Review Lecture 1

Computing over a ring (commutative with unit, without zero divisors) or field.

May be ordered (eg $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{R}$) or unordered (eg $\mathbb{Z}_2$ or $\mathbb{C}$)

First Main Goal. Explain and prove**: Theorem** (lb, Shub, Smale, '89) For fields, R.
HN_R is the canonical NP-complete over R. So, **P = NP $\Leftrightarrow$ $HN_R \in$ P over R.**

HN_R (Hilbert Nullstellensatz over R) is the problem of deciding, given any finite polynomial system over R, whether or not it is solvable over R.

If R = $\mathbb{Z}_2$, then this is the classical NP-completeness result.

Finite dimensional machine M over (R, <) or (R, =)

3 Spaces: Input space $I_M = R^n$, **State space** $S_M = R^m$, **Output space**, $O_M = R^l$

M is a **finite directed graph** with **4 types of nodes**: **Input, Computation, Branch and Output,** with associated maps. Let ***N*** be the set of nodes and assume a fixed labeling of nodes:
N **= { 1(input node), 2, …., N(output node) }**

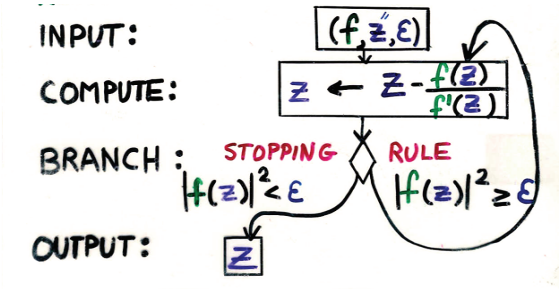
The Computing Endomorphism: H = (β, g): *NxS* → *NxS*
(node, state) → (next node, next state)

Next node map β: *NxS* → *N*
For η an input or computation node: $\beta(\eta, y) = \beta_\eta$ is the unique next node (independent of y).
For η a branch node: $\beta(\eta, y) = \beta_\eta^+$ if $y_1 \geq 0$ (= 0) and $\beta(\eta, y) = \beta_\eta^-$ if $y_1 < 0$ ($\neq 0$)
For η an output node: let $\beta(\eta, y) = \eta$

Next state map g: *NxS* → *S*
For η a computation node, $g(\eta, y) = g_\eta(y)$ where g_η: $S \to S$ is a poly map (rat map if R is a field);
For all other nodes, g(η, y) = y.

Will also assume linear maps, **I**: $I_M \to S_M$ and **O**: $S_M \to O_M$
Define: D_M, the **degree of M**; **m** = the **dimension of M**; **C_M**, the **constants of M**.
Will assume M never divides by 0 (by checking before a computation node).

Think of the Newton "machine" :



$N_f(z)=z-f(z)/f'(z)=(zf'(z)-f(z))/f'(z)$ is a rational function.

Alternate definition in terms of algebraic circuits (non-uniform) with constants and Turing machines (uniform).

All key concepts are defined using the computing endomorphism:

Computations, computation paths, halting time, input-output maps, computable functions, time-T halting sets, halting sets (semi-decidable sets), …

For x in the input space R^n, let $z^0 = (\eta^0, x^0)$ where $\eta^0 = 1$ and $x^0 = I(x)$.
The **computation** of M with input x is just the **orbit** of z^0 under iterations of H:

$$\mathbf{z^0 = (1, x^0), z^1, \ldots, z^{k+1} = H(z^k) = H(\eta^k, x^k) = (\eta^{k+1}, x^{k+1}), \ldots}$$

The **computation path** is the sequence of 1st coordinates: $\mathbf{1, \eta^1, \ldots, \eta^k, \ldots}$
The computation **halts** in time **T** (or less) on input x if $\boldsymbol{\eta^T = N}$.
The least such T is the **halting time, $T_M(x)$** (let $T_M(x) = \infty$, if no such T)
The **input-output map** is given by $\Phi_M(x) = O(x^T)$ where $T = T_M(x)$.
A (partial) **function is computable** over R if it is the input-output map of some machine over R.

Let **$F_M(x, y, T)$** be the assertion: **"Machine M with input x halts with output y in time ≤ T."**
This can be expressed:

$$\exists\, z^0, z^1, \ldots, z^T \in (NxS)^{T+1}\ \exists\, w \in S$$

$$[(z^0 = (1, I(x))) \,\&\, (z^T = (N, w)) \,\&\, (O(w) = y) \,\&_{k=1}^{T} (z^k = H(z^{k-1}))]$$

This essentially asserts that a certain "algebraic" system has a solution, as we shall see.

Let $\Omega_T = \{ x \in R^n \mid T_M(x) \le T \}$ be the **time-T halting set** of M and
$\Omega_M = \{ x \in R^n \mid T_M(x) < \infty \}$ be the **halting set** of M.

Path Decomposition Theorem

1. **Ω_T** is a **finite disjoint union of basic semi-algebraic sets** (quasi-algebraic sets). Thus the time-T halting set Ω_T is described by a semi-algebraic (quasi-algebraic) formula whose length may be exponential in T. **(Will want a more succinct description.)**
2. The halting set **Ω_M** is a countable disjoint union of basic semi-algebraic sets (quasi-algebraic sets)
3. The **input-output map** is a piecewise polynomial (rational) map.

The Register Equations and Succinct Descriptions of Time-T Halting Sets

$x \in \Omega_T \Leftrightarrow \exists$ a sequence $z = (z^0, z^1, \ldots, z^T) \in (NxS)^{T+1}$ satisfying the

(Time-T) Register equations, 1st form:

1. $z^0 = (1, I(x))$ and $\pi_M(z^T) = N$ (*initial and terminal conditions*) **Boundary Conditions**
2. $z^k = H(z^{k-1})$, k = 1, …, T **(*next state conditions*)**

(Time-T) Register equations, 2nd form:

1. (η^0, x^0), $= (1, I(x))$ and $\eta^T = N$
2. $\eta^k = \beta(\eta^{k-1}, x^{k-1})$, $x^k = g(\eta^{k-1}, x^{k-1})$, k = 1, …, T

We now extend these equations to R by injecting $N \hookrightarrow R^N$ given by $j \longrightarrow e_j$, the jth coordinate vector, j =1,…,N. (ie $e_j = (0,\ldots 0,1$ (jth coordinate), $0,\ldots,0)$) Let,

$$\beta'(\alpha, y) = \sum_{j=1}^{N} \alpha_j e_{\beta(j,y)}, \quad g'(\alpha, y) = \sum_{j=1}^{N} \alpha_j g(j, y)$$

So for $\alpha = e_j$, $\beta'(\alpha, y) = e_{\beta(j,y)}$ and $g'(\alpha, y) = g(j, y)$.

So we have extended H to H: $R^N \times R^m \longrightarrow R^N \times R^m$.

(Time-T) Register equations, 3rd form (over R):

1. $(\alpha^0, x^0) - (e_1, I(x)) = 0$ and $\alpha^T - e_N = 0$
2. $\alpha^k - \beta'(\alpha^{k-1}, x^{k-1}) = 0$ (soon), $x^k - g'(\alpha^{k-1}, x^{-1}) = 0$, k = 1, …, T

Let $\mathbf{w = (w_1, w_2,\ldots, w_t)} = (\alpha^0, x^0, \ldots, \alpha^T, x^T)$

Let $\mathbf{R'_T(x, w)}$ denote the time-T register equations in 3rd form.

Theorem A. $\mathbf{R'_T(x, w)}$ is equivalent to a semi-algebraic (quasi-algebraic) system $\Phi_T(x, w)$ in n +t variables with t = (N+m)(T+1), with at most 4(N +m)T polynomial equations (of degree at most N D_M + 1 in) plus 2T inequalities.

Proof of Theorem A. The only subtle part is analyzing: $\alpha^k - \beta'(\alpha^{k-1}, x^{k-1}) = 0$

Define linear maps: $\beta'^{-}, \beta'^{+} : R^N \to R^N$ where

$\beta'^{-}(\alpha) = \sum \alpha_j e_{\beta j}^{-}$ and $\beta'^{+}(\alpha) = \sum \alpha_j e_{\beta j}^{+}$, where for j not a branching node, $\beta_j^- = \beta_j^+ = \beta_j$

NB. If α is the j-th coordinate vector e_j, then $\beta'^{-}(\alpha) = \beta_j^-$ and $\beta'^{+}(\alpha) = \beta_j^+$ and both of these are equal to β_j for non-branching nodes.

So, if $\alpha = e_j$:

$\alpha' - \beta'(\alpha, x) = 0 \Leftrightarrow B(\alpha', \alpha, x): \ (x_1 < 0 \to \alpha' - \beta'^{-}(\alpha) = 0) \wedge (x_1 \geq 0 \to \alpha' - \beta'^{+}(\alpha) = 0)$

$\alpha' - \beta'(\alpha, x) = 0 \Leftrightarrow B^*(\alpha', \alpha, x): \ (x_1 \geq 0 \vee \alpha' - \beta'^{-}(\alpha) = 0) \wedge (x_1 < 0 \vee \alpha' - \beta'^{+}(\alpha) = 0)$

If R is unordered, use $B^*_{=}(\alpha', \alpha, x): \ (x_1 = 0 \vee \alpha' - \beta'^{-}(\alpha) = 0) \wedge (x_1 \neq 0 \vee \alpha' - \beta'^{+}(\alpha) = 0)$ ■

Get More Succinct Descriptions.

Suppose M is a machine over (R, =) where R a field, or over (R,<) where R is $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{R}$.

Theorem B. The register equations can be replaced by k quadratic equations, $q_1(x,u) = 0, \ldots, q_k(x,u) = 0$, in n + s variables, $x_1, \ldots, x_n, u_1, \ldots u_s$, where s, $k \leq (n + mT)^c$ and c is a constant depending on N and D_M and not on n, m, or T.

In case R is $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{R}$, can replace the quadratic system, $q_1(x,u) = 0, \ldots, q_k(x,u) = 0$, by a single degree 4 polynomial equation, $q(x, u) = 0$ where $q(x,u) = \sum_{i=1}^{k} q_i(x,u)^2$.

Proof of Theorem B.

- Suppose M is a machine over (R, =) where R is a field. Use: $x \neq 0 \Leftrightarrow \exists u(xu = 1)$
- Suppose M is a machine over an ordered field where positive elements have square roots. For example, (ℝ, <) or any real closed field. Then use: $x > 0 \Leftrightarrow \exists u(xu^2 = 1)$
- Now for $(\mathbb{Z}, <)$, use: $x > 0 \Leftrightarrow \exists u \geq 0 \ (x = 1 + u)$ and by Lagrange: $u \geq 0 \Leftrightarrow \exists u_1 u_2 u_3 u_4 \ (u = u_1^2 + u_2^2 + u_3^2 + u_4^2)$
- Modify for $\mathbb{Q}$

The following Lemma finishes the proof of Theorem B.

Lemma. Suppose R is a ring or field. Then any system of polynomial equations $p_1(x) = 0, \ldots, p_t(x) = 0$ in n variables of degree at most D and with at most K monomials per equation (so $K = O(n^D)$) is n-equivalent to a quadratic polynomial system $q_1(x,u) = 0, \ldots, q_{t+t'}(x,u) = 0$ in n+n' variables with $n', t' \leq KD$. Here, $x = (x_1, \ldots, x_n)$ and $u = (u_1, \ldots, u_{n'})$. (NB. In our case, D is dependent only on the machine M.) ■ ■

All arguments are uniform in everything in sight.