Course Notes for 2.33.1: Static Undecidability

Olivier Bournez

Version of October 28, 2015

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

Chapter 2

Computer Algebra: Richardson's Theorem

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

2.1 Richardson 68's Theorem

2.1.1 The theorem

Theorem 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*[1] *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.*

(remark: the theorem is not stated here with minimal hypotheses. It is stated in its original form. We will see in chapter 3 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).

[1] 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.

2.1.2 Richardson 68's Theorem (continued)

Theorem 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[2] 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.*

End of Course 5

2.1.3 Proof Idea of First Theorem

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.

[2] 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.

- 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.

2.1.4 Proof Idea of Second Theorem

A formal proof will be given in two chapters.
Basically,
$\exists x \; G(n, x) < 1$ iff $|G(n, x) - 1| - (G(n, x) - 1) \equiv 0$.

2.2 Consequences

From a Computer Algebra point of view:

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

2.3 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 3

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 4:

Theorem 3 *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.*

3.1 From Integers to Reals

Lemma 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 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 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 $P \in \mathbb{Z}[X_1, \ldots, X_k]$, then the existence of integer solutions to P is equivalent to the existence of real solutions to the equation in $k+3$ variables

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

□

3.2 From Equalities to Inequalities

Lemma 3 *For all $f \in \Sigma_k$, there exists a polynomial $g \in \mathbb{Z}[X_1, \ldots, X_k]$ such that*

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

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 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 $f(x_1, \ldots, x_k) < 1$ has a real solution.*

Proof: Starting from $P \in \mathbb{Z}[X_1, \ldots, X_k]$, we consider previous function Φ. We consider $M^2\Phi \leq 1$ with M still to be determined. If $(x, y, z, y_1, \ldots, y_k)$ 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 $M^2\Phi \leq 1$ 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 accroissements 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 $M^2\Phi \leq 1$ 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 $M^2\Phi \leq 1$:

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

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

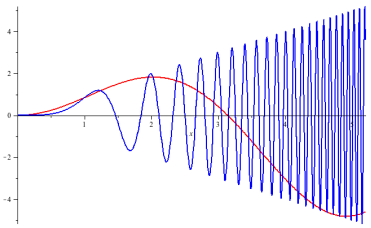
3.3 From Multivariate to Univariate

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

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



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:

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

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

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

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

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

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 *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 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 5 *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. □

3.4 Application to Simplification

Theorem 6 *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 4

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 [3].

4.1 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

$(a_1, \ldots, a_n) \in A$ 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.

4.2 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 $a_0, a_1, \ldots, a_n \in \{0,1\}^{n+1}$ can be encoded by integer $\sum_{i=0}^{n} a_i 2^i$.

We will use the following result:

Lemma 6 *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 $a_0, a_1, \ldots, a_n \in \{0,1\}^{n+1}$ and $b_0, b_1, \ldots, b_n \in \{0,1\}^{n+1}$ two sequences of $n+1$ bits. we consider the encoding a and b of these sequences: we write $a \ll b$ for $\forall 0 \leq i \leq n, \; a_i \leq b_i$.

We will admit the following result:

Lemma 7 *$a \ll b$ if and only if $\binom{b}{a} = \frac{b!}{(b-a)!a!}$ is odd.*

Lemma 8 *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 6. We have m odd if and only if $\exists x \; m = 2x + 1$, hence the result follows after substitution. □

4.3 Davis-Putnam-Robinson's theorem

This section is devoted to prove the following result:

Theorem 7 *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 $i \in \{1, \ldots, n\}$, 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 3 *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

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

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, ..., 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

$$b = 2^{x+y+n}, \tag{4.1}$$

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

$$1 + bU = U + b^y : \tag{4.2}$$

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 $1 \leq l \leq n$. We write

$$i_l \ll U : \tag{4.3}$$

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

$$U = \sum_{i=1}^{n} i_l : \tag{4.4}$$

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

$$x_j \ll (b/2 - 1)U, \tag{4.5}$$

for $j = 1, 2$.

By adding equation

$$1 \ll i_1, \tag{4.6}$$

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

$$i_n = b^{y-1}, \tag{4.7}$$

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

$$bi_l \ll i_{l+1} \tag{4.8}$$

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

$$bi_l \ll i_j + i_{l+1}, \tag{4.9}$$

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:

$$x_1 = x + b(x_1 + \sum_{l \in A(1)} i_l - \sum_{l \in S(1)} i_l) \tag{4.10}$$

and

$$x_2 = b(x_j + \sum_{l \in A(2)} i_l - \sum_{l \in S(2)} i_l), \tag{4.11}$$

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

$$bi_l \ll i_{l+1} + U - 2x_c, \tag{4.12}$$

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 (4.12) 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.

4.4 Matiyasevich's theorem

We will admit the following result, due to Matiyasevich:

Theorem 8 (Matiyasevich) *The set of integers u, v, w such that $u = v^w$ is Diophantine.*

We get:

Corollary 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. □

4.5 On the impossibility of solving Diophantine equations

Corollary 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 5

Static Undecidability Results

We let the reader understand that using these results or similar constructions, one can prove the undecidability of several decision problems for dynamical systems.

In particular:

- Determining whether a polynomial dynamical system has a Hopf bifurcation is undecidable [1].
- "Dynamical Systems where proving chaos is equivalent to proving Fermat's conjecture" [2].

We ask our reader to read the papers [2] and [1].

Chapter 6

Bibliography

6.1 References

The discussion on Diophantine equations is based on (some parts copied from) excellent book [3].

Chapter 3 is based on [4], and on some exercices given in 2011 in course INF423 of Ecole Polytechique, based on a "Petite Classe" written by Bruno Salvy.

Bibliography

[1] NCA Costa and FA Doria. Undecidable hopf bifurcation with undecidable fixed point. *International Journal of Theoretical Physics*, 33(9):1885–1903, 1994.

[2] N.C.A. Costa, FA Doria, and A.F.F. Amaral. Dynamical system where proving chaos is equivalent to proving fermat's conjecture. *International journal of theoretical physics*, 32(11):2187–2206, 1993.

[3] N. Jones. *Computability and complexity, from a programming perspective*. MIT press, 1997.

[4] Daniel Richardson. Some undecidable problems involving elementary functions of a real variable. *Journal of Symbolic Logic*, 33(4):514–520, 1968.