\right|, \mu+1,\lambda)) \\
&\leqslant e^{-\mu-1} + e^{-\mu-1-p(1+\mathrm{norm}\infty,1(x,\mu,\lambda))} \\
&\quad \times p\left(\max(|x|, |x| + e^{-\mu-1-p(1+\mathrm{norm}\infty,1(x,\mu,\lambda))}, \mu+1,\lambda)\right) \\
&\leqslant e^{-\mu-1} \\
&+ e^{-\mu-1-p(\max(1+|x|,\mu+1,\lambda))} p(\max(|x|,|x|+1,\mu+1,\lambda)) \\
&\leqslant 2e^{-\mu-1} \leqslant e^{-\mu}
\end{aligned}$$

□

Theorem 13.4 (Periodic approximability) *Let $f : \mathbb{R} \to \mathbb{R}$ be a τ-periodic function. Assume that there exists $a, b \in \mathbb{K}$ such that $b - a = \tau$ and f is GPAC-approximable over $]a, b[$ but Γ. Then f is GPAC-approximable over $\mathbb{R}$ but $(\Gamma \cup \{a, b\}) + \tau\mathbb{Z}$.*

Proof: First note that we can assume that $a + b = 0$: define $g(x) = f(x+\delta)$ where $\delta = \frac{a+b}{2}$, take a GPAC-approximation $\tilde{f}$ of f over $]a, b[$ but Γ. Observe that $\tilde{g}(x,\mu,\lambda) = \tilde{f}(x+\delta,\mu,\lambda)$ provides an approximation of g over $]a-\delta, b-\delta]$ but $\Gamma - \delta$. Then f is approximable over $\mathbb{R}$ but $(\Gamma \cup \{a, b\}) + \tau\mathbb{Z}$ if and only if g is approximable over $\mathbb{R}$ but $((\Gamma - \delta) \cup \{a - \delta, b - \delta\}) + \tau\mathbb{Z}$. Now observe that $(a - \delta) + (b - \delta) = a + b - 2\delta = 0$.

For a similar reason, we can assume that $\tau = 1$ by rescaling x. It follows that we can assume that $a = -1/2$ and $b = 1/2$. Let $\tilde{f}$ be a GPAC-approximation of f over $]\frac{-1}{2}, \frac{1}{2}[$ but Γ. We use the same trick as in Theorem 13.3 to ensure that $\tilde{f}$ is defined over $\mathbb{R}\times]0,+\infty[^2$. Let $\mathtt{sp}$ be a polynomial such that $\tilde{f} \in \mathrm{GVAL}[\mathtt{sp}]$. Apply Proposition 13.2 to $\tilde{f}$ to get a polynomial q. Recall that q acts as a modulus of continuity:

$$\left|\tilde{f}(x,\mu,\lambda) - f(y,\mu,\lambda)\right| \leqslant |x-y| q(\mathtt{sp}(\max(|x|,|y|,\mu,\lambda)))$$

for any $x, y \in]a, b[$ and $\mu, \lambda > 0$. Let $p \in \mathbb{K}[\mathbb{R}]$ be a nondecreasing polynomial such that $p(x) \geqslant q(\mathtt{sp}(x))$ for all $x \geqslant 0$. Define for any $x \in \mathbb{R}$ and $\mu, \lambda > 0$:

$$\tilde{F}(x,\mu,\lambda) = \tilde{f}(x - \mathrm{rnd}(x, \mu+1+p(1+\mathrm{norm}\infty,1(\mu,\lambda)),\lambda), \mu+1,\lambda)$$

Clearly $\tilde{F} \in \mathrm{GVAL}[\mathrm{poly}]$. Let $\mu, \lambda > 0$ and $x \in]a, b[$ such that $d(x, (\Gamma \cup \{a, b\}) + \tau\mathbb{Z}) \geqslant \lambda^{-1}$. It follows that there exists $n \in \mathbb{Z}$ such that $x = n + u$ where $u \in]\frac{-1}{2} + \lambda^{-1}, \frac{1}{2} - \lambda^{-1}[$ and $d(u, \Gamma) \geqslant \lambda^{-1}$. Apply Lemma 13.6 to get that $|\mathrm{rnd}(x, \mu+1+p(1+\mathrm{norm}\infty,1(\mu,\lambda)),\lambda) - n| \leqslant e^{-\mu-1-p(1+\mathrm{norm}\infty,1(\mu,\lambda))}$ so in particular $|x - \mathrm{rnd}(x, \mu+1+p(1+\mathrm{norm}\infty,1(\mu,\lambda)),\lambda) - u| \leqslant e^{-\mu-1-p(1+\mathrm{norm}\infty,1(\mu,\lambda))}$. In particular, $|x - \mathrm{rnd}(x, \mu+1+p(1+\mathrm{norm}\infty,1(\mu,\lambda)),\lambda)| \leqslant$

1. It follows that:

$$\begin{aligned}
\left|f(x)-\tilde{F}(x,\mu,\lambda)\right| &\leqslant \left|f(x)-\tilde{f}(u,\mu+1,\lambda)\right|+\left|\tilde{F}(x,\mu,\lambda)-\tilde{f}(u,\mu+1,\lambda)\right| \\
&\leqslant \left|f(x-n)-\tilde{f}(u,\mu+1,\lambda)\right| \\
&\quad+\left|x-\operatorname{rnd}(x,\mu+1+p(1+\operatorname{norm}\infty,1(\mu,\lambda)),\lambda)-u\right| \\
&\quad\times p(\max(|u|, \\
&\left|x-\operatorname{rnd}(x,\mu+1+p(1+\operatorname{norm}\infty,1(\mu,\lambda)),\lambda)\right|,\mu+1,\lambda)\big) \\
&\leqslant e^{-\mu-1}+e^{-\mu-1-p(1+\operatorname{norm}\infty,1(\mu,\lambda))}p\big(\max(1,1,\mu+1,\lambda)\big) \\
&\leqslant e^{-\mu-1}+e^{-\mu-1-p(\max(1,\mu+1,\lambda))}p(\max(1,\mu+1,\lambda)) \\
&\leqslant 2e^{-\mu-1}\leqslant e^{-\mu}
\end{aligned}$$

□

13.7 Generable fields

In Section 13.2, we introduced the notion of *generable field*, which are fields with an additional stability property. We used this notion to ensure that the class of functions we built is closed under composition. It is well-known that if we allow any choice of constants in our computation, we will gain extra computational power because of uncomputable real numbers. For this reason, it is wise to make sure that we can exhibit at least one generable field consisting of computable real numbers only, and possibly only polynomial time computable numbers in the sense of computable analysis [Brattka et al., 2008].

Intuitively, we are looking for a (the) smallest generable field, call it $\mathbb{R}_G$, in order to minimize the computation power of the real numbers it contains. The rest of this section is dedicated to the study of this field. We first recall Definition 13.2.

Definition 13.12 (Generable field) *A field* $\mathbb{K}$ *is* generable *if and only if* $\mathbb{Q}\subseteq\mathbb{K}$ *and for any* $\alpha\in\mathbb{K}$*, and* $(f:\mathbb{R}\to\mathbb{R})\in\mathrm{GVAL}_{\mathbb{K}}$*,* $f(\alpha)\in\mathbb{K}$*.*

13.7.1 Extended stability

By definition of a generable field, $\mathbb{K}$ is preserved by unidimensional generable functions. An interesting question is whether $\mathbb{K}$ is also preserved by multidimensional functions. This is not immediate because because of several key differences in the definition of multidimensional generable functions. We first recall a folklore topology lemma.

Lemma 13.12 (Offset of a compact set) *Let* $X\subseteq U\subseteq\mathbb{R}^n$ *where* U *is open and* X *is compact. Then there exists* $\varepsilon>0$ *such that* $X_\varepsilon\subseteq U$ *where the* ε*-offset of* X *is defined by* $X_\varepsilon=\bigcup_{x\in X}B_\varepsilon(x)$*.*

Proof:This is a very classical result: let $F = \mathbb{R}^n \setminus U$, then F is closed so the distance function[3] d_F to F is continuous. Since X is compact, $d_F(X)$ is a compact subset of $\mathbb{R}_+$, and $d_F(X)$ is nowhere 0 because $X \subseteq U \subseteq F$ where U is open. Consequently $d_F(X)$ admits a positive minimum ε. Let $x \in X_\varepsilon$, then $\exists y \in X$ such that $\left\|x-y\right\| < \varepsilon$, and by the triangle inequality, $\varepsilon \leqslant d_F(y) \leqslant \left\|x-y\right\| + d_F(x)$ so $d_F(x) > 0$ which means $x \notin F$, in other words $x \in U$. □

Lemma 13.13 (Polygonal path connectedness) *An open, connected subset U of $\mathbb{R}^n$ is always* polygonal-path-connected*: for any $a, b \in U$, there exists a polygonal path[a] from a to b in U. Furthermore, we can take all intermediate vertices in $\mathbb{Q}^n$.*

[a]A polygonal path is a connected sequence of line segments

Proof:This is a textbook property, e.g. Theorem 3-5 in [**?**]. □

Proposition 13.5 (Generable path connectedness) *An open, connected subset U of $\mathbb{R}^n$ is always* generable-path-connected*: for any $a, b \in U \cap \mathbb{K}^n$, there exists $(\phi : \mathbb{R} \to U) \in \mathrm{GVAL}_{\mathbb{K}}$ such that $\phi(0) = a$ and $\phi(1) = b$.*

Proof: Let $a, b \in U \cap \mathbb{K}^n$ and apply Lemma 13.13 to get a polygonal path $\gamma : [0,1] \to U$ from a to b. We are going to build a highly smoothed approximation of γ. This is usually done using bump functions but bump functions are not analytic, which complicates the matter. Furthermore, we need to build a path which domain of definition is $\mathbb{R}$, although this will be a minor annoyance only. We ignore the case where $a = b$ which is trivial and focus on the case where $a \neq b$.

Let $X = \gamma([0,1])$ which is a compact connected set. Apply Lemma 13.12 to get $\varepsilon > 0$ such that $X_\varepsilon \subseteq U$. Without loss of generality, we can assume that $\varepsilon \in \mathbb{Q}$ so that it is generable.

Assume for a moment that γ is trivial, that is γ is a line segment from a to b. Let $\alpha \in \mathbb{N} \subseteq \mathbb{K}$ such that $\frac{1}{\tanh(\alpha)} \leqslant 1 + \frac{2\varepsilon}{\|b-a\|}$. It exists because $\frac{1}{\tanh(x)} \xrightarrow[x\to\infty]{} 1$. Define $\phi(t) = a + \frac{1+\mu(t)}{2}(b-a)$ where $\mu(t) = \frac{\tanh((2t-1)\alpha)}{\tanh(\alpha)}$. One can check that μ is an increasing function and that $\mu(0) = -1$ and $\mu(1) = 1$. Furthermore, if $t > 1$, $|\mu(t) - 1| < \frac{2\varepsilon}{\|b-a\|}$, and conversely, if $t < 0$, $|\mu(t) + 1| < \frac{2\varepsilon}{\|b-a\|}$. Consequently, $\phi(0) = a$, $\phi(1) = b$ and $\phi([0,1])$ is the line segment between a and b, so $\phi([0,1]) \subseteq X$. Furthermore, if $t < 0$, $\left\|a - \phi(t)\right\| \leqslant \left|\frac{1+\mu(t)}{2}\right| \|b-a\| < \varepsilon$, and if $t > 1$, $\left\|b - \phi(t)\right\| \leqslant \left|\frac{1-\mu(t)}{2}\right| \|b-a\| < \varepsilon$. We conclude from this analysis that $\phi(\mathbb{R}) \subseteq X_\varepsilon \subseteq U$. It remains to show that $\phi \in \mathrm{GVAL}_{\mathbb{K}}$. Using Lemma 13.1, it suffices to show that $\tanh \in \mathrm{GVAL}_{\mathbb{K}}$ and $\frac{1}{\tanh(\alpha)} \in \mathbb{K}$. Since $\mathbb{K}$ is a field, we need to show that $\tanh(\alpha) \in \mathbb{K}$ which is a consequence of $\mathbb{K}$ being a generable field and tanh being a generable function. We already saw in Example 13.2 that $\tanh \in \mathrm{GVAL}_{\mathbb{Q}} \subseteq \mathrm{GVAL}_{\mathbb{K}}$.

In the general case where γ is a polygonal path, there are $0 = t_1 < t_2 < \ldots < t_k = 1$ such that $\gamma\restriction_{[t_i, t_{i+1}]}$ is the line segment between $x_i = \gamma(t_i)$ and $x_{i+1} = \gamma(t_{i+1})$, furthermore we can always take $x_i \in \mathbb{Q}^n$. Note that we can choose any parametrization for

[3]We always use the infinite norm $\|\cdot\|$ in this paper but it works for any distance

the path so in particular we can take $t_i = \frac{i}{k}$ and ensure that $t_i \in \mathbb{Q}$ for $i \in [\![0,k]\!]$. Since by hypothesis $x_0, x_n \in \mathbb{K}^n$, we get that $x_i \in \mathbb{K}^n$ and $t_i \in \mathbb{K}$ for all $i \in [\![0,k]\!]$.

Let us denote by $\phi_\varepsilon^{a,b}$ the path built in the previous case. We are simply going to add several instances of this path, with the necessary shifting and scaling. Since the errors will sum up, we will increase the approximation precision of each segment. Define $\phi(t) = a + \sum_{i=1}^{k-1}\left(\phi_{\varepsilon/k}^{x_i,x_{i+1}}\left(\frac{t-t_i}{t_{i+1}-t_i}\right) - x_i\right)$ and consider the following cases:

- if $t < 0$, then $\left\|\phi_{\varepsilon/k}^{x_i,x_{i+1}}\left(\frac{t-t_i}{t_{i+1}-t_i}\right) - x_i\right\| < \frac{\varepsilon}{k}$ for all $i \in [\![1,k-1]\!]$, so $\left\|a - \phi(t)\right\| < \frac{k-1}{k}\varepsilon$ and $\phi(t) \in X_\varepsilon$
- if $t \in [t_j, t_j+1]$ for some j, then $\left\|\phi_{\varepsilon/k}^{x_i,x_{i+1}}\left(\frac{t-t_i}{t_{i+1}-t_i}\right) - x_i\right\| < \frac{\varepsilon}{k}$ for all $i > j$, and conversely $\left\|\phi_{\varepsilon/k}^{x_i,x_{i+1}}\left(\frac{t-t_i}{t_{i+1}-t_i}\right) - x_{i+1}\right\| < \frac{\varepsilon}{k}$ for all $i < j$. Finally $u = \phi_{\varepsilon/k}^{x_j,x_{j+1}}\left(\frac{t-t_j}{t_{j+1}-t_j}\right)$ belongs to the line segment from x_j to x_j+1. Since $a = x_1$, we get that $\left\|u - \phi(t)\right\| \leqslant \frac{k-1}{k}\varepsilon$ and thus $\phi(t) \in X_\varepsilon$.
- if $t > 1$ then $\left\|b - \phi(t)\right\| < \varepsilon$ for the same reason as $t < 0$, and thus $\phi(t) \in X_\varepsilon$.

We conclude that $\phi(\mathbb{R}) \subseteq X_\varepsilon \subseteq U$ and one easily checks that $\phi(0) = a$ and $\phi(1) = b$. Furthermore $\phi \in \mathrm{GVAL}_\mathbb{K}$ by Lemma 13.1 and because the x_i and t_i belong to $\mathbb{K}$ (see the details in the case of the trivial path). □

The immediate corollary of this result is that $\mathbb{K}$ is also preserved by multidimensional generable functions. Indeed, by composing a multidimensional function with a unidimensional one, we get back to the unidimensional case and conclude that any generable point in the input domain must have a generable image.

Corollary 13.2 (Generable field stability) *Let* $(f :\subseteq \mathbb{R}^d \to \mathbb{R}^\ell) \in \mathrm{GVAL}_\mathbb{K}$, *then* $f(\mathbb{K}^d \cap \operatorname{dom} f) \subseteq \mathbb{K}^\ell$.

Proof: Apply Definition 13.3 to get $n \in \mathbb{N}$, $p \in M_{n,d}(\mathbb{K})[\mathbb{R}^n]$, $x_0 \in \operatorname{dom} f \cap \mathbb{K}^d$, $y_0 \in \mathbb{K}^n$ and $y : \operatorname{dom} f \to \mathbb{R}^n$. Let $u \in \operatorname{dom} f \cap \mathbb{K}^d$. Since $\operatorname{dom} f$ is open and connected, by Proposition 13.5, there exists $(\gamma : \mathbb{R} \to \operatorname{dom} f) \in \mathrm{GVAL}$ such that $\gamma(0) = x_0$ and $\gamma(1) = u$. Apply Definition 13.3 to γ to get $\bar{n} \in \mathbb{N}$, $\bar{p} \in M_{\bar{n},1}(\mathbb{K})[\mathbb{R}^{\bar{n}}]$, $\bar{x}_0 \in \mathbb{K}$, $\bar{y}_0 \in \mathbb{K}^{\bar{n}}$ and $\bar{y} : \mathbb{R} \to \mathbb{R}^{\bar{n}}$. Define $z(t) = y(\gamma(t)) = y(\bar{y}_{1..d}(t))$, then $z'(t) = J_y(\gamma(t))\gamma'(t) = p(y(\gamma(t)))\gamma'(t) = p(z(t))\bar{p}_{1..d}(\bar{y}(t))$ and $z(0) = y(\gamma(0)) = y(x_0) = y_0$. In other words $(\bar{y}, z)$ satisfy:

$$\begin{cases}\bar{y}(0) = x_0 \in \mathbb{K}^d \\ z(0) = y_0 \in \mathbb{K}^n\end{cases} \qquad \begin{cases}\bar{y}' = \bar{p}(\bar{y}) \\ z' = p(z)\bar{p}_{1..\ell}(\bar{y})\end{cases}$$

Consequently $(z : \mathbb{R} \to \mathbb{R}^\ell) \in \mathrm{GVAL}$ so, by definition of a generable field, $z(\mathbb{K}) \subseteq \mathbb{K}^z ell$. Conclude by noticing that $z(1) = y(\gamma(1)) = y(u)$. □

13.7.2 Generable real numbers

In this section, we formalize the notion of generable field with an operator and study its properties. Recall that the smallest field we are looking for is a subset of $\mathbb{R}$ but it must also contains $\mathbb{Q}$. We consider the following operator G on subset of real numbers.

$$G:\left\{\begin{array}{ccc}\mathscr{P}(\mathbb{R}) & \to & \mathscr{P}(\mathbb{R})\\ X & \mapsto & \bigcup_{f\in\text{GVAL}_X} f(X)\end{array}\right.$$

Remark 13.10 (G monotone and non-decreasing) *One can check that G is monotone ($X\subseteq G(X)$ for any $X\subseteq\mathbb{R}$). Indeed for any $x\in X$, the constant function $u\mapsto x$ belongs to* GVAL_X. *Moreover, it is non-decreasing because* $\text{GVAL}_X\subseteq\text{GVAL}_Y$ *if* $X\subseteq Y$.

It is clear that by definition, a field is generable if and only if it is G-stable. An interesting property of G is that its definition can be simplified. More precisely, by rescaling the functions, we can always assume that the image of G is produced by the evaluation of generable functions at a particular point, say 1, instead of the entire field.

Lemma 13.14 (Alternative definition of G) *If X is a field then,*

$$G(X)=\left\{f(1):f\in\text{GVAL}_X\right\}$$

Proof: Let $x\in G(X)$, then there exists $f\in\text{GVAL}_X$ and $t\in X$ such that $x=f(t)$. Consequently there exists $d\in\mathbb{N}$, $y_0\in X^d$, $p\in X^d[\mathbb{R}^d]$ and $y:\mathbb{R}\to\mathbb{R}^d$ satisfying Definition 13.1:

- $y'=p(y)$ and $y(0)=y_0$
- $y_1=f$

Consider $g(u)=f(ut)$ and note that $g(1)=f(t)=x$. We will see that $g\in\text{GVAL}_X$. Indeed, consider $z(u)=y(tu)$ then for all $u\in\mathbb{R}$:

- $z(0)=y(0)=y_0\in X^d$;
- $z'(u)=ty'(tu)=tp(z(u))=q(z(u))$ where $q=tp$ is a polynomial with coefficients in X since $t\in X$ and X is a field
- $z_1(u)=y_1(tu)=g(u)$

□

A consequence of this alternative definition is a simple proof that G preserves the property of being a field. This will turn out to be crucial fact later on.

Lemma 13.15 (G maps fields to fields) *If X is a field, then $G(X)$ is a field.*

Proof: Let $x,y\in G(X)$, by Lemma 13.14 there exists $f,g\in\text{GVAL}_X$ such that $x=f(1)$ and $y=g(1)$. Apply Lemma 13.1 to get that $f\pm g$ and fg belong to GVAL_X And thus $x\pm y$ and xy belong to $G(X)$.

Finally the case of $\frac{1}{x}$ (when $x \neq 0$) is slightly more subtle: we cannot simply compute $\frac{1}{f}$ because f may cancel. Instead we are going to compute $\frac{1}{g}$ where $g(1) = f(1)$ but g nevers cancels.

First, note that we can always assume that $x > 0$ because $G(X)$ is closed under the negation, and $-\frac{1}{x} = \frac{1}{-x}$. Since $f(1) = x > 0$ and f is continuous, it means there exists $\varepsilon > 0$ such that $f(t) > 0$ for all $t \in [1-\varepsilon, 1+\varepsilon]$ and we can take $\varepsilon \in \mathbb{Q}$. Define $g(t) = f(t) + \left(1 + f(t)^2\right)\left(\frac{t-1}{\varepsilon}\right)^2$. It is not hard to see that $g(1) = f(1)$ and that $g(t) > 0$ for all $t \in \mathbb{R}$. Furthermore, $g \in \mathrm{GVAL}_X$ because of Lemma 13.1. Note that we use the part of the lemma which does not assume that X is a generable field!

Using Lemma 13.1, we conclude that $\frac{1}{g} \in \mathrm{GVAL}_X$ and thus $\frac{1}{x} \in G(X)$. □

Not only G maps fields to fields, but it also preserves polynomial-time computability. This is of major interest to us to show that there exists a generable field with low complexity numbers. Here $\mathbb{R}_P$ denotes the set of polynomial time computable real numbers [Ko, 1991].

Lemma 13.16 (G preserves polytime computability) *G maps subsets of polynomial time computable real numbers into themselves,* i.e. *for any $X \subseteq \mathbb{R}_P$, $G(X) \subseteq \mathbb{R}_P$.*

Proof: Let $X \subseteq \mathbb{R}_P$ and $x \in G(X)$, $f \in \mathrm{GVAL}_X$ and $t \in X$ such that $x = f(t)$. We can use [**?**] to conclude that x is polynomial time computable, thus $x \in \mathbb{R}_P$. □

Finally, the core of what makes G very special is its finiteness property. Essentially, it means that if $x \in G(X)$ then x really only requires a finite number of elements in X to be computed. In the framework of order and lattice theory, this shows that G is a Scott-continuous function between the complete partial order (CPO) $(\mathscr{L}, \subseteq)$ and itself.

Lemma 13.17 (Finiteness of G) *For any $X \subseteq \mathbb{R}$ and $x \in G(X)$, there exists a finite $Y \subseteq X$ such that $x \in G(Y)$.*

Proof: Let $x \in G(X)$, then there exists $f \in \mathrm{GVAL}_X$ and $t \in X$ such that $x = f(t)$. Then there exists $y_0 \in X^d$ and a polynomial p with coefficients in X such that f satisfies Definition 13.1. Define Y as the subset of X containing t, the components of y_0 and all the coefficients of p. Then Y is finite and $f \in \mathrm{GVAL}_Y$. Furthermore $t \in Y$ so $x \in G(Y)$. □

We can now define the set of "generable real numbers", call it $\mathbb{R}_G$. The main result of this section is that $\mathbb{R}_G$ is the smallest generable field. But more surprisingly, we show that all the elements of $\mathbb{R}_G$ are polynomial time computable (in the sense of Computable Analysis).

Definition 13.13 (Generable real numbers)

$$\mathbb{R}_G = \bigcup_{n \geqslant 0} G^{[n]}(\mathbb{Q}).$$

Theorem 13.5 ($\mathbb{R}_G$ is generable subfield of $\mathbb{R}_P$) *$\mathbb{R}_G$ is the smallest generable field for inclusion. Furthermore, it form a generable subfield of polynomial time computable real numbers in the sense of Computable Analysis, i.e. $\mathbb{R}_G \subseteq \mathbb{R}_P$.*

Proof: First observe that any generable field must contain $\mathbb{R}_G$. Indeed, let $\mathbb{K}$ be a generable field: then $G(\mathbb{K}) \subseteq \mathbb{K}$ by definition. But G is non-decreasing thus $G(\mathbb{Q}) \subseteq G(\mathbb{K}) \subseteq \mathbb{K}$. By applying G repeatedly, we get that $G^{[n]}(\mathbb{Q}) \subseteq \mathbb{K}$ for all n. Thus $\mathbb{R}_G \subseteq \mathbb{K}$.

Conversely, we need to show that $\mathbb{R}_G$ is a field. Observe that since G is monotone, $G^{[n]}(\mathbb{Q})$ is an increasing sequence (for inclusion). Let $x, y \in \mathbb{R}_G$, then there exists $n \in \mathbb{N}$ such that $x, y \in G^{[n]}(\mathbb{Q})$. Apply Lemma 13.15 to get that $G^{[n]}(\mathbb{Q})$ is a field. It follows that $x+y, x-y, xy$ and $\frac{x}{y}$ (if $y \neq 0$) belong to $G^{[n]}(\mathbb{Q}) \subseteq \mathbb{R}_G$. Thus $\mathbb{R}_G$ is a field.

It remains to show that $\mathbb{R}_G$ is a *generable* field. This follows from Lemma 13.17: let $x \in G(\mathbb{R}_G)$, then there exists a **finite** $Y \subseteq \mathbb{R}_G$ such that $x \in G(Y)$. Using the same reasoning as above, there exists $n \in \mathbb{N}$ such that $Y \subseteq G^{[n]}(\mathbb{Q})$. Thus $x \in G(Y) \subseteq G(G^{[n]}(\mathbb{Q})) = G^{[n+1]}(\mathbb{Q}) \subseteq \mathbb{R}_G$. It follows that $G(\mathbb{R}_G) \subseteq \mathbb{R}_G$, *i.e.* it is generable.

Finally, since $\mathbb{Q} \subseteq \mathbb{R}_P$, iterating Lemma 13.16 yields that $G^{[n]}(\mathbb{Q}) \subseteq \mathbb{R}_P$ for all $n \in \mathbb{N}$ and thus $\mathbb{R}_G \subseteq \mathbb{R}_P$. □

Chapter 14

Simulating Turing Machines by Analytic Functions

14.1 Coding of configurations of Turing machines

14.1.1 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

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

(1) represent the tape contents of the Turing machine M. We suppose the head to be reading symbol a_0 and $a_i \in \{0, 1, \dots, 9\}$ 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.

14.1.2 Coding a configuration: using integers.

Take

$$\begin{aligned} y_1 &= a_0 + a_1 10 + \dots + a_n 10^n, \\ y_2 &= a_{-1} + a_2 10 + \dots + a_{-k} 10^{k-1}, \end{aligned}$$

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

Then the triple
$$\gamma^{TM}_{\mathbb{N}^3}(C) = (y_1, y_2, q) \in \mathbb{N}^3$$
encodes the current configuration C of M by an element of $\mathbb{N}^3$.

14.1.3 Coding a configuration: using integers (variant).

Then the triple
$$\gamma^{TM}_{\mathbb{N}^2}(C) = (q + (m+1)y_1, y_2, q) \in \mathbb{N}^3$$
encodes the current configuration C of M by an element of $\mathbb{N}^2$.

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

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

Then the triple

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

14.1.5 Coding a configuration: using $[0,1]$

Take

$$\begin{aligned} y_1 &= a_0 10^{-1} + a_1 10^{-2} + \cdots + a_n 10^{-n-1}, \\ y_2 &= a_{-1} 10^{-1} + a_2 10^{-2} + \cdots + a_{-k} 10^{-k}, \end{aligned}$$

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

Then the triple

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

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

14.2 Discrete time simulation: $\gamma^{TM}_{\mathbb{N}^2}$

14.2.1 Koiran-Moore 99's theorem

Definition 14.1 *Let U_n be the smallest class of functions $f : \mathbb{R}^n \to \mathbb{R}$ containing rational constants, π, 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.

Theorem 14.1 (Koiran-Moore's Theorem [Koiran and Moore, 1999]) *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 $f^{[t]}(a + bw, 0) = (0, 0)$.*

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)

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

$$h_{10}(y_2 - a) = \begin{cases} 1 & \text{if } a_{-1} = a \\ 0 & \text{otherwise} \end{cases}$$

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 $(0, 0)$.

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

14.3.1 Graça-Campagnolo-Buescu's Theorem 1

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

Theorem 14.2 (Graça-Campagnolo-Buescu's [Graça et al., 2005a, Theorem 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 θ 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 $\left\| f - f_M \right\| \leq \delta$, and for all $\overline{x_0} \in \mathbb{R}^3$ satisfying $\left\| \overline{x_0} - x_0 \right\| \leq \epsilon$, 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 ϵ results from the chosen encodingn, which is over the integers. In fact, the bound is maximal with respect to that encoding.

14.3.2 Some basic functions

Mod function ω

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 $f(n) = n \bmod 10$. 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

$$\omega(x) = a_0 + a_5 \cos(\pi x) + \left(\sum_{j=1}^{4} a_j \cos\left(\frac{j\pi x}{5} + b_j \sin(\frac{j\pi x}{5} \right) \right), \tag{14.1}$$

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

$$\forall n,\ x \in [n - \eta_\epsilon, n + \eta_\epsilon] \Rightarrow |\omega(x) - n \bmod 10| \leq \epsilon. \tag{14.2}$$

Error correcting function σ

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 σ is a contraction on the vicinity of integers:

Lemma 14.1 *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 $\forall \delta \in [-\epsilon,\epsilon]$, $|\sigma(n+\delta) - n| < \lambda_\epsilon \delta$.*

Proof: It is sufficient to consider the case where $n = 0$. Because σ is odd, we only study σ in the interval $[0,\varepsilon]$. Let $g(x) = \sigma(x)/x$. This function is strictly increasing in $(0,1/2]$. Then, noting that $g(1/2) = 1$ and $\lim_{x\to 0} g(x) = 1 - 0.4\pi \approx -0.256637$, we conclude that there exists some $\lambda_\varepsilon \in (0,1)$ such that $|\sigma(x)| < \lambda_\varepsilon |x|$ for all $x \in [-\varepsilon,\varepsilon]$ □

For the rest of this document, we suppose that $\epsilon \in [0,1/2)$ is fixed and that λ_ϵ is the respective contracting factor given by Lemma 14.1.

The function σ 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.).

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 $l_3 : \mathbb{R}^2 \to \mathbb{R}$, that has the property that whenever $\overline{a}$ is an approximation of $a \in \{0,1,2\}$, then $|a - l_3(\overline{a}, l)| < 1/y$, 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.

Lemma 14.2 ([Graça, 2007, Lemma 4.2.3]) $\left|\frac{\pi}{2} - \arctan x\right| < \frac{1}{x}$ *for* $x \in (0,\infty)$

Proof: Let $f(x) = \frac{1}{x} + \arctan x - \frac{\pi}{2}$. It is easy to see that f is decreasing in $(0,\infty)$ and that $\lim_{x\to\infty} f(x) = 0$. Therefore $f(x) > 0$ for $x \in (0,\infty)$ and the result holds. □

Lemma 14.3 ([Graça, 2007, Lemma 4.2.4]) $\left|\frac{\pi}{2} + \arctan x\right| < \frac{1}{|x|}$ *for* $x \in (-\infty,0)$.

Proof: Take $f(x) = \frac{1}{x} + \arctan x + \frac{\pi}{2}$ and proceed as in Lemma 14.2. □

In order to define function l_3 we first define a preliminary function l_2 satisfying similar conditions, but only when $a \in \{0,1\}$.

Lemma 14.4 ([Graça, 2007, Lemma 4.2.5]) *Let $l_2 : \mathbb{R}^2 \to \mathbb{R}$ be given by*

$$l_2(x,y) = \frac{1}{\pi}\arctan(4y(x-1/2)) + \frac{1}{2}.$$

Suppose that $a \in \{0,1\}$. Then, for any $\overline{a}, y \in \mathbb{R}$ satisfying $|a - \overline{a}| \le 1/4$ and $y > 0$, we get $|a - l_2(\overline{a}, y)| < 1/y$.

Proof:

- Consider $a = 0$. Then $\bar{a} - 1/2 \leq -1/4$ implies $|4y(\bar{a} - 1/2)| \geq y$. Therefore, by Lemma 14.3

  $$\left|\frac{\pi}{2} + \arctan(4y(\bar{a} - 1/2))\right| < \frac{1}{|4y(\bar{a} - 1/2)|} \leq \frac{1}{y}.$$

  Moreover, multiplying the last inequality by $1/\pi$ and noting that $\frac{1}{\pi y} < \frac{1}{y}$, it follows that

  $$\left|a - l_2(\bar{a}, y)\right| < 1/y.$$

- Consider $a = 1$. Remark that $\bar{a} - 1/2 \geq 1/4$ and proceed as above, using Lemma 14.2 instead of Lemma 14.3.

□

Lemma 14.5 ([Graça, 2007, Lemma 4.2.7]) *Let $a \in \{0,1,2\}$ and let $l_3 : \mathbb{R}^2 \to \mathbb{R}$ be given by*

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

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 $|a - l_3(\overline{a}, y)| < 1/y$.

Proof: Let us start by noticing that for all $x, y \in \mathbb{R}$ for which $l_2(x, y)$ is defined, we have that $0 < l_2(x, y) < 1$. Consider the case where $a = 0$ and $\bar{a} \in [-1/4; 1/4]$. By other words, take $\varepsilon \leq 1/4$. Then $\left|(\sigma(\bar{a}) - 1)^2 - 1\right| < 1/4$, and by the previous lemma,

$$1 - 1/y < l_2\left((\sigma(\bar{a}) - 1)^2, y\right) < 1$$

Similarly, we conclude

$$-1 < 2l_2(\bar{a}/2, y) - 1 < -1 + 2/y$$

Since $y \geq 2$, this implies

$$-1 < l_2\left((\sigma(\bar{a}) - 1)^2, y\right)\left(2l_2(\bar{a}/2, y) - 1\right) < (1 - 1/y)(-1 + 2/y)$$

Or

$$0 < l_2\left((\sigma(\bar{a}) - 1)^2, y\right)\left(2l_2(\bar{a}/2, y) - 1\right) + 1 < 3/y$$

Hence, for $a = 0, \left|a - l_3(\bar{a}, y)\right| < 1/y$. Proceeding similarly for $a = 1,2$ and $\varepsilon \leq 1/4$, the same result follows.

It remains to consider the more general case $|a - \bar{a}| \leq \varepsilon$. Taking $d = \left\lceil -\log(4\varepsilon)/\log\lambda_\varepsilon \right\rceil$ and applying d times the function σ to $\bar{a}$, it follows that $\left|a - \sigma^{[d]}(\bar{a})\right| \leq 1/4$ and we fall back in the previous case (use $\sigma^{[d]}(\bar{a})$ instead of $\bar{a}$) □

An observation

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

Lemma 14.6 *If* $|\alpha_i|, |\overline{\alpha}_i| \leq K$ *for* $i = 1, \ldots, n$ *then*

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

14.3.3 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 14.3 *Let* $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ *be the transition function of some Turing machine. Then, given some* $0 \leq \epsilon < 1/2$, θ *admits an analytic extension* $h_M : \mathbb{R}^3 \to \mathbb{R}^3$ *with the property that*

$$\left\|(y_1, y_2, q) - (\overline{y}_1, \overline{y}_2, \overline{q})\right\| \leq \epsilon \Rightarrow \left\|\theta(y_1, y_2, q) - h_M(\overline{y}_1, \overline{y}_2, \overline{q})\right\| \leq \epsilon \qquad (14.3)$$

where $(y_1, y_2, q) \in \mathbb{N}^3$ *encodes some configuration of* M.

14.3.4 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 ω is given by (14.1).

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

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

   $$|a_0 - \omega \circ \sigma^{[l]}(\overline{y}_1)| \leq \epsilon, \quad \text{with } l = \left\lceil \left| \frac{\log(\chi_\epsilon/\epsilon)}{\log \lambda_\epsilon} \right| \right\rceil, \qquad (14.4)$$

   where χ_ϵ is given (14.2). 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 ϵ.

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

$$q_{next} = \sum_{i=0}^{9}\sum_{j=1}^{m}\left(\prod_{r=0,r\neq i}^{9}\frac{(\sigma^{[n]}(\overline{y})-r)}{(i-r)}\right)\left(\prod_{s=1,s\neq j}^{m}\frac{(\sigma^{[n]}(\overline{q})-s)}{(j-s)}\right)q_{i,j}, \tag{14.5}$$

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

$$9|y-\sigma^{[n]}(\overline{y})| + (m-1)|q-\sigma^{[n]}(\overline{q})| \leq \frac{\epsilon}{10m^2K^{m+7}}. \tag{14.6}$$

Thus from (14.5), (14.6) and Lemma 14.6, w concluse 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 ϵ , 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 , y_1^{next}, can next be approximated by

$$\overline{y}_1^{next} = \overline{P}_1\frac{1}{2}(1-H)(2-H) + \overline{P}_2H(2-H) + \overline{P}_3(-\frac{1}{2})H(1-H), \tag{14.7}$$

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 (14.4). Notice that when exact values are used, $\overline{y}_1^{next} = y_1^{next}$. 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 **??** and the definition of y_1, one can see that it is suffices to take

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

Using the same argument for $\overline{P}_2$ and $\overline{P}_3$ we conclude that $|\overline{y}_1^{next} - y_1^{next}| < \epsilon$.
Similarly, and for the left side of the tape, we can define $\overline{y}_2^{next}$ such that $|\overline{y}_2^{next} - y_2^{next}| < \epsilon$.
Finally, $h_M : \mathbb{R}^3 \to \mathbb{R}^3$ is defined by $h_M(\overline{y}_1, \overline{y}_2, \bar{q}) = (\overline{y}_1^{next}, \overline{y}_2^{next}, \overline{q}_{next})$.

□

14.3.5 Proof of Graça-Campagnolo-Buescu's Theorem 1

Let $0 \le \delta < \epsilon$. Then, using Theorem 14.3, one can find a map h_M such that (14.3) holds. Let $i \in \mathbb{N}$ satisfy $\sigma^{[i]}(\epsilon) \le \epsilon - \delta$. Define a map $f_M = \sigma^{[i]} \circ h_M$. Then, if $x_0 \in \mathbb{N}^3$ is an initial configuration,

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

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

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

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

14.4 Discrete time simulation: $\gamma^{TM}_{(0,1)^2}$

Now we use the previous construction to simulate a Turing machine on a compact set $X = (-1,1)^3$. This is a trick similar to the one used in [Bournez et al., 2013], based on a change of variable.

If the iterations of $f_M(x, y, z)$ simulate M over $\mathbb{N}^3$, then

$$\tilde{f_M}(\tilde{x}, \tilde{y}, \tilde{z}) = \psi(f_M(\psi^{-1}(\tilde{x}, \tilde{y}, \tilde{z}))$$

does the same over $(0,1)$ considering

$$\psi(x, y, z) = (\tilde{x}, \tilde{y}, \tilde{z}) = (\frac{2}{\pi}\arctan x, \frac{2}{\pi}\arctan y, \frac{2}{\pi}\arctan z)$$

And thus,

$$\begin{aligned} x &= \tan(\tilde{x}\frac{\pi}{2}) \\ y &= \tan(\tilde{y}\frac{\pi}{2}) \\ z &= \tan(\tilde{z}\frac{\pi}{2}) \end{aligned}$$

We get:

Theorem 14.4 *Let $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of a Turing machine M, under the encoding $\gamma^{TM}_{(0,1)^2}$ described above. Then θ admits an analytic extension $f_M : (0,1)^3 \to (0,1)^3$, robust to perturbations in the following sense: there exists $0 < \epsilon$ such that for all $\overline{x_0} \in \mathbb{R}^3$ satisfying $\left\|\overline{x_0} - x_0\right\| \le \epsilon$, where $x_0 \in (0,1)^3$ represents an initial configuration,*

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

Basically, the simulation of M can be carried out if the states are not perturbed more than

$$\tilde{\varepsilon} = \arctan(m + \varepsilon) - \arctan(m) \tag{14.8}$$

14.5 Continuous time simulation: $\gamma^{TM}_{\mathbb{N}^3}$

14.5.1 Statement

Theorem 14.5 *$\theta : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of a Turing machine M, under the encoding $\gamma^{TM}_{\mathbb{N}^3}$ described above and let $0 < \epsilon < 1/4$. Then there is an analytic function $g_M : \mathbb{R}^6 \to \mathbb{R}^6$ such that the ODE $z' = g_M(z,t)$ robustly simulates M in the following sense: there is some $0 < \eta < 1/2$ such that for all g satisfying $\left\|g - g_M\right\| < 1/2$, and for all $\overline{x_0} \in \mathbb{R}^3$ satisfying $\left\|\overline{x_0} - x_0\right\| \le \epsilon$ the solution of*

$$z' = g(z,t), \quad z(0) = (\overline{x_0}, \overline{x_0})$$

has the following property: for all $j \in \mathbb{N}$, and for all $t \in [j, j + 1/2]$,

$$\left\|z_2(t) - \theta^{[j]}(x_0)\right\| \le \eta.$$

14.5.2 Iteration maps with ODEs

In this section we show how to iterate a map from integers to integers with smooth ODEs. By a smooth ODE we understand an ODE

$$y' = f(t,y) \tag{14.9}$$

where f is of class C^k, for $1 \le k \le \infty$ (but not necessarily analytic). Basically, we will describe the construction presented by Branicky in [Branicky, 1995], but following the approach of Campagnolo, Costa, and Moore [**?**, **?**, **?**]. Then using the map f_M given by Theorem **??**, we will be able to simulate TMs with smooth ODEs. This result will be extended in the next section to the case of polynomial ODEs robust to perturbations.

Suppose that $f : \mathbb{Z}^k \to \mathbb{Z}^k$ is a map. For simplicity, let us assume $k = 1$. For better readability, we also break down the procedure in three subtasks.

Construction 1

Consider a point $b \in \mathbb{R}$ (the target), some $\gamma > 0$ (the targeting error), and time instants t_0 (departure time) and t_1 (arrival time), with $t_1 > t_0$. Then obtain an IVP (the targeting equation) defined with an ODE (14.9) where $f : \mathbb{R}^2 \to \mathbb{R}$, such that the solution y satisfies

$$\left|y(t_1) - b\right| < \gamma \tag{14.10}$$

independently of the initial condition $y(t_0) \in \mathbb{R}$

Let $\phi : \mathbb{R} \to \mathbb{R}_0^+$ be some function satisfying $\int_{t_0}^{t_1} \phi(t)dt > 0$ and consider the following ODE

$$y' = c(b-y)^3\phi(t) \tag{14.11}$$

where $c > 0$. There are two cases to consider: (i) $y(t_0) = b$, (ii) $y(t_0) \neq b$. In the first case, the solution is given by $y(t) = b$ for all $t \in \mathbb{R}$ and (14.13) is trivially satisfied. For the second case, note that (14.11) is a separable equation, which gives

$$\frac{1}{\left(b - y(t_1)\right)^2} - \frac{1}{\left(b - y(t_0)\right)^2} = 2c\int_{t_0}^{t_1} \phi(t)dt \Rightarrow$$
$$\frac{1}{2c\int_{t_0}^{t_1}\phi(t)dt} > \left(b - y(t_1)\right)^2$$

Hence, (14.13) is satisfied if c satisfies $\gamma^2 \geq \left(2c\int_{t_0}^{t_1}\phi(t)dt\right)^{-1}$ i.e. if

$$c \geq \frac{1}{2\gamma^2\int_{t_0}^{t_1}\phi(t)dt} \tag{14.12}$$

14.5.3 Construction 2

Obtain an IVP defined with an ODE (14.9) where $f : \mathbb{R}^2 \to \mathbb{R}$, such that the solution r satisfies

$$r(x) = j \text{ whenever } x \in [j - 1/4, j + 1/4] \text{ for all } j \in \mathbb{Z} \tag{14.13}$$

We want a function $r : \mathbb{R} \to \mathbb{R}$ satisfying this condition for the following reason. Suppose that on Construction 1, $\gamma < 1/4$ and that $b \in \mathbb{N}$. Then $r\left(y(t_1)\right) = b$, i.e. r corrects the error present in $y(t_1)$ when approaching an integer value b. This will be useful later in this document.

First let $\theta_j : \mathbb{R} \to \mathbb{R}$, $j \in \mathbb{N} - \{0, 1\}$, be the function defined by

$$\theta_j(x) = 0 \text{ if } x \leq 0, \quad \theta_j(x) = x^j \text{ if } x > 0$$

For $j = \infty$ define

$$\theta_j(x) = 0 \text{ if } x \leq 0, \quad \theta_j(x) = e^{-\frac{1}{x}} \text{ if } x > 0 \tag{14.14}$$

These functions can be seen [Campagnolo et al., 2000b] as a C^{j-1} version of Heaviside's step function $\theta(x)$, where $\theta(x) = 1$ for $x \geq 0$ and $\theta(x) = 0$ for $x < 0$

With the help of θ_j, we define a "step function" $s : \mathbb{R} \to \mathbb{R}$, that matches the identity function on the integers, as follows:

$$\begin{cases} s'(x) = \lambda_j \theta_j(-\sin 2\pi x) \\ s(0) = 0 \end{cases}$$

where

$$\lambda_j = \int_{1/2}^{1} \theta_j(-\sin 2\pi x)\,dx > 0$$

For $x \in [0, 1/2]$, $s(x) = 0$ since $\sin 2\pi x \geq 0$. On $(1/2, 1)$, s strictly increases and satisfies $s(1) = 1$.

Using the same argument for $x \in [j, j+1]$, for all integer j, we conclude that $s(x) = j$ whenever $x \in [j, j+1/2]$. Then defining $r : \mathbb{N} \to \mathbb{N}$ by $r(x) = s(x + 1/4)$, it is easy to see that r satisfies the conditions set for Construction 2.

One should remark that, for each $j \in \mathbb{N} \cup \{\infty\} - \{0, 1\}$, we get a different function r, but they all have the same fundamental property (14.14). So, we choose to omit the reference to index j when defining r (this does not represent any problem in later results)

14.5.4 Construction 3

Iterate the map $f : \mathbb{Z} \to \mathbb{Z}$ with a smooth ODE (14.9).

Let $\tilde{f} : \mathbb{R} \to \mathbb{R}$ be an arbitrary smooth extension to the reals of f, and consider the IVP defined with the smooth ODE

$$\begin{cases} z_1' = c_{j,1} \left(\tilde{f}\,(r\,(z_2)) - z_1\right)^3 \theta_j(\sin 2\pi t) \\ z_2' = c_{j,2}\,(r\,(z_1) - z_2)^3 \theta_j(-\sin 2\pi t) \end{cases} \tag{14.15}$$

and the initial condition

$$\begin{cases} z_1(0) = x_0 \\ z_2(0) = x_0 \end{cases}$$

where $x_0 \in \mathbb{N}$. We shall use the previous two constructions to iterate f. First, we use Construction 1 with parameters satisfying: $\gamma \leq 1/4, t_0 = 0, t_1 = 1/2, \phi = \phi_1$ where $\phi_1(t) = \theta_j(\sin 2\pi t)$ and $c = c_{j,1}$ given by (14.12) With these parameters, let us look to (**??**). We have that for $t \in [0, 1/2], z_2'(t) = 0$. Therefore the first equation of (14.15) becomes

$$z_1' = c\,(b - z_1)^3\,\phi(t)$$

where $b = f\,(x_0)$. Thus one has $\left|z_1(1/2) - f\,(x_0)\right| < \gamma \leq 1/4$. Now, for $t \in [1/2, 1], z_1'(t) = 0$ and Construction 2 ensures that $r\,(z_1(t)) = f\,(x_0)$ (z_1 "remembers" the value of $f\,(x_0)$ for $t \in [1/2, 1]$). If we take Construction 1 but now changing $t_0 = 1/2, t_1 = 1$, the function ϕ to $\phi(t) = \phi_2(t) = \theta_j(-\sin 2\pi t)$ and $c = c_{j,2}$ accordingly, the second equation of (14.15) becomes

$$z_2' = c\,(b - z_2)^3\,\phi(t)$$

where $b = f\,(x_0)$. Hence, one has $\left|z_2(1) - f\,(x_0)\right| < \gamma \leq 1/4$. Now, for $t \in [1, 3/2], z_2'(t) = 0$, and Construction 2 ensures that $\tilde{f}\,(r\,(z_2(t))) = f^{[2]}\,(x_0)$ (z_2 "remembers" the value of $f(x_0)$ for $t \in [1, 1+1/2]$). Noting that both $\sin 2\pi t$ and $-\sin 2\pi t$ are periodic with period one, we see that the above procedure can be repeated for all time intervals $[j, j+1]$, where $j \in \mathbb{N}$ Moreover, one has that for any given $x_0 \in \mathbb{N}$

$$r(z_2(t)) = f^{[j]}(x_0) \text{ whenever } t \in [j, j+1/2]$$

for all $j \in \mathbb{N}$

In this sense (14.15) simulates the iteration of the function $f : \mathbb{Z} \to \mathbb{Z}$. A straightforward adaptation of this construction can be applied for the more general case when $f : \mathbb{Z}^k \to \mathbb{Z}^k$, for $k \geq 1$. We then obtain an ODE with $2k$ equations, with a pair of equations simulating each component $f_1, \ldots, f_k$ of f

14.5.5 Robust simulations of Turing machines with polynomial ODEs

We now adapt the construction presented in the previous section to simulate a TM with polynomial ODEs, even under the influence of perturbations. The idea is to iterate the map f_M given by Theorem **??** with ODEs, as described in the previous section. The problem is that the ODE must be polynomial and hence analytic, and therefore we can no longer use the functions θ_k for $1 \leq k \leq \infty$. Instead, we have to use a new variation of this construction.

The main idea of the construction to be presented in this section is the following. Let $\psi : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of a Turing machine M. If we want to iterate ψ with analytic ODEs, using a system similar to (14.15), we cannot allow z_1' and z_2' to be 0 in half-unit intervals cf. Corollary **??**. Instead, we allow them to be very close to zero, which will add some errors to the system (14.15) . Therefore at time $t = 1$ both variables will have values close to $\psi(x_0)$. But Theorem (**??**) shows that there exists some analytic function f_M, robust to errors, that simulates ψ. This allows us to repeat the process an arbitrary number of times, keeping the error under control. We now state the main results of this section.

Theorem 14.6 *Let $\psi : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of a Turing machine M, under the encoding* (**??**) *and let $0 < \varepsilon < 1/4$. There is a polynomial $p_M : \mathbb{R}^{m+4} \to \mathbb{R}^{m+3}$, with $m \in \mathbb{N}$, and a constant $y_0 \in \mathbb{R}^m$ such that the ODE$z' = p_M(t,z)$ simulates M in the following sense: for all $x_0 \in \mathbb{N}^3$ and for all $\bar{x}_0 \in \mathbb{R}^3$ satisfying $\|\bar{x}_0 - x_0\|_\infty \leq \varepsilon$, the solution $z(t)$ of the IVP defined by the previous ODE plus the initial condition $(\bar{x}_0, y_0)$, defined for $t_0 = 0$, satisfies*

$$\left\| z_1(j) - \psi^{[j]}(x_0) \right\|_\infty \leq \varepsilon$$

for all $j \in \mathbb{N}$, where $z \equiv (z_1, z_2)$ with $z_1 \in \mathbb{R}^3$ and $z_2 \in \mathbb{R}^m$

Indeed, we will prove the following robust version of Theorem 14.6.

Theorem 14.7 *Given the conditions of Theorem 14.6, there is a PIVP function $f_M : \mathbb{R}^7 \to \mathbb{R}^6$ and a constant $y_0 \in \mathbb{R}^3$ such that the ODE$z' = f_M(t,z)$ robustly simulates M in the following sense: for all g satisfying $\|g - f_M\|_\infty < 1/2$, there is*

$0<\eta<1/2$ such that for all $(\bar{x}_0,\bar{y}_0)\in\mathbb{R}^6$ satisfying $\|(\bar{x}_0,\bar{y}_0)-(x_0,y_0)\|_\infty\le\varepsilon$, the solution $z(t)$ of

$$z'=g(t,z),\quad z(0)=(\bar{x}_0,\bar{y}_0)$$

satisfies, for all $j\in\mathbb{N}$ and for all $t\in[j,j+1/2]$,

$$\left\|z_1(t)-\psi^{[j]}(x_0)\right\|_\infty\le\eta$$

where $z\equiv(z_1,z_2)$ with $z_1\in\mathbb{R}^3$ and $z_2\in\mathbb{R}^3$

Proof: Let us prove Theorem 4.6 .2 . For simplicity, and without loss of generality, we consider that the function to be iterated, ψ, is a one-dimensional map as in (14.15). We begin with some preliminary results about the introduction of perturbations in (14.15). □

14.5.6 Studying the perturbed targeting equation

(cf. Construction 1) Because the iterating procedure relies on the basic ODE (14.11), we have to study the following perturbed version of (14.11)

$$z'=c(\bar{b}(t)-z)^3\phi(t)+E(t) \tag{14.16}$$

where $|\bar{b}(t)-b|\le\rho$ and $|E(t)|\le\delta$. We take the departure time to be $t_0=0$ and the arrival time to be $t_1=1/2$ as in (14.15) .

Therefore we must require that $\int_0^{1/2}\phi(t)dt>0$, where c satisfies (14.12) and $\gamma>0$ is the targeting error. Let $\bar{z}$ be the solution of this new ODE, with initial condition $\bar{z}(0)=\bar{z}_0$ and let z_+,z_- be the solutions of $z'=c(b+\rho-z)^3\phi(t)+\delta$ and $z'=c(b-\rho-z)^3\phi(t)-\delta$ respectively, with initial conditions $z_+(0)=z_-(0)=\bar{z}_0$. For simplicity denote

$$\begin{aligned} f(t,z)&=c(\bar{b}(t)-z)^3\phi(t)+E(t)\\ f_+(t,z)&=c(b+\rho-z)^3\phi(t)+\delta\\ f_-(t,z)&=c(b-\rho-z)^3\phi(t)-\delta \end{aligned} \tag{14.17}$$

We have that for all $(t,x)\in\mathbb{R}^2$

$$f_-(t,x)\le f(t,x)\le f_+(t,x) \tag{14.18}$$

Since $\bar{z}$ is the solution of the ODE$z'=f(t,z)$ and $z_\pm$ are the solutions of the ODEs$z'=f_\pm(t,z)$ all with the same initial condition $\bar{z}(0)=z_+(0)=z_-(0)=\bar{z}_0$, from (14.18) and a standard differential inequality from the basic theory of ODEs (see e.g. [**?**, Appendix T], it follows that $z_-(t)\le\bar{z}(t)\le z_+(t)$ for all $t\in\mathbb{R}$. Now, if we put upper and lower bounds on z_+ and z_- respectively, we get immediately bounds for $\bar{z}$. Let us study what happens with z_+. For convenience, let y_+ be the solution of

$$y'=c(b+\rho-y)^3\phi(t) \tag{14.19}$$

(i.e. $y'=f_+(t,y)-\delta$), with initial condition $y_+(0)=\bar{z}_0$. Since $f_+(t,x)>f_+(t,x)-\delta$ for all $(t,x)\in\mathbb{R}^2$, we have similarly to the case of $z_-,\bar{z}$, and z_+, that

$$y_+(t)\le z_+(t)\text{ for all }t\in[0,1/2] \tag{14.20}$$

We consider two cases: 1. $\bar{z}_0 \le b+\rho$. Since y_+ is the solution of the targeting equation (14.19), we have from Construction 4.5 .1 and (14.20) that

$$b+\rho-\gamma < y_+(1/2) \Longrightarrow b+\rho-\gamma < z_+(1/2) \tag{14.21}$$

Moreover, since $z_+(0) = \bar{z}_0 \le b+\rho$, we have that if $z_+(t) \ge b+\rho$ for some $t \in (0,1/2)$ then (14.17) gives $z'_+(t) = f_+(t,z) \le \delta$. Therefore $z_+(t)$ cannot grow at a rate bigger than δ when its value exceeds $b+\rho$. This and (14.21) give

$$b+\rho-\gamma < z_+(1/2) < b+\rho+\delta/2$$

2. $\bar{z}_0 > b+\rho$. Since y_+ is solution of the targeting equation (14.19), we have from Construction 1 and (14.20) that

$$b+\rho < y_+(t) < z_+(t) \text{ for all } t \in [0,1/2]$$

This condition then gives $c\left(b+\rho-y_+\right)^3 \phi(t)+\delta > c\left(b+\rho-z_+\right)^3 \phi(t)+\delta$ which together with (14.19) implies

$$\delta > z'_+(t) - y'_+(t) \text{ for all } t \in [0,1/2]$$

Integrating the last equation, we have

$$\frac{\delta}{2} > z_+(1/2) - y_+(1/2) \Longrightarrow \frac{\delta}{2} + y_+(1/2) > z_+(1/2)$$

The latter inequality plus the fact that $b+\rho < y_+(t) < b+\rho+\gamma$, where y_+ is solution of (14.19), yield that

$$b+\rho < z_+(1/2) < b+\rho+\gamma+\frac{\delta}{2}$$

Now that we have studied both cases $\bar{z}_0 \le b+\rho$ and $\bar{z}_0 > b+\rho$, we conclude that

$$b+\rho-\gamma < z_+(1/2) < b+\rho+\gamma+\frac{\delta}{2}$$

A similar analysis can be performed for $z_-(1/2)$, ultimately yielding

$$|\bar{z}(1/2) - b| < \rho+\gamma+\frac{\delta}{2} \tag{14.22}$$

14.5.7 Removing the θ_j 's from (14.15)

We must remove the θ_j 's in two places: in the function r and in the terms $\theta_j(\pm\sin 2\pi t)$. Since in (14.15) we are using an extension $\tilde{f}:\mathbb{R}\to\mathbb{R}$ (actually, $\tilde{f} \equiv f_M : \mathbb{R}^3 \to \mathbb{R}^3$, but we consider the one-dimensional case for simplicity) of $f:\mathbb{N}\to\mathbb{N}$ $\left(\equiv \psi : \mathbb{N}^3 \to \mathbb{N}^3\right)$ which is robust to perturbations, we no longer need the corrections performed by r. On the other hand we cannot use this technique to treat the terms $\theta_j(\pm\sin 2\pi t)$. We need to substitute $\phi(t) = \theta_j(\sin 2\pi t)$ by an analytic (PIVP) function $\zeta:\mathbb{R}\to\mathbb{R}$ with the following ideal behavior:

1. (i) ζ has period 1
2. (ii) $\zeta(t) = 0$ for $t \in [1/2, 1]$
3. (iii) $\zeta(t) \geq 0$ for $t \in [0, 1/2]$ and $\int_0^{1/2} \zeta(t)dt > 0$

Of course, conditions (ii) and (iii) are incompatible due to Proposition 2.5.4. Instead, we approach ζ using a function ζ_ϵ, where $\epsilon > 0$. This function must satisfy the following conditions:

1. (ii)' $|\zeta_\epsilon(t)| \leq \epsilon$ for $t \in [1/2, 1]$
2. (iii)' $\zeta_\epsilon(t) \geq 0$ for $t \in [0, 1/2]$ and $\int_0^{1/2} \zeta_\epsilon(t)dt > I > 0$, where I is independent of ϵ

Our idea to define such a function ζ_ϵ is to use the function l_2 introduced in Proposition **??** Then define

$$\zeta_\epsilon(t) = l_2(\vartheta(t), 1/\epsilon) \tag{14.23}$$

where $\epsilon > 0$ is the precision up to which ζ_ϵ should approximate 0 in the interval $[1/2, 1]$ and $\vartheta : \mathbb{R} \to \mathbb{R}$ is an elementary periodic function of period 1 satisfying the following conditions: (a) $|\vartheta(t)| \leq 1/4$ for $t \in [1/2, 1]$

(b) $\vartheta(t) \geq 3/4$ for $t \in (a, b) \subseteq (0, 1/2)$. Notice that Proposition 4.2 .5 and (a) ensure that $|\zeta_\epsilon(t)| < \epsilon$ for $t \in [1/2, 1]$, i.e. they ensure (ii)', and that Proposition 4.2 .5 and (b) ensure $|\zeta_\epsilon(t)| > 1 - \epsilon$ for $t \in (a, b)$ which gives $\int_0^{1/2} \zeta_\epsilon(t) \geq (1-\epsilon)(b-a) > 3(b-a)/4$ for $\epsilon < 1/4$, which yields (b) (remark that for all $(t, x) \in \mathbb{R}^2$, $l_2(t, x) > 0$ and thus $\zeta_\epsilon(t) > 0$ for all $t \in \mathbb{R}$). It is not difficult to see that one can pick $\vartheta : \mathbb{R} \to \mathbb{R}$ as

$$\vartheta(t) = \frac{1}{2}\left(\sin^2(2\pi t) + \sin(2\pi t)\right) \tag{14.24}$$

since it satisfies all the conditions imposed above (e.g. $a = 0.16$ and $b = 0.34$). Hence, $\theta_j(\sin 2\pi t)$ will be replaced by the PIVP function $\zeta_\epsilon(t) = l_2(\vartheta(t), 1/\epsilon)$, where ϑ is given by (14.24). Similarly, $\theta_j(-\sin 2\pi t)$ will be replaced by the PIVP function $\zeta_\epsilon(-t)$

14.5.8 Performing Construction 3 with PIVP functions

We are now ready to perform a simulation of an integer map with a system similar to (14.15) but only using PIVP (and hence analytic) functions. Choose a targeting error $\gamma > 0$ such that

$$2\gamma + \delta/2 \leq \varepsilon < 1/4 \tag{14.25}$$

where $\delta = \left\| g - f_M \right\|_\infty < 1/2$ is the maximum amplitude of the perturbations that can affect our system of ODEs (we suppose, without loss of generality, that $\delta/2 < \varepsilon$) and take the following system of ODEs

$$\begin{aligned} z_1' &= c_1 \left(f_M \circ \sigma^{[m]}(z_2) - z_1\right)^3 \zeta_{\epsilon_1}(t) \\ z_2' &= c_2 \left(\sigma^{[n]}(z_1) - z_2\right)^3 \zeta_{\epsilon_2}(-t) \end{aligned} \tag{14.26}$$

with initial conditions $z_1(0) = z_2(0) = \bar{x}_0$, where σ is the error-contracting function defined in (**??**) and $c_1, c_2, m, n, \epsilon_1$, and ϵ_2 are still to be defined. We would like that (14.26) satisfies the following property: on $[0,1/2]$,

$$\left|z_2'(t)\right| \le \gamma \tag{14.27}$$

This can be achieved by taking $\epsilon_2 = \gamma/K$, where K is a bound for $c_2\left(\sigma^{[n]}(z_1) - z_2\right)^3$ in the interval $[0,1]$. Since $|x|^3 \le x^4+1$ for all $x \in \mathbb{R}$, we can take $\epsilon_2 = \gamma/c_2\left(\sigma^{[n]}(z_1) - z_2\right)^{-4} +$ γ/c_2 Now notice that $z_2(0)$ has an error bounded by ε. This plus (14.27) and the fact that z_2' might be subjected to perturbations of amplitude not exceeding δ, imply that

$$|z_2(t) - x_0| \le \varepsilon + (\delta+\gamma)/2 = \eta < 1/2 \quad \text{for } t \in [0,1/2] \tag{14.28}$$

Therefore, for m satisfying $\sigma^{[m]}(\eta) < \gamma$, we have that $\left|\sigma^{[m]}(z_2(t)) - x_0\right| < \gamma$ for all $t \in [0,1/2]$. Hence, from the study of the perturbed targeting equation (14.16), where $\phi(t) = \zeta_{\epsilon_1}(t)$ and c_1 is obtained accordingly, we have (take $\rho = \gamma$ and consider (14.25))

$$\left|z_1(1/2) - \psi(x_0)\right| < 2\gamma + \frac{\delta}{2} \le \varepsilon \tag{14.29}$$

For the interval $[1/2,1]$ the roles of z_1 and z_2 are switched. Similarly to the reasoning done for z_2 on $[0,1/2]$, take $\epsilon_1 = \gamma/c_1\left(f_M \circ \sigma^{[m]}(z_2) - z_1\right)^{-4} + \gamma/c_1$ so that on $[0,1/2]$

$$\left|z_1'(t)\right| \le \gamma$$

From this inequality, (14.29) and the fact that z_2' might be subjected to perturbations of amplitude not exceeding δ, we get that

$$\left|z_1(t) - \psi(x_0)\right| \le \varepsilon + (\delta+\gamma)/2 = \eta < 1/2 \quad \text{for } t \in [1/2,1]$$

Therefore, for $n = m$, we have $\left|\sigma^{[n]}(z_1(t)) - \psi(x_0)\right| < \gamma$ for all $t \in [1/2,1]$. Hence, from the study of the perturbed targeting equation (14.16), where $\phi(t) = \zeta_{\epsilon_2}(t)$ and c_2 is obtained accordingly, we have

$$\left|z_2(1) - \psi(x_0)\right| < 2\gamma + \frac{\delta}{2} \le \varepsilon$$

Now we can repeat the procedure for intervals $[1,2], [2,3]$, etc. to conclude that for all $j \in \mathbb{N}$ and for all $t \in [j, j+1/2]$

$$\left|z_1(t) - \psi^{[j]}(x_0)\right| \le \eta$$

Moreover, z_1 is defined as the solution of an ODE written in terms of PIVP functions. As a corollary, we prove Theorem 14.6.

Proof:[of Theorem 14.6] From the previous proof, it follows that

$$\left|z_1(t) - \psi^{[j]}(x_0)\right| \le \eta < 1/2$$

Let k be an integer such that $\sigma^{[k]}(\eta) < \varepsilon$. Then the function y_1 defined by $y_1 = \sigma^{[k]}(z_1(t))$ is also a PIVP function (see Theorem **??**) satisfying

$$\left|y_1(t) - \psi^{[j]}(x_0)\right| \le \varepsilon$$

for all $j \in \mathbb{N}$ and for all $t \in [j, j+1/2]$. □

14.5.9 Proof (sketch)

We adapt the construction in [Branicky, 1995] to simulate the iteration of the transition function of a TM with ODEs, using Theorem 14.2 to generalize Branicky's construction to analytic and robust flows. To iterate a function θ we use a pair of functions to control the evolution of two "simulation" variables z_1 and z_2. Both simulation variables have values close to x_0 at $t = 0$. The first variable is iterated during half of an unit period while the second remains approximately constant (its derivative is kept close to zero by a control function that involves our error-contracting function l_2). Then, the first variable remains controlled during the following half unit period of time and the second variable is brought up close to it. Therefore, at time $t = 1$ both variables have values close to $\theta(x_0)$. Theorem 14.2 shows that there exists some analytic function robust to errors that simulates θ. This allow us to repeat the process an arbitrary number of times, keeping the error under control.

We begin with some preliminary results. There exists an ODE whose solution can be as close as desired to an arbitrary fixed value $b \in R$ at $t = 1/2$, for any initial condition at $t = 0$. Let $\varphi : \mathbb{R} \to \mathbb{R}^+$ be some function. For an arbitrary error $\gamma > 0$ we define a perturbed version, where we allow an error $\rho \geq 0$ on b and a perturbation term bounded by $\delta \geq 0$:

$$z' = -c(z - \overline{b}(t))^3 \phi(t) + E(t), \quad \text{with } c \geq \left(2\gamma^2 \int_0^{1/2} \phi(t)dt\right)^{-1}. \tag{14.30}$$

where $|\overline{b}(b) - b| \leq \rho$ and $|E(t)| \leq \delta$. Using the theory of ODEs, we can conclude that $|z(\frac{1}{2}) - b| < \gamma + \rho + \delta/2$ regardless to the initial condition at $t = 0$.

For the control functions mentioned above, we use $s : \mathbb{R} \to [-1/8, 1]$ defined by

$$s(t) = \frac{1}{2}\left(\sin^2(2\pi t) + (\sin(2\pi t)\right).$$

On $s \in [0, 1/2]$, s ranges between 0 and 1 and on $[1/2, 1]$, s ranges between $-1/8$ and 0.

We can now present the proof of the theorem. Let M be some Turing machine, let f_M be a map satisfying the conditions of Theorem 14.3 (replacing ϵ by γ), and let $\overline{x}_0 \in \mathbb{R}^3$ be an approximation, with error ϵ, of some initial configuration x_0. Take also $\delta < 1/2$ and $\gamma > 0$ such that $2\gamma + \delta/2 \leq \epsilon < 1/2$ (we suppose, without loss of generality, that $\delta/2 < \epsilon$). This condition will be needed later.

Consider the system of differential equations $z' = g_M(z, t)$ given by

$$\begin{aligned} z_1' &= c_1(z_1 - f_M \circ \sigma^{[m]}(z_2))^3 \varphi_1(t, z_1, z_2), \\ z_2' &= < c_2(z_2 - \sigma^{[n]}(z_1))^3 \varphi_2(t, z_1, z_2) \end{aligned}$$

with initial conditions $z_1(0) = z_2(0) = x_0$, where

$$\begin{aligned} \varphi_1(t, z_1, z_2) &= l_2\left(s(t), \frac{c_1}{\gamma}(z_1 - f_M \circ \sigma^{[m]}(z_2))^4 + \frac{c_1}{\gamma} + 10\right) \\ \varphi_2(t, z_1, z_2) &= l_2\left(s(-t), \frac{c_2}{\gamma}(z_2 - \sigma^{[m]}(z_1))^4 + \frac{c_2}{\gamma} + 10\right) \end{aligned}$$

Because we want to show that the ODE $z' = g_M(z,t)$ simulates M in a robust manner, we also assume that an error of amplitude not exceeding δ is added to the right side of the equations in (14.31). Our simulation variables are z_1, z_2 and the control functions are φ_1, φ_2. Since φ_1, φ_2 are analytic they cannot be constant on any open interval as in [Branicky, 1995]. However, our construction guarantees that one of the control functions is kept close to zero, while the other one reaches a value close to 1. For instance, on $[0,1/2]$, $|s(-t)| \leq 1/8$ and, by Lemma (14.4), φ_2 is therefore less than $\gamma(c_2 \left\| z_2 - \sigma^{[n]}(z_1) \right\|^3)^{-1}$. This guarantees that z_2' is sufficiently small on $[0,1/2]$ and, therefore,

$$\left\| z_2(\frac{1}{2}) - x_0 \right\| < (\gamma+\delta)/2 + \epsilon < \frac{1}{2}$$

Hence, for m large enough $\left\|\sigma^{[m]}(z_2) - x0\right\| < \gamma$. Moreover, on some subinterval of $[0,1/2]$, $s(t)$ is close to 1 and therefore φ_1 is also close to 1. Thus, the behavior of z_1 is given by (14.30) and $\| z_1(1) - \theta(x_0) \| < 2\gamma + \delta/2 \leq \epsilon$.

Now, for interval $[1/2,1]$ the roles of z_1 and z_2 are switched. One concludes that if $n \in \mathbb{N}$ is chosen so that $\sigma^{[n]}(5\gamma/2+\delta) < \gamma$, then $\left\| z_2(1) - f_M(x_0) \right\| \infty < 2\gamma + \delta/2 \leq \epsilon$. We can repeat this process for z_1 and z_2 on subsequent intervals, which shows that for $j \in \mathbb{N}$, if $t \in [j, j+1]$ then $\left\| z_2(t) - \theta^{[j]}(x_0) \right\| \leq \epsilon$ as claimed.

14.6 Discrete time simulation: Graça-Campagnolo-Buescu's Theorem 2

Theorem 14.8 (Graça-Campagnolo-Buescu's Theorem 2 [Graça et al., 2005a]) $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ *be the transition function of a Turing machine* M*, under the encoding* $\gamma^{TM}_{\mathbb{N}^3}$ *described above and let* $0 < \epsilon < 1/4$*. Then there is an analytic function* $z : \mathbb{R}^4 \to \mathbb{R}^3$ *with the following property:*

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

14.6.1 Proof of Graça-Campagnolo-Buescu's Theorem 2

Notice that all the functions we use in the proof of 14.5 above are analytic. Moreover, note that if we apply the error-contracting function σ to z_1 we can make the error arbitrarily small. Therefore, Theorem 14.5 implies Theorem 14.8.

14.7 Continuous time simulation: $\gamma^{TM}_{(0,1)^2}$

Now we use the previous construction to simulate a Turing machine on a compact set $X = (-1,1)^3$. This is a trick used in [Bournez et al., 2013], based on a change of variable.

If ξ is a solution of $y' = g_M(y)$ simulating M on $\mathbb{R}^6$, we can pick $\tilde{\xi} = \frac{2}{\pi}\arctan\xi$ (and hence $\xi = \tan\left(\frac{\tilde{\xi}\pi}{2}\right)$) as the corresponding simulation of M on $(-1,1)^6$. In general

$$\tilde{\xi}' = (\frac{2}{\pi}\arctan\xi)' = \frac{2}{\pi}\frac{1}{1+\xi^2}\xi' = \frac{2}{\pi}\frac{1}{1+\xi^2}g_M(\xi) \implies \tag{14.31}$$

$$\tilde{\xi}' = \frac{2}{\pi}\frac{1}{1+\xi^2}g_M(\xi) = \frac{2}{\pi}\frac{1}{1+\tan^2\left(\frac{\tilde{\xi}\pi}{2}\right)}g_M\left(\tan\left(\frac{\tilde{\xi}\pi}{2}\right)\right) = f_M(\tilde{\xi})$$

where

$$f_M(x) = \frac{2}{\pi}\frac{1}{1+\tan^2\left(\frac{x\pi}{2}\right)}g_M\left(\tan\left(\frac{x\pi}{2}\right)\right).$$

Hence, the system $y' = f_M(y)$ simulates M on X, with input w coded by $\gamma^{TM}_{(0,1)^2}$. Moreover, robustness among states still exists, and the simulation of M can be carried out if the states are not perturbed more than

$$\tilde{\varepsilon} = \arctan(m+\varepsilon) - \arctan(m) \tag{14.32}$$

14.8 Discrete time simulation with a GPAC: $\gamma^{TM}_{\mathbb{N}^3}$

More generally, if Turing machine M has l tapes, then its transition function is defined over $\mathbb{N}^{2l+1}$ and also admits a closed-form robust extension to $\mathbb{R}^{2l+1}$.

The following result is an adaptation of Theorem [**?**] from [Graça et al., 2005b] and shows that GPACs can iterate the transition function of a given Turing machine. It is stated in [Bournez et al., 2007]: basically, follow the idea above, and use the stability properties of GPAC generable functions to conclude that all invoved functions correspond to functions computable by polynomial ordinary differential equations with suitable initial values.

Proposition 14.1 ([Bournez et al., 2007, Graça et al., 2005b]) *Suppose that $\psi_M : \mathbb{N}^{2l+1} \to \mathbb{N}^{2l+1}$ is the transition function of a Turing machine M, under the $\gamma^{TM}_{\mathbb{N}^3}$, $x_0 \in \mathbb{N}^{2l+1}$ represents an initial configuration and $\varepsilon, \delta > 0$ are constants satisfying $\varepsilon + \delta < 1/2$. Then there is a computable polynomial p and some computable value $\alpha \in \mathbb{R}^n$*

$$z' = p(z,t), \qquad z(0) = (\tilde{x}_0, \alpha)$$

such that for all $\tilde{x}_0 \in \mathbb{R}^{2l+1}$ satisfying $\|\tilde{x}_0 - x_0\|_\infty \le \varepsilon$, one has[a]

$$\left\| z_1(t) - \psi_M^{[j]}(x_0) \right\|_\infty \le \delta.$$

for all $j \in \mathbb{N}$ and for all $t \in [j, j+1/2]$.

[a]For simplicity, we denote the solution z of the initial-value problem by (z_1, z_2), where $z_1 \in \mathbb{R}^{2l+1}$ and $z_2 \in \mathbb{R}^n$.

If there exists some computable value $\alpha \in \mathbb{R}^n$ such that $z' = p(z,t)$ has the properties described in Proposition 14.1, we say that the GPAC $z' = p(z,t)$ simulates the Turing machine M on input x.

14.9 Simulating Type-2 machines with GPACs

From previoius results, we know how to simulate a Turing machine. However, the error of the output is bounded by some fixed quantity $\varepsilon > 0$, whereas in Type-2 machines we would like that the output is given with error bounded by 2^{-n}, where n is one of the inputs of the machine. The next theorem shows how this can be done with a GPAC.

Theorem 14.9 ([Bournez et al., 2007]) *Let $f : [a,b] \to \mathbb{R}$ be a computable function. Then there exists a GPAC and some index i such that if we set the initial conditions $(x, \bar{n}) \in [a,b] \times \mathbb{R}$, where $|\bar{n} - n| \leq \varepsilon < 1/2$, with $n \in \mathbb{N}$, there exists some $T \geq 0$ such that the output y_i of the GPAC satisfies $|y_i(t) - f(x)| \leq 2^{-n}$ for all $t \geq T$.*

14.9.1 GPAC-computability

Actually, the following is true.

Theorem 14.10 (Bournez-Campagnolo-Graça-Hainry's Theorem [Bournez et al., 2007]) *Let a and b be computable reals. A function $f : [a,b] \to \mathbb{R}$ is computable iff it is GPAC-computable.*

Bibliography

[Antsaklis, 2000] Antsaklis, P. J., editor (2000). *Proceedings of the IEEE*, volume 88.

[Arnold, 1978] Arnold, V. I. (1978). *Ordinary Differential Equations*. MIT Press.

[Asarin, 1995] Asarin, E. (1995). Chaos and undecidabilty (draft). Avalaible in `http://www.liafa.jussieu.fr/~asarin/`.

[Asarin and Bouajjani, 2001] Asarin, E. and Bouajjani, A. (2001). Perturbed Turing machines and hybrid systems. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS-01)*, pages 269–278, Los Alamitos, CA. IEEE Computer Society Press.

[Asarin and Maler, 1998] Asarin, E. and Maler, O. (1998). Achilles and the tortoise climbing up the arithmetical hierarchy. *Journal of Computer and System Sciences*, 57(3):389–398.

[Asarin et al., 1995] Asarin, E., Maler, O., and Pnueli, A. (1995). Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1):35–65.

[Asarin et al., 2001] Asarin, E., Schneider, G., and Yovine, S. (2001). On the decidability of the reachability problem for planar differential inclusions. In Benedetto, M. D. D. and Sangiovanni-Vincentelli, A. L., editors, *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 28-30, 2001, Proceedings*, volume 2034 of *Lecture Notes in Computer Science*, pages 89–104. Springer.

[Bournez, 1999] Bournez, O. (1999). *Complexité Algorithmique des Systèmes Dynamiques Continus et Hybrides*. Phd thesis, Ecole Normale Supérieure de Lyon.

[Bournez et al., 2007] Bournez, O., Campagnolo, M. L., Graça, D. S., and Hainry, E. (2007). Polynomial differential equations compute all real computable functions on computable compact intervals. *Journal of Complexity*, 23(3):317–335.

[Bournez et al., 2013] Bournez, O., Graça, D. S., and Hainry, E. (2013). Computation with perturbed dynamical systems. *Journal of Computer System Science*, 79(5):714–724.

[Branicky, 1995] Branicky, M. S. (1995). Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100.

[Brattka et al., 2008] Brattka, V., Hertling, P., and Weihrauch, K. (2008). *New Computational Paradigms. Changing Conceptions of What is Computable*, chapter A tutorial on computable analysis. Springer-Verlag, New York.

[Brockett, 1989] Brockett, R. W. (1989). Smooth dynamical systems which realize arithmetical and logical operations. In Nijmeijer, H. and Schumacher, J. M., editors, *Three Decades of Mathematical Systems Theory*, volume 135 of *Lecture Notes in Computer Science*, pages 19–30. Springer.

[Brockett, 1991] Brockett, R. W. (1991). Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra and its Applications*, 146:79–91.

[Bush, 1931] Bush, V. (1931). The differential analyzer. A new machine for solving differential equations. *J. Franklin Inst.*, 212:447–488.

[Campagnolo, 2001] Campagnolo, M. L. (2001). *Computational complexity of real valued recursive functions and analog circuits*. PhD thesis, IST, Universidade Técnica de Lisboa.

[Campagnolo et al., 2000a] Campagnolo, M. L., Moore, C., and Costa, J. F. (2000a). Iteration, inequalities, and differentiability in analog computers. *Journal of Complexity*, 16(4):642–660.

[Campagnolo et al., 2000b] Campagnolo, M. L., Moore, C., and Costa, J. F. (2000b). Iteration, inequalities, and differentiability in analog computers. *J. Complexity*, 16(4):642–660.

[Ceraens and Viksna, 1996] Ceraens, K. and Viksna, J. (1996). Deciding reachability for planar multi-polynomial systems. In *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, page 389. Springer-Verlag.

[Coddington and Levinson, 1972] Coddington, E. E. and Levinson, N. (1972). *Theory of Ordinary Differentiel Equations*. McGraw-Hill.

[Filippov, 1988] Filippov, A. (1988). *Differential equations with discontinuous right-hand sides*. Kluwer Academic Publishers.

[Graça, 2004] Graça, D. S. (2004). Some recent developments on Shannon's General Purpose Analog Computer. *Mathematical Logic Quaterly*, 50(4-5):473–485.

[Graça, 2007] Graça, D. S. (2007). *Computability with Polynomial Differential Equations*. PhD thesis, Instituto Superior Técnico.

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

[Graça et al., 2005b] Graça, D. S., Campagnolo, M. L., and Buescu, J. (2005b). Robust simulations of Turing machines with analytic maps and flows. In Cooper, S. B., Löwe, B., and Torenvliet, L., editors, *CiE 2005: New Computational Paradigms*, LNCS 3526, pages 169–179. Springer.

[Graça et al., 2007] Graça, D. S., Campagnolo, M. L., and Buescu, J. (2007). Computability with polynomial differential equations. *Advances in Applied Mathematics.* To appear.

[Graça and Costa, 2003] Graça, D. S. and Costa, J. F. (2003). Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5):644–664.

[Hirsch et al., 2003] Hirsch, M. W., Smale, S., and Devaney, R. (2003). *Differential Equations, Dynamical Systems, and an Introduction to Chaos.* Elsevier Academic Press.

[Jones, 1997] Jones, N. (1997). *Computability and complexity, from a programming perspective.* MIT press.

[Ko, 1991] Ko, K.-I. (1991). *Complexity Theory of Real Functions.* Progress in Theoretical Computer Science. Birkhaüser, Boston.

[Koiran et al., 1994] Koiran, P., Cosnard, M., and Garzon, M. (1994). Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132(1-2):113–128.

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

[Kurganskyy and Potapov, 2005] Kurganskyy, O. and Potapov, I. (2005). Computation in one-dimensional piecewise maps and planar pseudo-billiard systems. In Calude, C., Dinneen, M. J., Paun, G., Pérez-Jiménez, M. J., and Rozenberg, G., editors, *Unconventional Computation, 4th International Conference, UC 2005, Sevilla, Spain, October 3-7, 2005, Proceedings*, volume 3699 of *Lecture Notes in Computer Science*, pages 169–175. Springer.

[Lipshitz and Rubel, 1987] Lipshitz, L. and Rubel, L. A. (1987). A differentially algebraic replacement theorem, and analog computability. *Proceedings of the American Mathematical Society*, 99(2):367–372.

[Loff et al., 2007] Loff, B., Costa, J. F., and Mycka, J. (2007). Computability on reals, infinite limits and differential equations. *Applied Mathematics and Computation*, 191(2):353–391. To appear.

[Moore, 1990] Moore, C. (1990). Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64(20):2354–2357.

[Moore, 1991] Moore, C. (1991). Generalized shifts: unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4(3):199–230.

[Moore, 1998] Moore, C. (1998). Finite-dimensional analog computers: Flows, maps, and recurrent neural networks. In Calude, C. S., Casti, J. L., and Dinneen, M. J., editors, *Unconventional Models of Computation (UMC'98)*. Springer.

[Odifreddi, 1992] Odifreddi, P. (1992). *Classical Recursion Theory*, volume 125 of *Studies in Logic and the foundations of mathematics*. North-Holland.

[Pour-El, 1974] Pour-El, M. B. (1974). Abstract computability and its relations to the general purpose analog computer. *Trans. Amer. Math. Soc.*, 199:1–28.

[Pour-El and Richards, 1989] Pour-El, M. B. and Richards, J. I. (1989). *Computability in Analysis and Physics*. Springer.

[Rogers Jr., 1987] Rogers Jr., H. (1987). *Theory of Recursive Functions and Effective Computability*. MIT Press.

[Rubel, 1989] Rubel, L. A. (1989). A survey of transcendentally transcendental functions. *American Mathematical Monthly*, 96(9):777–788.

[Ruohonen, 1993] Ruohonen, K. (1993). Undecidability of event detection for ODEs. *Journal of Information Processing and Cybernetics*, 29:101–113.

[Ruohonen, 1997] Ruohonen, K. (1997). Undecidable event detection problems for ODEs of dimension one and two. *Theoretical Informatics and Applications*, 31(1):67–79.

[Shannon, 1941] Shannon, C. E. (1941). Mathematical theory of the differential analyser. *Journal of Mathematics and Physics MIT*, 20:337–354.

[Siegelmann and Sontag, 1995] Siegelmann, H. T. and Sontag, E. D. (1995). On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150.