Transfer Results for Complexity: An Algebraic setting for the problem "P=NP?"

Theorem (**BCSS, 1996** uses theory of heights; another proof by **Koiran**, 1997 uses bounds on sizes of coefficients of polynomials gotten from good elimination of quantifier algorithms for algebraically closed fields.)

Let $\widetilde{\mathrm{Q}}$ be the algebraic closure of $\mathbb{Q}$. Then, P = NP over $\mathbb{C}$ ⇔ P = NP over $\widetilde{\mathrm{Q}}$.

Can replace $\mathbb{C}$ by any algebraically closed field F of characteristic 0.
Thus, the question "P = NP?" has the same answer for all algebraically closed fields of characteristic 0.

UP ⇐) (also C. Michaux,1994)

Suppose K ⊂ L are algebraically closed fields. We show: P = NP over K ⇒ P = NP over L. This will follow from **the NP-completeness of HN** for fields and **Model Completeness** for algebraically closed fields:

Model Completeness (**Strong Transfer Principle**) for the Theory of algebraically closed fields: Suppose K ⊂ L are algebraically closed fields and Φ is a 1^{st} order sentence in the language of fields with constants from K. Then Φ is true in K ⇔ Φ is true in L.

So now, suppose P = NP over K. This implies there is a polynomial p and a machine M over K that decides HN over K in time p(input size).

Let f = $\{f_i(\mathrm{y}^i, x) = 0\}_{i=1}^m$ be the general system of m polynomials of degree d in n variables $\mathrm{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_n)$ and variable coefficients $\mathrm{y} = (\mathrm{y}^1, \ldots, \mathrm{y}^m)$ where $\mathrm{y}^i = (\mathrm{y}^i_1, \ldots \mathrm{y}^i_k)$.
Let k’ be the input size of the system.

Recall: $F_M(x,y,T)$ is the assertion that “machine M with input x halts with output y in time T.” This is equivalent to the assertion that “the polynomial system gotten from the time-T Register Equations (with condition x^0=I(x) and O(x^T) = y) has a solution over R.” By abuse of notation, we will also denote the latter by $F_M(x,y,T)$ when expressed as a 1^{st} order formula.

Let Φ_f be the sentence: $\forall \mathrm{y} = (\mathrm{y}^1, \ldots, \mathrm{y}^m)$

$$\{[\exists \mathrm{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_n) \mathop{\&}_{i=1}^{m} (f_i(\mathrm{y}^i, x) = 0) \Leftrightarrow F_M(y, 1, p(k'))] \,\&\, [\neg \exists x \mathop{\&}_{i=1}^{m} (f_i(\mathrm{y}^i, x) = 0) \Leftrightarrow F_M(y, 0, p(k'))]\}$$

Φ_f asserts that for each specialization of the coefficients y = (y^1, …y^m), the resulting system has a solution if and only if the Machine M with input y halts with output 1 in time p(k) and the resulting system has no solution if and only if M with input y halts with output 0 in time p(k).

Φ_f is true in K, so by model completeness Φ_f is true in L. So HN ∈ P over L. So P = NP over L.

Eliminating constants

We now make the assumption that at any computation node of a machine M over L, the computation performed is either $+$, $-$, or $\times$ two elements of L. (Any machine can be so converted with at most a multiplicative constant increase in halting time. Exercise.)

(Down) ⇛) Suppose L is an algebraically closed field of characteristic 0.
Will show: $HN \in P$ over L ⇛ $HN \in P$ over $\widetilde{Q}$.

It follows from model completeness, that given a system $f_1, \ldots, f_k \in \widetilde{Q}[x_1, \ldots, x_n]$,
if it is solvable over L, then it is already solvable over $\widetilde{Q}$.
So, if machine M decides HN_L over L, it also decides $HN_{\widetilde{Q}}$.

Trouble is, M may have built in constants from L. Need to simulate M by a machine M' without these constants that behaves correctly on inputs from $\widetilde{Q}^\infty$ with at most a polynomial slow down.

So suppose: Problem (Y, Y_{yes}) over L is decided by a machine M over L with constants from L.
Wlog we can assume the constants are $\{u_1, \ldots, u_s, v_1,\ldots,v_k\}$ with $u_1, \ldots, u_s$ algebraically independent over $\widetilde{Q}$ and $v_1,\ldots v_k$ algebraic over $\widetilde{Q}$ $(u_1, \ldots, u_s)$.
We have $\widetilde{Q} \subset \widetilde{Q}(u_1, \ldots, u_s) = F \subset F(v_1,\ldots, v_k) = H$ and $(Y, Y_{yes})|_F$ and $\left(Y,\ Y_{yes}\right)|_{\widetilde{Q}}$ are decided by a machine over H with constants $v_1,\ldots,v_k$. Here $(Y, Y_{yes})|_F$ denotes $(Y \cap F^\infty, Y_{Lyes} \cap F^\infty)$.
(Thus, if $HN_L = (Y, Y_{yes})$, then $\left(Y,\ Y_{yes}\right)|_{\widetilde{Q}} = HN_{\widetilde{Q}}$, by above.)

Proposition 1. Suppose $F \subset F(v_1,\ldots, v_k) = H$, v_j, algebraic over F. Let M be a machine with constants $v_1,\ldots,v_k$ that decides $(Y, Y_{yes})|_F$. Then there is a $c \in \mathbb{R}^+$ and a machine M' over F that solves $(Y, Y_{yes})|_F$ in time $T_{M'}(y) \leq cT_M(y)$ for all $y \in Y \cap F^\infty$. (Do not need ch 0 here.)

Proof. Consider H as a vector space over F of dimension q. So H may be represented as F^q where the inclusion $F \subset H$ is represented as F in F^q as the first coordinate.

Now construct M' over F so that on inputs from F^∞, M' simulates M. The state space of M' is $(F^q)_\infty$ so it also represents H_∞. An initial subroutine takes $x = (x_1, \ldots, x_n) \in F^\infty$ to initial coordinates in $(F^q)_\infty$, ie to (.., ..., . $(x_1, 0, \ldots,0), (x_2, 0, \ldots,0), \ldots ,(x_n, 0, \ldots,0), (0, \ldots 0),\ldots$).

Addition and multiplication in H are represented by fixed symmetric bilinear maps:
B_+: $F^q \times F^q \longrightarrow F^q$ and $B_\times$: $F^q \times F^q \longrightarrow F^q$.
M' can incorporate these polynomial maps in computation nodes. Subtraction is simulated by multiplication by (-1) followed by B_+.

Since $(x_1, \ldots, x_q)$ represents the 0 element in F^q (ie H) if and only if all the x_i are 0, branching in M' is simulated by checking if the first q coordinates in the state space are 0. Shifting right or left is simulated by shifting right or left q times. Care is taken to keep track of the intended lengths of sequences in the computation. A final subroutine assures that the appropriate finite sequence is outputted.

Proposition 2. Suppose $\widetilde{Q} \subset \widetilde{Q}(u_1, \ldots, u_s)$ with the $\{u_1, \ldots, u_s\}$ algebraically independent over $\widetilde{Q}$. Let M be a machine with constants in $\widetilde{Q}(u_1, \ldots, u_s)$ that decides $\left(Y,\ Y_{yes}\right)|_{\widetilde{Q}}$. Then there is a machine M' over $\widetilde{Q}$ and $c \in \mathbb{N}^+$ that solves $\left(Y,\ Y_{yes}\right)|_{\widetilde{Q}}$ in time $T_{M'}(y) \le T_M(y)^c$ for all $y \in Y \cap \widetilde{Q}^\infty$.

Proof.

The constants of M are polynomials in $\widetilde{Q}[u_1, \ldots, u_s]$. Let $a = (a_1, \ldots, a_k)$ be a sequence of all coefficients in $\widetilde{Q}$ occurring in these polynomials. So each of these polynomials is of the form $p(a, u)$ with coefficients in $\mathbb{Z}$.

Our aim is to construct a machine M' over $\widetilde{Q}$ that given $y = (y_1,\ldots,y_n) \in Y \cap \widetilde{Q}^\infty$ generates a computation path that simulates the computation path γ_y traversed by y when input to M. Here, $\gamma_y = (\eta_0, \eta_1, \ldots, \eta_t, \ldots)$.

The critical construction is to simulate the "branching" structure of γ_y.

Let η_t be a branch node in γ_y and let f_t be the associated step t branching polynomial.
We consider f_t as a polynomial in y, a and u over $\mathbb{Z}$,
Recall: f_t is given as a straight line program.

The computation path branches right or left according to whether or not $f_t(y, a, u) = 0$.
So the machine has to decide if $f_t(y, a, u) = 0$, or not.

The idea is to eliminate $u = (u_1, \ldots, u_s)$ by quickly constructing a witness $w = (w_1, \ldots, w_s) \in \widetilde{Q}^s$ with the property: $f_t(y, a, w) = 0 \Leftrightarrow f_t(y, a, u)=0$.

Theorem (Cauchy, 1829).
Let $f(x) = a_d x^d + a_{d-1}x^{d-1} + \ldots + a_1$, $a_d \ne 0$. Let $M = \max_{j \ne d} |a_j/a_d|$.
Then if $|x| > M + 1$, $f(x) \ne 0$.

Given $G \in \mathbb{Z}[t_1,\ldots, t_n]$. **Define τ (G):**
Consider the finite sequences: $(u_0, u_1,\ldots,u_n, u_{n+1}, \ldots, u_{n+s} = G)$ where $u_0 = 1, u_1 = t_1, \ldots, u_n = t_n$, and for $n < k \le n+s$, $u_k = v*w$ for some $v, w \in \{u_0, u_1,\ldots, u_{k-1}\}$ and $*$ is $+, -$ or $\times$. Then **τ (G)** is the minimum such s +1.

Witness Theorem (BCSS, 1986).
Let $F(x, t) = F(x_1, \ldots, x_r, t_1, \ldots, t_s)$ be a polynomial in r + s variables with coefficients in $\mathbb{Z}$ and let $F_x \in \widetilde{Q}[t_1, \ldots, t_s]$ be defined as $F_x(t) = F(x,t)$ for each $x \in \widetilde{Q}^r$.
Suppose N is a positive integer satisfying: $\log N \ge 4(r+s)\tau^2 + 4\tau$, $\tau = \tau(F)$.

Then for $x \in \widetilde{Q}^r$, there exists an algebraic number $w_1 \in \{2^N, x_1^N,\ldots,x_r^N\}$ such that $w = (w_1, \ldots, w_s)$ where $w_{i+1} = w_i^N$ is a witness for $F_x \in \widetilde{Q}[t_1, \ldots, t_l]$.

Back to proof of Proposition 2.

Again, let $f_t(y_1,\ldots,y_n, a_1, ..., a_k, u_1, \ldots, u_s)$ be the step t branching polynomial for computation path γ_y of M evaluated for input $y = (y_1,\ldots,y_n)$. Let $r = n + k$.

M' decides whether or not the value is 0 without using the algebraically independent elements $u_1, \ldots, u_s$ as follows by generating r + 1 potential "witnesses," each of form:
$w = (w_1, \ldots, w_s) \in \widetilde{Q}^s$ with the property that: $f_t(y, a, u) = 0$ if and only if all $f_t(y, a, w) = 0$.
M' then computes all the $f_t(y, a, w)$ and branches left if one is $\neq 0$ and right if all $= 0$.

Note: $\tau(f_t) \leq t + C$ where C is the sum of all the τ (p) over all constants in M.

To get these r + 1 potential witnesses, apply the Witness Theorem by letting
$W = 4(r+s)(t+C)^2 + 4(t+C)$ and repeat squaring W times each of
$2, y_1,\ldots,y_n, a_1, ..., a_k$ to get r + 1 potential w_1's: $2^N, y_1^N,\ldots,y_n^N, a_1^N, ..., a_k^N$ where $N = 2^W$.

For each of these r +1 w_1's , let $w = (w_1, \ldots, w_s)$ where $w_{i+1} = w_i^N$ is as in the Witness Theorem.
■