BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 21, Number 1, July 1989

ON A THEORY OF COMPUTATION AND COMPLEXITY OVER THE REAL NUMBERS: *NP*-COMPLETENESS, RECURSIVE FUNCTIONS AND UNIVERSAL MACHINES[1]

LENORE BLUM[2], MIKE SHUB AND STEVE SMALE

ABSTRACT. We present a model for computation over the reals or an arbitrary (ordered) ring R. In this general setting, we obtain universal machines, partial recursive functions, as well as NP-complete problems. While our theory reflects the classical over $\mathbf{Z}$ (e.g., the computable functions are the recursive functions) it also reflects the special mathematical character of the underlying ring R (e.g., complements of Julia sets provide natural examples of R. E. undecidable sets over the reals) and provides a natural setting for studying foundational issues concerning algorithms in numerical analysis.

Introduction. We develop here some ideas for machines and computation over the real numbers $\mathbf{R}$.

One motivation for this comes from scientific computation. In this use of the computer, a reasonable idealization has the cost of multiplication independent of the size of the number. This contrasts with the usual theoretical computer science picture which takes into account the number of bits of the numbers.

Another motivation is to bring the theory of computation into the domain of analysis, geometry and topology. The mathematics of these subjects can then be put to use in the systematic analysis of algorithms.

On the other hand, there is an extensively developed subject of the theory of discrete computation, which we don't wish to lose in our theory. Toward this end we define machines, partial recursive functions, and other objects of study over a ring R. Then in the case where R is the ring of integers $\mathbf{Z}$, we have the same objects (or perhaps equivalent objects) as the classical ones. Computable functions over $\mathbf{Z}$ are thus ordinary computable functions. R.E. sets over $\mathbf{Z}$ are ordinary R.E. sets. But when the ring is specialized to the real numbers, we have computable functions which are reasonable for the study of algorithms of numerical analysis. R.E. sets over $\mathbf{R}$ are no longer countable and include, for example, basins of attraction of complex analytic dynamical systems.

Received by the editors April 21, 1988.

1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 03D15, 68Q15; Secondary 65V05.

[1]Partially supported by NSF grants. Some of this work was done while Blum and Smale were visiting Shub at the IBM, T. J. Watson Research Center.

[2]Partially supported by the Letts-Villard Chair at Mills College and the International Computer Science Institute, Berkeley.

There is another virtue of developing a theory of machines over a ring. It forces a more algebraic approach, closer to classical mathematics, than the approach from logic.

Here is an abbreviated description of some of the results of this paper, in this context of machines over a ring R.

> (I) Most Julia sets are not R.E. over the reals, so their complements, the basins of attraction, provide natural examples of R.E. *undecidable sets* over $\mathbf{R}$ (§§1 and 10).

The Julia set example provides an interesting link between the theory of computation and dynamical systems. A perhaps deeper link is the *computing endomorphism* (§3) which is an important conceptual and technical tool used in our development.

> (II) An analogue of Cook's NP-completeness theorem is proved over the real numbers. The *NP-complete problem* over $\mathbf{R}$ is the 4-*Feasibility problem*, i.e. the problem of deciding whether or not a real degree 4 polynomial $f\colon \mathbf{R}^n \to \mathbf{R}$ has a zero (§6).

This result, in addition to focusing attention on the 4-Feasibility problem over $\mathbf{R}$ has some interesting consequences which point to the *subtle differences* between the theories of NP over $\mathbf{R}$ and over $\mathbf{Z}$. For example, by straightforward counting arguments, any NP-problem over $\mathbf{Z}$ is seen to be solvable in $2^{\text{poly}(n)}$ time. (See e.g. *Garey-Johnson.*) An analogous result over the reals is far from obvious since there are a continuum of possible guesses over $\mathbf{R}$. It is not even clear *a priori* that NP-problems over $\mathbf{R}$ are decidable. However, since the 4-Feasibility problem over $\mathbf{R}$ is decidable (by *Tarski-Seidenberg*) and since the current best upper bound for decidability of the existential theory of the reals is $4^{O(n)}$ (see *Renegar*, also *Canny* and *Grigorev-Vorobjov*), we also get exponential upper bounds for NP-problems over $\mathbf{R}$ but for much deeper reasons than the case over $\mathbf{Z}$. For another interesting difference between the two theories, note that by Hilbert's Tenth Problem, the 4-Feasibility problem restated over $\mathbf{Z}$ is *not* even decidable over $\mathbf{Z}$ and so *not* in NP over $\mathbf{Z}$.

PROBLEM. What is the relation between the problems $P \stackrel{?}{=} NP$ over $\mathbf{R}$, and $P \stackrel{?}{=} NP$ over $\mathbf{Z}$?

> (III) Computable functions over R are characterized intrinsically by a class we call *partial recursive functions* over R. For $R = \mathbf{Z}$, these are the usual partial recursive functions (§7).

> (IV) There exists a *universal machine* over R. This machine, inspired by the Universal Turing Machine, does the computation of any machine over R. The universal machine over R turns out to be independent of R. Moreover, by avoiding Gödel coding via prime numbers, the algebraic structure of the universal machine remains intact (§8).

(V) Inspired by the work of Davis, Robinson and Putnam, and Matijasèvic on Hilbert's tenth problem, we give a "diophantine-like" description of R.E. sets, for a certain class of machines (§9).

There are a large number of contributions of mathematicians and computer scientists which predate and relate to this work. A very brief survey, with some comparisons, follows.

Of course, the work of Turing, Gödel, Church and others in the thirties forms the core of the existing framework for our work. Although much of the classical theory of computation deals with computing over the natural numbers, certain approaches have considered other underlying domains.

Close to the classical approach, *Rabin* developed a theory of computable algebra and fields in which the underlying domains can be effectively coded by natural numbers and are thus, necessarily countable.

On the other hand, the theories of computation over abstract structures, are perhaps more general than ours. See e.g., *Friedman* (or as discussed by Shepherdson in *Harrington, et al.*), *Tiuryn*, and *Moschovakis*. These general approaches both exploit and explore the logical properties of procedures. But, when applied to specific structures such as the reals, they do not yield the concrete mathematical results (e.g. about Julia sets or the 4-Feasibility problem) that quite naturally follow from the more mathematical model developed in this paper.

Recursive analysis provides yet another approach. See, *e.g. Friedman-Ko, Pour-El-Richards, Hoover and Kreitz-Weihrauch*. Some tools here are recursive functionals, computable real numbers and oracle Turing machines where, roughly, one imagines a real number fed to the machine bit by bit. To contrast, we view a real number not as its decimal (or binary) expansion, but rather a mathematical entity as is generally the practice in numerical analysis. Thus, for example, we suppose Newton's algorithm for finding the zeroes of a polynomial f to be performed on an arbitrary real, not just a computable real; the fundamental components of the algorithm in our model, as in practice, are the rational operations $N_f(x) = x - f(x)/f'(x)$, not the bit operations.

The development of algebraic complexity theory, in particular the work of Ostrowski, *Pan, Winograd, Strassen and Schönhage* (see *von zur Gathen* for a recent survey) gave rise to the "real number model" approach to computation. Decision and computation tree models as in *Rabin, Steele-Yao, Ben-Or*, and the tame machines in *Smale*, are such real number models of computation but considerably less powerful or general purpose than ours (e.g., they have bounded halting time and none are universal; also they don't allow for uniform algorithms as do our infinite dimensional machines).

More closely related are the register machines of *Shepherdson-Sturgis* and the RAM's or random access machines. (See *Aho-Hopcroft-Ullman* or *Machtey-Young* for discrete versions.) While a definition of a real RAM is given in *Preparata-Shamos*, the formal development of a theory is not

pursued. Indeed, in their book, only a subclass of machines, equivalent to the class of decision trees, is utilized. Perhaps closest to our approach is the work of *Herman-Isard* on computability over arbitrary fields. Also close in spirit is a theory of real Turing machines outlined by *Abramson.* Nimrod Megiddo has also considered an example of an NP-complete over $\mathbf{R}$ in our model.

Some other related papers are *Borodin, Valiant, Endler, Lovász*, and *Eaves-Rothblum* and *Traub-Wozniakowski.* Books having significant contact with this paper include *Davis, Eilenberg, Manin, Manna, Minsky* and *Rogers.*

Especially in §§5 and 6 below, the influence of complexity work of Cook and Karp (see *Garey-Johnson*) among many others, is evident. In our §8, *Robinson, Matijasèvic, Davis, and Putnam*, and *Denef* have been influential.

We would like to acknowledge helpful discussions with Martin Davis and Steve Simpson.

Sections

1. Examples of machines over R. Even before defining our notion of a machine, we give some examples. The first examples are related to the theory of complex dynamical systems. We present them in some detail.

EXAMPLE 1. Consider a complex polynomial map $g: \mathbf{C} \to \mathbf{C}$. This map g will be considered as an endomorphism, mapping $\mathbf{C}$ into itself. Thus it makes sense to iterate it. That is $g(g(z)) = g^2(z)$ is defined as well as the kth iterate $g^k(z)$, for each $z \in \mathbf{C}$.

LEMMA. *There is a real constant, $C = C_g$ such that if $|z| > C$, then $|g^k(z)| \to \infty$ as $k \to \infty$.*

This is true because the highest order term of a polynomial dominates the others for $|z|$ sufficiently large. Moreover, if $g_0(z) = z^d$, $|g_0^k(z)| = |z|^{d^k}$.

Now we may define a "machine" M from g, by the following flow chart (see Figure 1).

This M is a machine over $\mathbf{R}$, not $\mathbf{C}$, since it uses the real comparison $|z| < C_g$; in the context of this machine, we view $\mathbf{C}$ as $\mathbf{R}^2$.

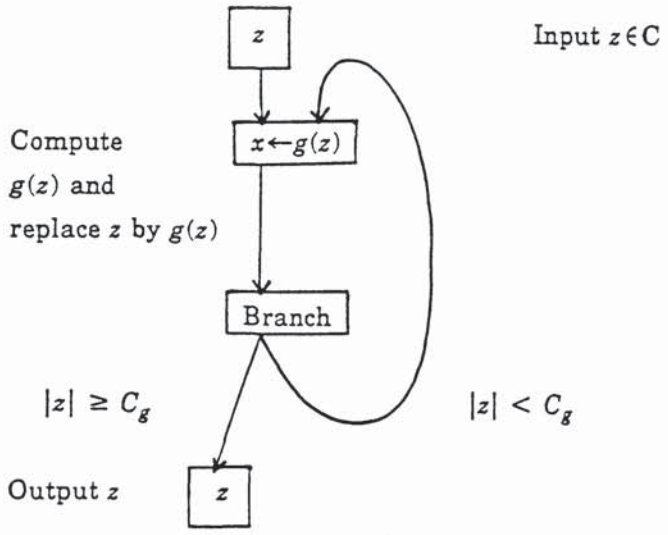

FIGURE 1

One can see that the "halting set" Ω_M of the machine M is precisely the set of points which eventually tend to ∞ under iterates of g. The halting set is analogous to the R.E. sets of recursive function theory, and eventually we will define a class of machines which contains not only this g machine, but machines equivalent to Turing machines as well. Thus we call Ω_M an R.E. set over $\mathbf{R}$. Note that it is certainly not a usual R.E. set since it is not countable. It is natural to ask, is Ω_M "decidable" or, inspired by the classical tradition, is the complement of Ω_M the halting set of some other machine over $\mathbf{R}$?[3]

Of course at this point, not even having a definition of a machine over $\mathbf{R}$, the question can't be answered. But later we will show

PROPOSITION 1. *Any R.E. set over* $\mathbf{R}$ *is the countable union of basic semialgebraic sets.*

Here a basic *semialgebraic* set is a subset of Cartesian space R^n defined by a set of polynomial inequalities of the form

$$h_i(x) < 0, \qquad i = 1, \ldots, l,$$
$$h_j(x) \leq 0, \qquad j = l+1, \ldots, m.$$

A general reference for semialgebraic sets is *Becker*.

Using this proposition, we answer our question in the next example for a class of halting sets Ω_M.

[3]If the complement of Ω_M is the halting set of some other machine M', we can construct a machine to decide for each $z \in \mathbf{C}$ "Is $z \in \Omega_M$?" schematically as follows (see Figure 2).

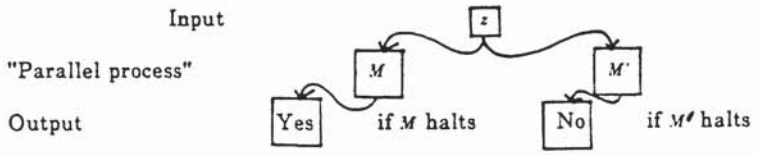

FIGURE 2

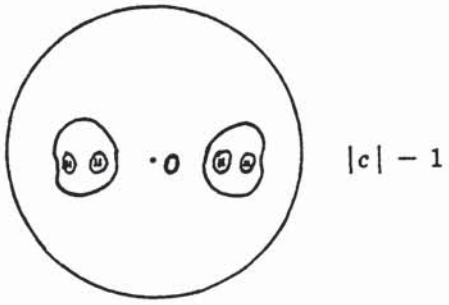

FIGURE 3

EXAMPLE 2. Specialize g to be a polynomial $g(z) = z^2 + c$, where $|c| > 4$. In that case the complement of Ω_M is a Cantor set, a certain Julia set of complex dynamical system theory. Let $J = \mathbf{C} - \Omega_M$.

To see that J is a Cantor set, first note that the exterior of the circle of radius $|c| - 1$ is mapped into itself, in fact any point in it moves away from zero and tends to ∞ under iteration. As 0 is the only critical point of $z^2 + c$ and it maps to c which is in the exterior of the circle of radius $|c| - 1$ we see that the inverse image of the interior of the circle is two discs interior to the circle (see Figure 3).

Now the inverse image of these discs is four discs, 2 each in the interior of the two, etc., The intersection of the inverse images is precisely the set of points which don't tend to ∞. At the very first stage we have the two inverse mappings of the disc into its interior, by the Schwarz lemma each of these is a strict contraction. Therefore any infinite nesting of inverse images contains exactly one point and the intersection of the inverse images is a Cantor set.

Now since a basic semialgebraic set has a finite number of connected components, it follows from the previous proposition that J is not an R.E. set and hence

PROPOSITION 2. *Ω_M for the case $g(z) = z^2 + c$, $|c| > 4$, is an R.E. set over* $\mathbf{R}$ *which is not decidable over* $\mathbf{R}$.

EXAMPLE 3. We now suppose, more generally, that $g = p/q: \hat{\mathbf{C}} \to \hat{\mathbf{C}}$ is a rational endomorphism of degree at least 2 of the Riemann sphere $\hat{\mathbf{C}} = \mathbf{C} \cup \{\infty\}$.

Thus, $(\hat{\mathbf{C}}, g)$ is a discrete complex analytic dynamical system. (See e.g. *Blanchard.*) Of primary interest is the long term behavior of points in $\hat{\mathbf{C}}$ under the "action" of g. And so a key object of study is the *orbit* of a point z_0 under g: $z_0, z_1 = g(z_0), \ldots, z_k = g^k(z_0), \ldots$.

The simplest case is that of a *fixed point* of g, i.e. a point $z_0 \in \hat{\mathbf{C}}$ such that $g(z_0) = z_0$. We say a fixed point is *attracting* if the modulus of the derivative of g at z_0 is less than 1, i.e. $|g'(z_0)| < 1$. This implies there is a neighborhood U of z_0 that is contracted into itself under g, i.e., $g(U) \subset U$; and so if the orbit of a point eventually enters U, it will asymptotically approach z_0. A fixed point is *repelling* if $|g'(z_0)| > 1$, so nearby points are

pushed away by g. (In the *nonhyperbolic* case, i.e. when $|g'(z_0)| = 1$, the behavior of nearby points is not as clear cut.)

Now more generally, a point $z_0 \in \mathbf{C}$ is periodic (of *period* n) if $g^n(z_0) = z_0$ for some $n \in \mathbf{Z}^+$. It is *attracting* (respectively *repelling*) if in addition, $|(g^n)'(z_0)| < 1$ (respectively $|(g^n)'(z_0)| > 1$), where $(g^n)'(z_0)$ is the derivative of the nth iterate of g at z_0. By the Chain Rule, these properties remain the same for all points in the orbit of z_0: $z_0, z_1 = g(z_0), \ldots, z_n = g^n(z_0)$. (The corresponding periodic properties of the point at ∞ are usually determined by the properties of 0 after the change of coordinates $z \to 1/z$.)

If z_0 is attracting of period n, the derivative condition implies there is a neighborhood U of z_0 such that $g^n(U) \subset U$. So orbits of points that eventually enter U under the action of g will asymptotically approach the orbit of z_0. Such points are said to be in the basin (*of attraction*) of z_0; the *basin* (*of attraction*) of g is the union of all such basins.

PROPOSITION 3. *The basin of attraction of g is an R.E. set over* **R**.

To show this we construct a machine M whose halting set is the basin of g. Since there are only a finite number of attracting periodic points for rational maps (see *Blanchard*) there is a real polynomial h (of 2 real variables) such that $h(z) < 0$ if and only if z belongs to a finite union of discs around the attracting periodic points which is contracted into itself by g. Thus, a point is in the basin of g if and only if for some z in its orbit, $h(z) < 0$.

Now let the machine M be described by (Figure 4).

Clearly, Ω_M, the halting set of M, is the basin of attraction of g.

Again, it is natural to ask if the basin of attraction of g is decidable, or equivalently, if its complement is R.E.

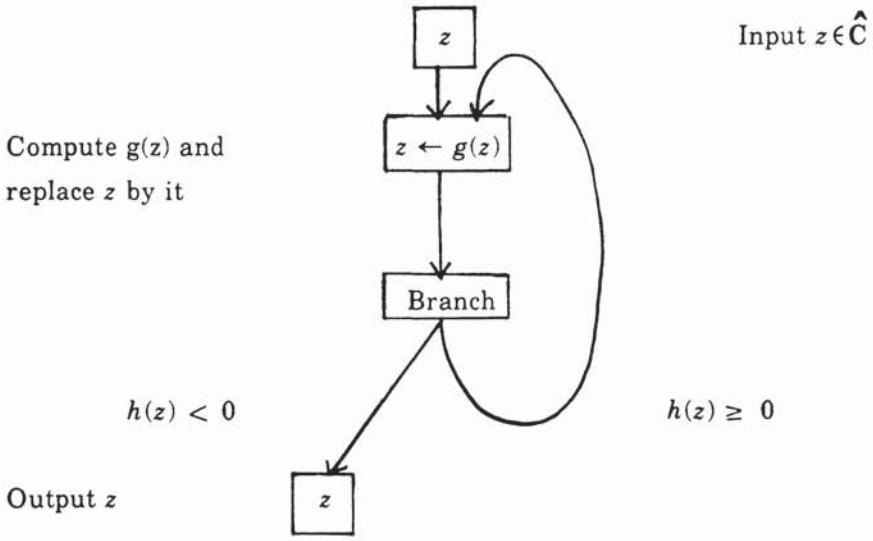

FIGURE 4

Quite generally, the complement of the basin of attraction is the Julia set of g. This is true at least for the set of *hyperbolic rational maps.* (These maps are open and nonempty in the set of rational maps of each degree, and conjectured to be open and dense. See *Sullivan.*)

The Julia set of g, J_g, is the closure of the set of repelling periodic points of g. In Example 2 above, the Julia set of g is the complement of Ω_M and, as shown, not decidable. To contrast, it is an easy, but instructive exercise to show that the Julia set of the map $g(z) = z^2$ is the unit circle, and hence decidable over **R**.

In §10 below, we shall investigate systematically conditions under which a Julia set can be R.E. over **R**. Indeed we show that "most" Julia sets are not R.E., and hence they and their complements are not decidable.

EXAMPLE 4. A more elementary example (due to Feng-Gao) of an R.E. set over **R** which is not decidable is the complement of the *Cantor Middle third set* in the unit interval. The demonstration is via the following "machine" (and the fact the Cantor Middle third set is an uncountable totally disconnected set) (see Figure 5).

EXAMPLE 5. Another type of example is the machine that computes *the greatest integer in* x, $\lfloor x \rfloor$, for $x \geq 0$ in **R**. (See Figure 6.)

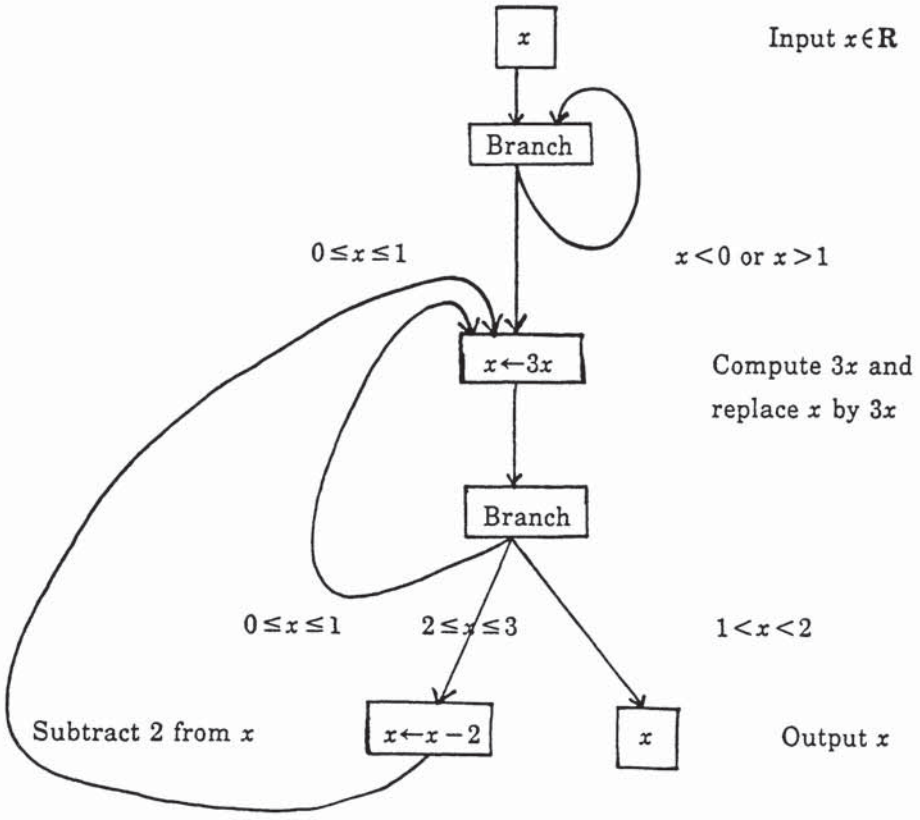

FIGURE 5

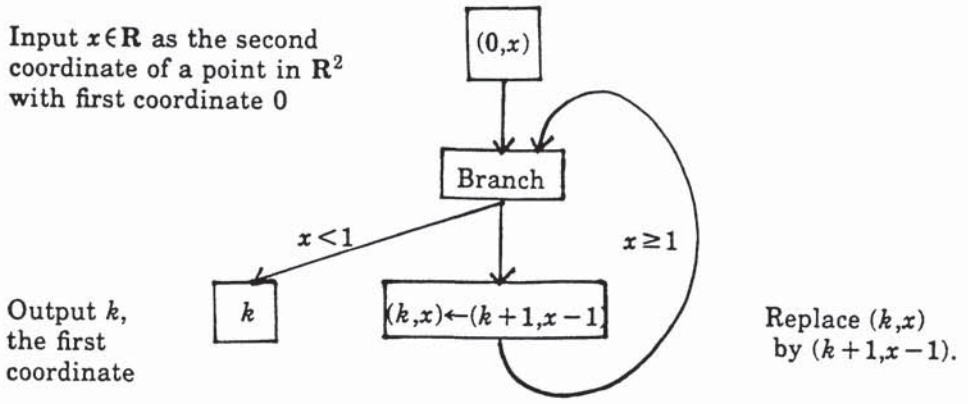

FIGURE 6

Note that for $x > 0$ in $\mathbf{R}$, the "cost" of computing $\lfloor x \rfloor$ by the above machine is $\lfloor x+1 \rfloor$ comparisons and $\lfloor x \rfloor$ (pairs of) basic arithmetic computations. Using binary search these costs could be reduced to $O(\log(x+1))$ but essentially no more in our model (See Proposition 3 in §4). It is of interest to note that in models of computation where $\lfloor x \rfloor$ as well as the basic arithmetic operations can be computed in constant time, seemingly hard problems such as factoring integers (*Shamir*) and testing satisfiability of propositional formulas (this follows from *Schönhage*) can be solved *efficiently*, i.e., in polynomial time.

EXAMPLE 6. Now let $S \subset \mathbf{Z}^+$, the positive integers. We construct a machine M_S over $\mathbf{R}$ that "decides" S. That is, for each input $n \in \mathbf{Z}^+$, M_S outputs 1 (*yes*) if $n \in S$ and 0 (*no*) if $n \notin S$.

M_S has a built in constant $s \in \mathbf{R}$ defined by its binary expansion

$$s = .s_1 s_2 \ldots s_n \ldots, \text{ where } s_n = \begin{cases} 1 & \text{if } n \in S, \\ 0 & \text{otherwise.} \end{cases}$$

M_S with its built in constant s, plays a role analogous to an "oracle" for a Turing machine that answers queries "Is $n \in S$?" at a cost of $n \log n$. (Using methods related to those used in Propositions 3 and 4 and the Remark at the end of §4, one can give an order n lower bound on

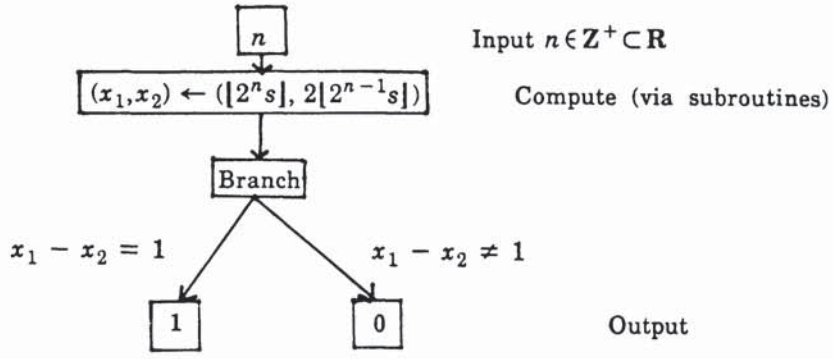

FIGURE 7

the cost of accessing the nth bit of binary representations of real numbers $s \in (0, 1)$ by machines over $\mathbf{R}$.) (See Figure 7.)

EXAMPLE 7. As a final example we describe the Travelling Salesman Problem (TSP) over an ordered ring R. Here we are given a nonnegative symmetric $n \times n$ matrix A over R with entries A_{ij} denoting the *distance* between "cities" i and j. Thus A is a *mileage chart* for the n cities $\{1, \ldots, n\}$. The "problem" is: Given *instance* A, *find* a tour of minimum distance. A *tour* is a cycle t of $\{1, \ldots, n\}$ and the *distance* function to be minimized is $D(A, t) = \sum_{i=1}^{n} A_{it(i)}$.

The Travelling Salesman "decision problem" over R is: Given (k, A), where $k \in R^+$ and A is a mileage chart, *decide* if there is a tour of distance $\leq k$. Over $\mathbf{Z}$, this is the famous NP-complete problem of classical complexity theory.

Of interest to us is the TSP over the reals, $\mathbf{R}$. In §4 we show that the "topological complexity" of the TSP over $\mathbf{R}$ is at least $(n - 1)!/2$. The topological complexity measures the branching necessary and sufficient to solve all instances of the n-city TSP over R by machines over R. In §5 we develop a notion of NP over an ordered ring R and show that Travelling Salesman decision problem is NP over the reals.

2. Machines over a ring R. Let R be a ring, commutative with unit, and we suppose that R is ordered. The main examples are $R = \mathbf{Z}$, the integers, and $R = \mathbf{R}$, the real numbers. In every case $\mathbf{Z}$ is a natural subring of R.

Let R^n denote the direct sum of R with itself n times. We often want to allow that $n = \infty$, in which case R^n is called the countable direct sum of R with itself. A point $x = (x_1, \ldots, x_n, \ldots)$ in R^∞ satisfies $x_k = 0$ for k sufficiently large. If n, m are both finite a map $f\colon R^n \to R^m$ is a *polynomial map* if the coordinate maps f_i are polynomials in the n variables for $i = 1, \ldots, m$. If R is a field, then f is *rational* if the f_i are rational functions in the n variables. The degree of f is the maximum of the degree of the f_i.

In case n is ∞ we impose a further condition on f in order that it be called polynomial or rational. This condition is that there is a k such that $f_i(x) = x_i$ if $i > k$ and $\partial f_i(x)/\partial x_j \equiv 0$ if $i \leq k$ and $j > k$. This means that at most k variables and coordinates are *active* in that computation. The least such k will be called the *dimension of* f. The *degree* of f is as above.

In case R is a field we will write $f\colon R^n \to R^m$, f rational, even though f may not be defined everywhere.

A machine M over R consists of an *input space* $\overline{I}$, *output space* $\overline{O}$ and *state space* $\overline{S}$, together with a connected directed graph whose nodes labelled $1, \ldots, N$ are of certain types and with associated functions. We proceed more precisely and at first in the finite dimensional case. Here $\overline{I}$, $\overline{O}$, and $\overline{S}$ are each R^l, R^m and R^n respectively, with $l, m, n < \infty$.

The directed graph of the machine M has 4 types of nodes as follows:

(1) Exactly one *input node*, node 1, characterized as having no incoming edge, and one outgoing edge. Associated to this input node is a linear injective map $I: \overline{I} \to \overline{S}$ (which just takes the input and puts it into the machine), and $\beta(1)$ the *next node.*

(2) *Output nodes* characterized by having no outgoing edges. To each such node, n, is associated a linear map $O_n: \overline{S} \to \overline{O}$.

(3) *Computation nodes*; each such node has a single outgoing edge, so that a *next node* $\beta(n)$ is defined. To n is associated a polynomial map $g_n: \overline{S} \to \overline{S}$ (an "endomorphism"). If R is a field then g_n could be taken rational.

(4) A *branch node* n has two outgoing edges, giving us *next nodes* $\beta^-(n)$ and $\beta^+(n)$. To n is associated a polynomial $h_n: \overline{S} \to R$ with $\beta^-(n)$ associated to the condition $h_n(x) < 0$, $\beta^+(n)$ to $h_n(x) \geq 0$.

If M is a finite dimensional machine over R, we may define the *input-output* map $\varphi_M: \Omega_M \to \overline{O}$, $\Omega_M \subset \overline{I}$, as follows. If for some computation node n, $g_n(x)$ is not defined because a polynomial denominator h vanishes at x we may modify M by changing (see Figure 8).

With these modifications we lose little generality in assuming that at computation nodes, $g_n: \overline{S} \to \overline{S}$ is defined on any input to the node. We often assume this in the following.

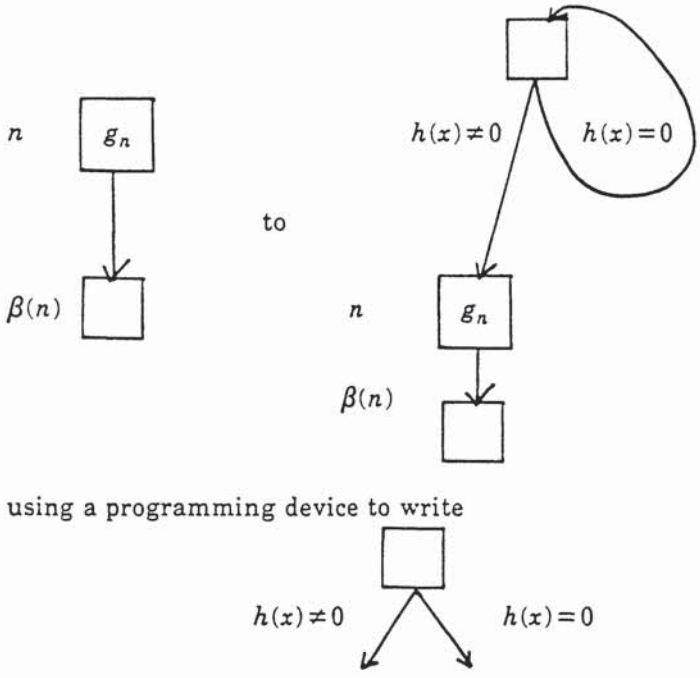

FIGURE 8

Let a machine M over R be given and let $y \in \overline{I}$. The computation on y by M goes in a natural way. First $I(y) \in \overline{S}$ and we are at node 1 in the computation process. If $\beta(1) = n$ is a computation node, the computation g_n is performed on the state $x = I(y)$ to produce $g_n(x)$ which replaces x. If n is a branch node, and $h_n(x) < 0$, the next node is $\beta^-(n)$. Otherwise the next node is $\beta^+(n)$. In both cases, the next state is still x. Thus the computation proceeds until an output node n is reached (if ever) and $O_n(x)$ for some $x \in \overline{S}$ is computed. In this case the computation is said to halt and produces $\varphi_M(y) = O_n(x)$. Denote $\Omega_M \subset \overline{I}$ as the set of y where the computation halts. Ω_M is the *halting set* of M. Thus M defines the input-output map $\varphi_M \colon \Omega_M \to \overline{O}$. Compare the example of §1, which essentially are finite dimensional machines over $\mathbf{R}$.

It is important to allow infinite dimensional machines in the construction of universal machines and to analyze uniform algorithms (algorithms which solve problems with inputs of arbitrarily large size).

We now discuss the modifications needed in the finite dimensional machines to make them infinite. In an infinite dimensional machine we take $\overline{I} = R^l$, $\overline{O} = R^m$ now allowing l, m to be $\leq \infty$, together with a (*finite*) directed graph. The 4 nodes as before are the same from a graph theoretical view, but the definition of the associated maps must be extended to cover the ∞-dimensional case. We have already defined a polynomial map $R^\infty \to R^\infty$. But this will not affect x_i for large i in the expression $x = (x_1, x_2, \dots) \in R^\infty$. This consideration forces us to introduce another type of node into the machine which we call a *fifth node.* A fifth node is designed to access the x_i with large i. For this we take $\overline{S} = \mathbf{Z}^+ + \mathbf{Z}^+ + R^\infty$ with a typical element $(i, j, x_1, x_2, \dots)$, i, j positive integers. A fifth node (one outgoing edge and unique *next node* $\beta(n)$ as in a computation node) operates on this element by transforming it to the element $(i, j, x_1, x_2, \dots, x_i$ replacing x_j in the jth place in $R^\infty, \dots) \in \overline{S}$. No other changes are made in the computation of a fifth node. If n denotes this node, then that map will be written as $g_n \colon \overline{S} \to \overline{S}$.

In the ∞-dimensional case the input map $I \colon \overline{I} \to \overline{S} = \mathbf{Z}^+ + \mathbf{Z}^+ + R^\infty$ can conveniently be taken as follows: non-zero-coordinates are given by $I(x)_{2k+1} = x_k$ for $k \leq l$ leaving starting values $i = 1, j = 1$ in the $\mathbf{Z}^+ + \mathbf{Z}^+$ part of $\overline{S}$. This leaves "working space" in $\overline{S}$.

For technical reasons, we also assume coordinate $I(x)_4$ denotes the length of x. Here, the *length* of a nonzero element $x \in R^l$ is defined as the largest k such that $x_k \neq 0$; the length of the 0 vector is 1.

The $g_n \colon \overline{S} \to \overline{S}$ of a computation node is required to be of the form $g_n(i, j, x) = (i'(i, j), j'(i, j), x'(i, j, x))$ with $i'(i, j) = i + 1$ or 1. (Similarly, $j'(i, j) = j + 1$ or 1) and $x'(i, j, x)$ satisfying the previous defined condition for polynomial (or rational) maps on ∞-dimensional spaces. The branch node polynomial $h \colon \mathbf{Z}^+ + \mathbf{Z}^+ + R^\infty \to R$, we suppose satisfies $\partial h / \partial x_i \equiv 0$, $i >$ some k. (This condition defines a *polynomial function.*) We let k_M denote the maximum dimension of the maps and functions associated with the computation and branch nodes of M. Similarly, d_M is the maximum degree of these maps.

The final modification for the ∞-dimensional case is on the output map $O_n: \overline{S} \to \overline{O}$. Suppose it to be of form $O_n(i, j, x)_k = x_{2k-1}$, $k = 1, 2, \ldots$.

The input-output map φ_M for the infinite dimensional machine M is defined just as in the finite dimensional case. For $l, m \leq \infty$ we say that a (partial) map $\varphi: R^l \to R^m$ is *computable over* R iff there is a machine M over R such that $\Omega_M = \Omega_\varphi$, the domain of φ, and $\varphi_M(x) = \varphi(x)$ for all $x \in \Omega_\varphi$. In such a case we say M *computes* φ. Machines M and M' will be called *equivalent* if they compute the same map.

A set $Y \subset R^n$ will be called R.E. over R if $Y = \Omega_M$ for some machine M over R. It will be said to be *decidable* if it and its complement are both R.E. over R. It is an easy exercise to show that $\mathbf{Z}$ is a decidable subset of R over R for any Archimedean ring R.

Sometimes it is convenient to restrict the inputs of a machine from $\overline{I}$ to a subset $Y \subset \overline{I}$. Thus Y would be considered the space of *admissible* inputs (for some problem, for example). Various notions in our theory may be relativized to such a space of admissible inputs. For example, if $Y \subset \overline{I}$ let $\Omega_{M,Y} = \Omega_M \cap Y$. Then we would say $\Omega_{M,Y}$ is an R.E. *set over* R *relative to* Y.

It is also convenient often to view R^k as being contained in R^l for $k \leq l \leq \infty$ via the natural injection $j: R^k \to R^l$ where $j(x) = (x, 0, \ldots), l - k$ zeros after x. Thus, e.g. we may think of $(1, 0, 0, \ldots) \in R^\infty$ as the element $1 \in R^1 \subset R^\infty$. In the infinite dimensional case, it is often useful to write $R^\infty = R^\infty + \cdots + R^\infty$ (m times) in $\overline{S}$.

When we talk about a machine over R in the sequel, this could mean either the finite dimensional or the infinite dimensional case. For $x \in \overline{S}$ we will sometimes use the notation $x]_k$, $k = 1, 2, \ldots$ to denote the kth coordinate of x in the finite dimensional case, and the $k+2$ coordinate of x if $\overline{S}$ is infinite dimensional. In the latter case, $x]_{-1}, x]_0$ will denote the first 2 coordinates of x respectively.

3. The computing endomorphism and the register equations. In this and the following two sections we develop the machinery and concepts needed to prove our Main Theorem on *NP*-completeness in §6.

First of all, we define a machine over R to be in *normal form* if it satisfies:

(1) At each branch node, n, $h_n(x) = x]_1$, for $x \in \overline{S}$. This is a minor condition making things a bit more convenient.

(2) There is one output node, and hence one output map $O: \overline{S} \to \overline{O}$.

(3) There is a given labeling of the nodes $\{1, 2, \ldots, N\}$ such that

(a) 1 labels the input node,

(b) N labels the output node.

(4) In the finite dimensional case, I is the natural injection and O is the natural projection.

PROPOSITION 1. *Given any machine M over R, there is an equivalent one in normal form.*

PROOF. One achieves property (1) by adding a computation node before and after each branch node using a little straightforward programming

exercise. To obtain (2), one just collapses all of the output nodes to a single one. (And in the finite dimensional case, perhaps expand the state space and number of nodes a bit to obtain $O_n = O$.) The existence of the labeling (3) is clear. For (4) we add computation nodes immediately after the input and before the output nodes.

For a machine M over R in normal form, there is a natural map $H = H_M$ from $\overline{N} \times \overline{S}$ to itself, called the *computing endomorphism* of the machine. Here $\overline{N} = \{1, \ldots, N\}$ is the set of nodes and $\overline{S}$ is the state space. We call $\overline{N} \times \overline{S}$ the *full state space* of M.

The computing endomorphism $H: \overline{N} \times \overline{S} \to \overline{N} \times \overline{S}$ has the form $H(n, x) = (\beta(n, \chi(x)), g_n(x))$ where β describes the next node and g_n the new state. Here $\chi: \overline{S} \to R$ is defined by

$$\chi(x) = \begin{cases} 1 & \text{if } x]_1 > 0, \\ 0 & \text{if } x]_1 = 0, \\ -1 & \text{if } x]_1 < 0. \end{cases}$$

We define $\beta: \overline{N} \times \{0, \pm 1\} \to \overline{N}$ as follows, depending on M:

$$\beta(n, \sigma) = \begin{cases} N & \text{if } n = N, \\ \beta(n) & \text{if } n < N \text{ and } n \text{ is a nonbranching node}, \\ \beta^+(n) & \text{if } n \text{ is branching and } \sigma = 0 \text{ or } 1, \\ \beta^-(n) & \text{if } n \text{ is branching and } \sigma = -1. \end{cases}$$

To define g_n as a function of n and x, let $g_n(x) = x$ for $n = 1$, N or a branch node. At a computation or fifth node we suppose $g_n(x)$ is the computation given by that node.

Thus the *computation* of a machine M over R with input $y \in \overline{I}$ is represented by a sequence $z_0, z_1, z_2, \ldots, z_k, \ldots$ for $z_k \in \overline{N} \times \overline{S}$, $z_0 = (1, I(y))$ and $H(z_{k-1}) = z_k$, $k = 1, \ldots$. In dynamical systems terminology, this sequence of $z_k = (n_k, x_k)$ is the *orbit* of the computing endomorphism with starting point z_0. Note that if M is infinite dimensional and $x_k = (i, j, \ldots)$, then $i, j \leq k$.

We remark that if g_n is a rational map, then $g_n(x)$ may not be defined for some $x \in \overline{S}$. Thus H is a "partial" map. However, by starting with $z_0 = (1, I(y))$ we are assured (by our convention on machines) that H can be iterated without concern.

A necessary and sufficient condition that a sequence $z_0, z_1, \ldots$ be a computation by the machine M is that

$$z_k = H(z_{k-1}), \qquad k = 1, 2, \ldots \tag{1a}$$

and secondarily, $z_0 = (n_0, x_0)$ has the form

$$n_0 = 1, \quad x_0 = I(y), \quad \text{for some } y \in \overline{I}. \tag{1b}$$

Let us call equations (1a),(1b) the register equations of the machines M. Equation (1a) may also be written

$$\beta(n_{k-1}, \chi(x_{k-1})) = n_k \qquad g_{n_{k-1}}(x_{k-1}) = x_k \quad \text{for } k = 1, 2, \ldots. \tag{1a'}$$

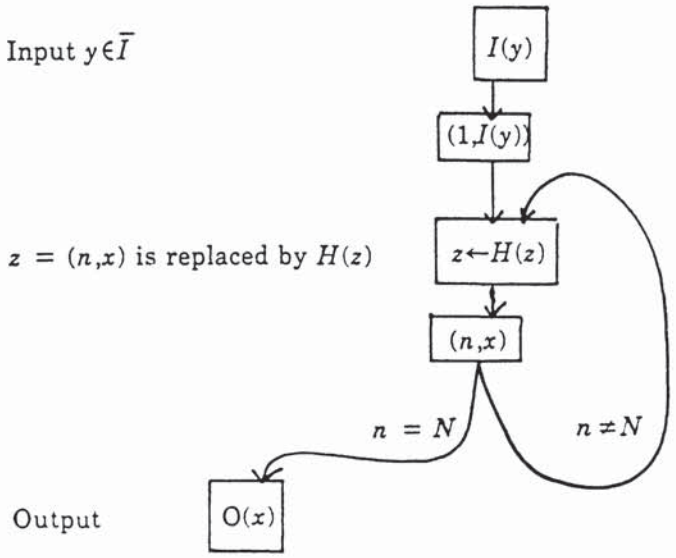

FIGURE 9

In the sequel, symbols such as n_k or x_k will sometimes represent specific elements (in $\overline{N}$ or $\overline{S}$ say), and sometimes variables. The intended usage should be clear from context.

One can represent the computation process of a machine over R by the following flow chart. (See Figure 9.) (Compare this flow chart with the "Julia set machines" of §§1 and 10.)

Thus for given y, one keeps computing H until node N is reached in which case the machine outputs O of that x in $\overline{S}$.

For reasons to become apparent in the next sections, we move to putting the register equations of a machine M over R into a more algebraic form. To do this we first put the computing endomorphism into a more algebraic form, at the same time extending it to an endomorphism of $R' \times R'^n$. Here R' is the ring generated by R and $\mathbf{Q}$ and, in the infinite dimensional case, R'^n is to be interpreted as $R' + R' + R'^{\infty}$ (similarly for R^n).

LEMMA 1. *β can be extended to a polynomial map (also denoted) $\beta: R' \times R' \to R'$ of degree $N + 1$.*

PROOF. For each $n \in \overline{N}$, let

$$a_n(y) = \prod_{j \neq n, j \in \overline{N}} \frac{(y - j)}{(n - j)}.$$

So, for $y \in \overline{N}$,

$$a_n(y) = \begin{cases} 1 & \text{if } y = n, \\ 0 & \text{if not.} \end{cases}$$

Let $B = \{\text{branch nodes of } M\}$. Then

$$\beta(y,\sigma) = \sum_{n\in\overline{N}-B} a_n(y)\beta(n) + \left(\frac{\sigma(\sigma+1)}{2} + (\sigma+1)(1-\sigma)\right)\sum_{n\in B} a_n(y)\beta^+(n) + \frac{\sigma(\sigma-1)}{2}\sum_{n\in B} a_n(y)\beta^-(n).$$

(Note that $\beta(n,\sigma)$ is independent of σ for all nonbranching nodes n.) It can be easily seen that the degree of β is $N+1$ and that β can be written as a sum of $6N$ monomials in the variables y and σ.

Now let $g(n,x) = g_n(x)$. As above, we can extend the definition of $g_n(x)$, or equivalently $g(n,x)$, to all n in R.

A formula for this is

$$g(y,x) = \sum_{n\in\overline{N}} a_n(y)g_n(x),$$

$a_n(y)$ as in the above proof.

Suppose R is a field and the computation at node n takes the form

$$g_n(x) = \left(\frac{p_n^1(x)}{q_n^1(x)}, \frac{p_n^2(x)}{q_n^2(x)}, \dots\right).$$

Here $p_n^l(x)$, $q_n^l(x)$ are polynomials if n is a rational computation node; otherwise, $p_n^l(x) = g_n^l(x)$, the lth coordinate of g, and $q_n^l(x) \equiv 1$. Then the above is modified simply to

$$g^l(y,x) = \frac{\sum_{n\in\overline{N}} a_n(y)p_n^l(x)}{\sum_{n\in\overline{N}} a_n(y)q_n^l(x)}.$$

Since $\sum_{n\in N} a_n(y) \equiv 1$ for $y \in \overline{N}$, the previous formula is recovered for $y \in \overline{N}$, whenever $q_n^l(x) \equiv 1$, all $n \in \overline{N}$.

It is clear that if M is finite dimensional, then g is a polynomial (or rational) endomorphism of $R' \times R'^n$. This is not necessarily the case if M is infinite dimensional due to the existence of fifth nodes.

Nevertheless, if M is infinite dimensional, it *is the case* that for each $k \in \mathbf{Z}^+$, there is a polynomial (or rational) endomorphism $g_{(k)}$ of $R' \times R'^n$ of dimension K_k that is identical to g on $\overline{N}\times\overline{S}_{(k)}$. Here $K_k = \max(k_M, k+2)$ and $\overline{S}_{(k)} = \{(i,j\dots) \in \overline{S} | i,j \le k\}$. Furthermore, these endomorphisms are *uniform* in k. This will follow from the above together with

LEMMA 2. *There is a universal map $F: \mathbf{Z}^+\times R'^n \to R'^n$ such that for each $k \in \mathbf{Z}^+$, $F_{(k)}$ $(= F(k,\cdot))$ is a polynomial endomorphism of R'^n of dimension $k+2$, degree $2k-1$ and $F_{(k)}$ is identical to the fifth node computation on $\overline{S}_{(k)}$.*

PROOF. Let $a: \mathbf{Z}^+ \times \mathbf{Z}^+ \times \mathbf{Z}^+ \times R' \times R' \to R'$ be defined by

$$a(k,i,j,v,w) = \prod_{p\in\overline{k}-\{i\}} \left(\frac{v-p}{i-p}\right) \prod_{q\in\overline{k}-\{j\}} \left(\frac{w-q}{j-q}\right)$$

where $\overline{k} = \{1, \ldots, k\}$. So, for $k \in \mathbf{Z}^+$ and $i, j, v, w \in \overline{k}$,

$$a(k, i, j, v, w) = \begin{cases} 1 & \text{if } (v, w) = (i, j), \\ 0 & \text{otherwise.} \end{cases}$$

Note, for k, i, j fixed, a is a polynomial in v, w of degree $2(k-1)$ which can be written as the sum of k^2 monomials.

For each $l, j \in \mathbf{Z}^+$, let

$$d^l(j) = \begin{cases} 1 & \text{if } l = j + 2, \\ 0 & \text{otherwise.} \end{cases}$$

Then the coordinate functions

$$\begin{aligned} F^l_{(k)}(x) &= F^l(k, x) \\ &= \sum_{(i,j) \in \overline{k} \times \overline{k}} a(k, i, j, x]_{-1}, x]_0)(d^l(j)x]_i + (1 - d^l(j))x^l) \end{aligned}$$

where x^l is the lth coordinate of x, define an appropriate map:

$F_{(k)}$ is identical to fifth node computation on $\overline{S}_{(k)}$ since if $x = (x_{-1}, x_0, x_1, x_2, \ldots)$ and $x_{-1}, x_0 \leq k$, then the only nonzero coefficient a occurs when $i = x_{-1}$ and $j = x_0$, and the only nonzero coefficient d^l occurs when $j + 2 = l$. So $F^{j+2}_{(k)}(x) = x_i$, while for $l \neq j + 2$, $F^l_{(k)}(x) = x^l$.

We also see that $F_{(k)}$ is polynomial of dimension $k+2$ of degree $2k-1$, and can be written as a sum of k^4 monomials (per coordinate function).

Now, letting

$$g_{(k)}(y, x) = \sum_{n \in \overline{N} - \overline{F}} a_n(y) g_n(x) + \sum_{n \in \overline{F}} a_n(y) F_{(k)}(x),$$

where $\overline{F} = \{\text{fifth nodes of } M\}$, we get polynomial (or rational) endomorphisms of $R' \times R'^n$ identical to g on $\overline{N} \times \overline{S}_{(k)}$, uniformly in k.

The dimension of $g_{(k)}$ is $K_k + 1$, the degree is $\max(d_M, 2k-1) + (N-1)$ and the number of monomials needed to describe each coordinate function of $g_{(k)}$ is k^4 plus a constant depending only on M. If there are no fifth nodes, each of these bounds can be replaced by constants depending only on M.

PROPOSITION 2. *There is a universal map* $\overline{H}: \mathbf{Z}^+ \times R' \times R'^n \to R' \times R'^n$ *such that for each* $k \in \mathbf{Z}^+$, $\overline{H}_{(k)}$ $(= \overline{H}(k, \cdot, \cdot))$ *is a composition of polynomials (or rational) maps and the characteristic function* χ, *and* $\overline{H}_{(k)}$ *is identical to the computing endomorphism on* $\overline{N} \times \overline{S}_{(k)}$. *(If* $n < \infty, \overline{S}_{(k)}$ *is to be interpreted as* $\overline{S}$.)

Thus, we can rewrite equations (1a′) as equations involving compositions of polynomials over R and the characteristic function χ,

$$\begin{aligned} c(\beta(n_{k-1}, \chi(x_{k-1})) - n_k) &= 0, \\ c_k(g_{(k)}(n_{k-1}, x_{k-1}) - x_k) &= 0, \end{aligned} \tag{1a″}$$

for $k = 1, 2, \ldots$, where c, c_k are the (smallest) positive integers sufficient to cancel denominators. (If $n < \infty, g_{(k)}$ is just g.) In case M has rational computation nodes, we replace the second set of equations by the coordinate polynomial equations:

$$c_k \left(\sum_{n \in \overline{N} - \overline{F}} a_n(n_{k-1}) p_n^l(x_{k-1}) + \sum_{n \in \overline{F}} a_n(n_{k-1}) F_{(k)}^l(x_{k-1}) \right.$$
$$\left. - x_k^l \sum_{n \in \overline{N}} a_n(n_{k-1}) q_n^l(x_{k-1}) \right) = 0.$$

Note that the coordinate equations of the second set for $l > K_k$ can be written as $x_{k-1}^l - x_k^l = 0$.

Now suppose R is a field with the property that any positive element is a square. Thus R could be the real numbers or any real closed field. With R satisfying the property we can eliminate the characteristic function χ in the above and thus put the register equations into an even finer algebraic form.

To do this we introduce new variables $u_1, u_2, \ldots, u_k, \ldots$ for $k = 1, 2, \ldots$ and replace (1a″) by the following polynomial equations over R.

$$(1a''') \qquad \begin{gathered} x_{k-1}]_1 (x_{k-1}]_1 u_{k-1}^2 + 1)(x_{k-1}]_1 u_{k-1}^2 - 1) = 0 \\ \beta(n_{k-1}, x_{k-1}]_1 u_{k-1}^2) - n_k = 0 \\ g_{(k)}(n_{k-1}, x_{k-1}) - x_k = 0 \end{gathered}$$

for $k = 1, 2, \ldots$.

Again, if M has rational computation nodes, the last set of equations can be modified appropriately.

PROPOSITION 3. *Suppose the field R has the property that positive elements are squares of elements in R, and suppose M is a machine over R. Then the sequence $(n_0, x_0), (n_1, x_1), \ldots (n_k, x_k) \in R \times R^n$ is a computation by M if and only if it satisfies* (1a‴) *for some $u_1, u_2, \ldots$ in R and* (1b). *If these conditions are satisfied then necessarily $(n_k, x_k) \in \overline{N} \times \overline{S}$.*

The proof is straightforward.

We can characterize computations via polynomial sets of equations and (1b) even more generally. For example, if the field R has the property that every positive element is a *bounded* sum of squares (e.g., by Lagrange, every positive rational number is the sum of 4 squares of rationals) we can replace every occurrence of u_{k-1}^2 above by $\sum_{j=1}^{b} u_{(k-1)j}^2$ where b is the given bound and $u_{(k-1)j}$ are new variables for $j = 1, \ldots, b$ and $k = 1, 2, \ldots$.

Over the ring of integers, we can replace (a″) by

$$(1a_z) \qquad x_{k-1}]_1 \left(x_{k-1}]_1 - \sum_{j=1}^{4} u_{(k-1)j}^2 - 1\right)\left(x_{k-1}]_1 + \sum_{j=1}^{4} u_{(k-1)j}^2 + 1\right) = 0$$

$$c\left(x_{k-1}]_1 - \sum_{j=1}^{4} u_{(k-1)j}^2 - 1\right)\left(x_{k-1}]_1 + \sum_{j=1}^{4} u_{(k-1)j}^2 + 1\right)\left(\beta(n_{k-1}, x_{k-1}]_1) - n_k\right) = 0$$

$$cx_{k-1}]_1 \left(x_{k-1}]_1 + \sum_{j=1}^{4} u_{(k-1)j}^2 + 1\right)\left(\beta\left(n_{k-1}, x_{k-1}]_1 - \sum_{j=1}^{4} u_{(k-1)j}^2\right) - n_k\right) = 0$$

$$cx_{k-1}]_1 \left(x_{k-1}]_1 - \sum_{j=1}^{4} u_{(k-1)j} - 1\right)\left(\beta\left(n_{k-1}, x_{k-1}]_1 + \sum_{j=1}^{4} u_{(k-1)j}^2\right) - n_k\right) = 0$$

and

$$c_k(g_{(k)}(n_{k-1}, x_{k-1}) - x_k) = 0$$

where $u_{(k-1)j}$ are new variables for $j = 1, \ldots, 4$ and $k = 1, 2, \ldots$.

4. Time T halting sets, equations, polynomials and computations. Let us consider now halting computations, i.e., those sequences (n_k, x_k) satisfying (1a′) or (1a″), (1b), and with the added condition $n_T = N$ for some $T < \infty$. Such a T will be called a halting time for the y in (1b) and the *halting time* for M to compute $\varphi_M(y)$ will be denoted by $T_M(y)$ or $T(y)$ and is the minimum T such that $n_T = N$. If $T = T(y)$, we then have the pair of equations satisfied:

$$(1c) \qquad n_T = N, \qquad O(x_T) = \varphi_M(y),$$

(O the output map).

Let $\overline{I}_T = \{y \in \overline{I} | T(y) \leq T\}$ be the time T halting set of M. Thus, $\Omega_M = \bigcup \overline{I}_T$, the union over $T \in \mathbf{Z}^+$. We remark that if $y, y' \in \overline{I}$ agree on the first $K_T = \max(k_M, T + 2)$ coordinates, and $y \in \overline{I}_T$, then $y' \in \overline{I}_T$ and $\varphi_M(y')$, $\varphi_M(y)$ agree on the first K_T coordinates.

We now consider the *time T halting equations*

(1a′) or (1a″) for $k = 1, \ldots, T$,
(1b′) $n_0 = 1$, $x_0 = I(y)$,
(1c′) $n_T = N$.

We can use these equations to get algebraic descriptions (modulo χ) for various relations. For example, let $G_{M,T} = \{(y, w) \in R^l \times R^m | w = \varphi_M(y)$ and $y \in \overline{I}_T\}$ be the *time T graph of φ_M*. Then,

> $(y, w) \in G_{M,T}$ *if and only* if there is a $\sigma = ((n_0, x_0), \ldots (n_T, x_T)) \in (R \times R^n)^{T+1}$ such that (y, σ, w) satisfy the time T halting equations *plus* $w = O(x_T)$.

(As before, we interpret R^n as $R + R + R^\infty$ in the infinite dimensional case.)

If M is finite dimensional, these equations are polynomial, modulo the characteristic function χ, and finite. If M is infinite dimensional they

are not finite, nor are I and O polynomial. However, for each T, we can easily modify the time T halting equations to get a *finite* polynomial system (modulo χ) by eliminating all j coordinate equations from the second set in (1a″) and (1b′), for $j > K_T = \max(k_M, T+2)$. We then have

> $(y, w) \in G_{M,T}$ if and only if there is a $\sigma \in (R \times R^{K_T})^{T+1}$ such that (y, σ, w) satisfy the "modified" time T halting equations *plus*
>
> $$w_j = \begin{cases} x_T]_{2j-1} & \text{for } 2j \leq K_T - 1, \\ y_j & \text{for } 2j > K_T - 1. \end{cases}$$

Note, the only nonfiniteness comes in the last equations, and then only in a nonessential way.

Over the *reals*, the time T halting equations are equivalent to a polynomial system (just replace (1a′) by (1a‴) for $k = 1, \ldots, T$). By modifying this system as above we get a finite polynomial system.

We can now use the following proposition to convert this system to a *single* polynomial equation of degree less than or equal to 4 which we call the *time T halting polynomial* equation for M.

PROPOSITION 1. *Over the real numbers,*

(a) *any system of polynomial equations is equivalent to a quadratic system.*

(b) *Any quadratic system is equivalent to a single equation of degree less than or equal to* 4.

Here equivalence means if one system has a solution then so has the other.

For the proof of (a), consider the polynomial equation $\sum_\alpha a_\alpha x^\alpha = 0$, $\alpha = (\alpha_1, \ldots, \alpha_n)$, α_i nonnegative integers. Let $t_\alpha = x^\alpha$ be new variables. One has an equivalent system for the t_α's of the type $t_{\alpha+\beta} = t_\alpha t_\beta$ together with $\sum_\alpha a_\alpha t_\alpha = 0$, α ranging over some set J.

For the proof of (b), one just takes the sum of the squares.

THEOREM. *Let $R = \mathbf{R}$ and let M be a given machine over $\mathbf{R}$. For each $T \in \mathbf{Z}^+$, there is a polynomial $f_T\colon \mathbf{R}^{K_T} \times \mathbf{R}^s \to \mathbf{R}$ of degree ≤ 4 such that $y \in \bar{I}_T$ if and only if there is a $z \in \mathbf{R}^s$ such that $f_T(y_1, \ldots, y_{K_T}, z) = 0$.*

Furthermore, s (and the number of monomials needed to express f_T) is bounded by a polynomial in T, depending only on M. (If the input space is finite dimensional, we replace K_T by l, the dimension of the input space.)

PROOF. The existence of a single polynomial follows from the above discussion and Proposition 1. (z encompasses all the variables in the "modified" time T halting equations, except in y, plus the variables needed to eliminate χ as well as those needed to convert the system into a quadratic one.) To get the polynomial bound on the number of variables (and monomials) we must count the number of variables and monomials appearing in the modified time T halting equations, the number of new variables (and equations) needed to get a quadratic system (and then the number of monomials needed to describe the resulting quartic equation). These

counts are mainly affected by the number of monomials needed to describe the "fifth node" polynomials $f_{(k)}$ for $k = 1, \ldots, T$. One thus uses the analyses given in §3 to get a requisite polynomial bound.

Observe that the construction of the polynomial f_T above is *uniform* in both T and M. We can restate the Theorem to reflect this, and in a somewhat different way.

THEOREM′. *There is a machine over* $\mathbf{R}$ *which on input* (M, T) (*where* M *is a machine over* $\mathbf{R}$ *and* $T \in \mathbf{Z}^+$) *outputs in time polynomial in* l_M (*the "length of* M") *and* T, *a polynomial* $f_{M,T}\colon \mathbf{R}^{K_T} \times \mathbf{R}^{\mathrm{poly}(l_M,T)} \to \mathbf{R}$ *of degree* ≤ 4 *satisfying the following diagram*

$$\begin{array}{ccccc} V = f_{M,T}^{-1}(0) & \subset & \mathbf{R}^{K_T} \times \mathbf{R}^{\mathrm{poly}(l_M,T)} & \xrightarrow{f_{M,T}} & \mathbf{R} \\ \downarrow & & \downarrow \pi_1 & & \\ \overline{I}_{M,T} = \pi_2^{-1}\pi_1(V) & \subset & \mathbf{R}^{\infty} & \xrightarrow[\pi_2]{} & \mathbf{R}^{K_T} \end{array}$$

Here $\overline{I}_{M,T}$ is the time T halting set of M and π_1, π_2 are the projections of $\mathbf{R}^{K_T} \times \mathbf{R}^{\mathrm{poly}(l_M,T)}$ and $\mathbf{R}^{\infty}$ onto R^{K_T} respectively.

(If the input space is finite dimensional, replace K_T by the dimension of $\overline{I}_M$ and ignore π_2.)

To make sense of Theorem′, we have to define the "length of M" and specify how machines M and polynomials f over $\mathbf{R}$ are to be represented in $\mathbf{R}^{\infty}$. This will be done more precisely in the next sections. Polynomials will be specified by their "powerfree" representations (see §5) and machines by their programs (see §8). The "length of M" will then be the *length* (as defined in §5) of the representation in $\mathbf{R}^{\infty}$ of the program for M.

We now develop the relationship between the time T halting sets $\overline{I}_T$ and the *time* T *halting computations* Γ_T. Here $\Gamma_T = \Gamma_{M,T}$ is the subset of $(R \times \overline{S})^{T+1}$ consisting of $((n_0, x_0), \ldots, (n_T, x_T))$ satisfying, for some $y \in \overline{I}$, the time T halting equations. (Note we automatically get that $y \in \overline{I}_T$ and $O(x_T) = \varphi_M(y)$.)

There are natural maps α, α' defined as follows:

$$\overline{I}_T \xrightarrow{\alpha} \Gamma_T \xrightarrow{\alpha'} \overline{I}_T,$$

$$\alpha(y) = ((1, I(y)), H(1, I(y)),\ H^2(1, I(y)), \ldots, H^T(1, I(y))) \text{ and}$$

$$\alpha'((n_0, x_0), \ldots, (n_T, x_T)) = y$$

where $n_0 = 1$ and $x_0 = I(y)$.

Then α and α' are inverses to each other and provide a set-theoretical isomorphism between $\overline{I}_T$ and Γ_T. Moreover α' is continuous and even is the restriction of a linear map. But α in general is not continuous, since H is not (recall H involves a characteristic function).

Now consider the map $\gamma_1\colon \Gamma_T \to R^{T+1}$ which is the restriction of the projection $(R \times \overline{S})^{T+1} \to R^{T+1}$. In fact $\gamma_1(\Gamma_T) \subset \overline{N}^{T+1} \subset R^{T+1}$. Define $\gamma\colon \overline{I}_T \to \overline{N}^{T+1}$ as the composition $\gamma_1 \cdot \alpha$. One can interpret $\gamma(y)$ as a *halting path* or as the *computation path* of y of *length* T in the directed graph of the machine M. So $\gamma(y)$ is the sequence of nodes traversed in

the computation $\varphi_M(y)$. If $\gamma \in \overline{N}^{T+1}$, let V_γ be the subset of $\overline{I}_T$ such that $\gamma(y) = \gamma$. Then it is easy to see that α restricted to V_γ is a continuous map from V_γ into Γ_T.

PROPOSITION 2. (a) *V_γ is a semialgebraic subset of $\overline{I}_T$, (basic, if the maps at computation nodes are polynomial), and φ_M restricted to V_γ is a rational map.*

(b) *$\overline{I}_T = \bigcup_{\gamma \in N^{T+1}} V_\gamma$ is semialgebraic. The V_γ's are disjoint.*

(c) *$\Omega_M = \bigcup_{T>0} \overline{I}_T$ is a countable union of semialgebraic sets.*

PROOF. Only (a) needs proof. By following the path γ and noting the branches taken, one sees that V_γ is defined by inequalities of the type

$$g_{k_l}(\cdots g_{k_2}(g_{k_1}(I(y))))]_1 < 0 \quad \text{and} \quad g_{k_m}(\cdots g_{k_2}(g_{k_1}(I(y))))]_1 \leq 0$$

where the g's are polynomial (or rational). If the g's are rational, each inequality is replaced by a disjunction of polynomial inequalities by using the correspondence: $p/q < 0$ iff ($p < 0$ and $q > 0$) or ($p > 0$ and $q < 0$). Similarly, by composing the computations along the path γ, one can see that φ_M restricted to V_γ is of the form $Og_{j_n}(\cdots g_{j_2}(g_{j_1}(I(y))))$.

If M is finite dimensional we are done. If M is infinite dimensional, we can, as in the modified T halting equations, use the bound K_T on the number of "active" coordinates and variables in a computation of length T to get semialgebraic descriptions of V_γ, and polynomial (or rational) descriptions of φ_M restricted to V_γ.

Note that M with inputs restricted to $\overline{I}_T$ is essentially a finite dimensional machine. Moreover, this machine is equivalent to one without loops, a tree as in *Smale*.

We remark that (b) gives an algebraic description of $\overline{I}_T$ via an exponential (in T) number of semialgebraic formulas. Compare this with the time T halting polynomial description of $\overline{I}_T$ given by the previous theorems for the case $R = \mathbf{R}$.

As discussed in §1, it follows from Proposition 2 that Julia sets in general are undecidable. There are other immediate applications. For example, we now easily get a lower bound on the halting time for computing the "greatest integer in." (See Example 5 in §1.)

PROPOSITION 3. *Suppose a machine M computes $\lfloor x \rfloor$ for $x \in \mathbf{R}^+$. Then $T_M(x) \geq \log(x)$ for an unbounded set of x.*

PROOF. Fix such a machine M. For each $L \in \mathbf{Z}^+$ let T_L be the maximum of the halting times $T_M(x)$ for $x \leq L$, $x \in \mathbf{R}^+$. (We can suppose $T_L < \infty$.) So for $x \leq L$, $x \in \mathbf{R}^+$ we have $x \in \overline{I}_{T_L}$, which is the union of at most 2^{T_L} sets V_γ, γ of length T_L.

On the other hand, for each nonnegative integer l, there is an open set U of points in $\mathbf{R}$ such that for all $x \in U$, $\varphi_M(x) = l$. If $l < L$, there must be such an open set in V_γ for some γ of length T_L. But φ_M restricted to V_γ is polynomial (or rational). So, φ_M restricted to V_γ must be identically equal to l. Thus, there must be *at least* L such sets V_γ.

So $2^{T_L} \geq L$, i.e., $T_L \geq \log L$.

In a similar fashion, we get lower bounds for the Travelling Salesman Problem over **R**. (See Example 7 in §1.)

Suppose M solves the TSP over **R**, i.e., for each mileage chart A over **R**, $\varphi_M(A)$ is a tour of minimum distance. (Of course, we are supposing $\overline{I} = \overline{O} = \mathbf{R}^\infty$ and some specified representation of mileage charts and tours in $\mathbf{R}^\infty$. For example, an $n \times n$ matrix A could be represented in $\mathbf{R}^\infty$ as n followed by a listing of the rows of A. A tour t of n cities could be represented as an ordering $(i_1, \ldots, i_n)$ of the integers $1, \ldots, n$ where $i_1 = 1$. Thus, $t(i_j) = i_{j+1}$ for $j = 1, \ldots, n-1$ and $t(i_n) = 1$.) For $n \in \mathbf{Z}^+$ let $T_M(n) = \max T_M(A)$, where A ranges over all n-city mileage charts over **R**. The *topological complexity* of M, as a function of n, is the number of halting paths of length $T_M(n)$.

PROPOSITION 4. *Suppose M solves the TSP over* **R**. *Then the topological complexity of M is at least $(n-1)!/2$. So*

$$T_M(n) \geq \log \frac{(n-1)!}{2}.$$

The proof is similar to above, noting that for each tour t, there is an open set of inputs U such that for all $A \in U$, $\varphi_M(A) = t$ or $\bar{t}$ (where t is the reverse of t). So, there must be such an open set in V_γ for some γ of length $T_M(n)$. So, since φ_M restricted to V_γ is polynomial (or rational), we have φ_M restricted to V_γ identically t or $\bar{t}$. The result now follows since there are $(n-1)!/2$ (unoriented) tours over n cities.

The *topological complexity* of the TSP measures the branching necessary and sufficient to solve all n-city instances. (It could be thought of as the "obstruction" to obtaining a rational formula solution.) This notion can be studied more generally. For example, a slightly more subtle argument shows the topological complexity of the Knapsack Problem over **R** is 2^n. (The Knapsack Problem is: Given $(x_1, \ldots, x_n) \in \mathbf{R}^n$. Find a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} x_i = 1$, if one exists.) For lower bounds on the topological complexity of solving 1 variable polynomial equations, see *Smale*.

REMARK. In the above two examples the machines compute functions that take values in a discrete set. We used open set arguments to get estimates on the number of components V_γ of $\overline{I}_T$, thus getting our lower bounds. If these open set arguments were not applicable, we could still use estimates on the number of connected components of $\overline{I}_T$ (see e.g., *Milnor* and *Thom*) to get lower bounds.

5. Complexity theory of machines over R. A background reference for this section and the next is Garey-Johnson, which gives an account of the ideas of *NP*-completeness of Cook and Karp. Here we are considering a version of this theory, more algebraic and especially over the real numbers. There are some differences in substance.

Toward defining the property, "a machine M over a ring R is in class P" (polynomial time), we introduce the notion of size. The length of a nonzero element $x \in R^n$ $(n \leq \infty)$ is defined as the largest k such that $x_k \neq 0$, where $x = (x_1, x_2, \ldots, x_k, 0, 0, \ldots)$. The length of 0 is 1. The size

of x is to be the length of x plus the height of x where the height is yet to be defined. The *height* of x is the maximum of the height (x_i) over all i, and it remains to say what the height is for an element of R. We only do this here for the cases $R = \mathbf{Z}$ the integers, $\mathbf{Q}$ the rational numbers and $\mathbf{R}$ the real numbers. If $R = \mathbf{Z}$, $x \in R$, then *height* $x = \log(|x| + 1)$. If $R = \mathbf{Q}$, $x = p/q \in R$, p, q relatively prime integers, then *height* x is the maximum of height p, height q. Finally if $R = \mathbf{R}$ and $x \in R$, then *height* $x = 1$. Note the difference in the case of $x \in \mathbf{Q} \subset \mathbf{R}$ depending on which field is considered. For some other rings R, the notion of height (e.g. *Mazur*) from algebraic number fields is suggestive.

The height over $\mathbf{Z}$ of an integer is essentially the number of bits. The height over $\mathbf{R}$ reflects the fact that as in scientific computation, the cost of multiplication is independent of the magnitude of a real number.

PROPOSITION 1. *Let M be a machine over R and for $y \in \overline{I}$, let $T_M(y)$ be the halting time as in the previous section. Then $a(y) \leq kT_M(y)$ where $a(y)$ is the number of arithmetic operations and uses of 5th nodes used to compute $\varphi_M(y)$. The constant k depends only on the machine M.*

The proof is straightforward from the fact that M has only a finite number of nodes. Thus, there is a bound on the degree and the number of active variables and coordinates over all the computations at nodes.

We now define the *standard cost function* $C_M(y)$ of machine M on input y to be the product of the time, $T_M(y)$, and the maximum height, $h_M(y)$, occurring in the computation of $\varphi_M(y)$. That is,

$$h_M(y) = \max_{0 \leq k \leq T_m(y)} \text{height}(x_k') \text{ where } H^k(1, I(y)) = (n_k, x_k) \text{ some } n_k \in \overline{N}.$$

where x_k' is identical to x_k on its first k_M coordinates, 0 elsewhere. Note that for $R = \mathbf{R}$, $C_M(y) = T_M(y)$. For $R = \mathbf{Z}$, $C_M(y)$ is polynomially related to the "bit complexity" of computing $\varphi_M(y)$.

Now say that a *machine M over R is in class P (polynomial time)* (or that the *computable function φ_M is in class P over R*) if there are constants c and $q \in \mathbf{Z}^+$ such that

$$C_M(y) \leq c(\text{size}(y))^q \quad \text{all } y \in \overline{I}.$$

Here and in what follows, often sense is made only in case height has been defined. This includes $R = \mathbf{Z}$, $\mathbf{R}$ at least. For $R = \mathbf{Z}$, class P over $\mathbf{Z}$ is identical to the classical polynomial time class with respect to the bit complexity measure of cost.

Suppose $Y \subset \overline{I}$ is a space of admissible inputs. We say that the *pair (M, Y) is in class P* if

$$C_M(y) \leq c(\text{size}(y))^q \quad \text{all } y \in Y.$$

A decision problem over R is a pair (Y, Y_{yes}) where $Y_{yes} \subset Y \subset \overline{I} = R^n$, $n \leq \infty$. A *problem instance* is a point $y \in Y$, with question: "Is y in Y_{yes}?" This formulation of a decision problem equates it with a membership problem.

An *algorithm* (or sometimes machine) which solves a decision problem (Y, Y_{yes}) over R is a pair (M, Y), where M is a machine with space of *admissible inputs* Y such that $\varphi_M(y) = 1$ (*yes*) or 0 (*no*) for all $y \in Y$, and $\varphi_M(y) = 1$ if and only if $y \in Y_{yes}$.

Then we say that the decision problem (Y, Y_{yes}) is in class P if there is an algorithm in class P which solves it.

PROBLEM 5.1. Would the class of decision problems in P over $\mathbf{Z}$ increase if the cost function were changed to $T_M(y)$? (Certainly, the class of polynomial time computable functions over $\mathbf{Z}$ would increase, e.g., consider the function $f(x) = |x|^{|x|}$.)

We will say that a decision problem (Y, Y_{yes}) over R is in class NP (*nondeterministic polynomial time*) if there are constants c and $q \in \mathbf{Z}^+$ and a machine M over R with $\overline{I}_M = \overline{I} \times \overline{I}'$, where $\overline{I} = \overline{I}' = R^n$, $n \leq \infty$ and the space of admissible inputs for M is $Y \times \overline{I}'$, such that

(a) the values of φ_M are 1 (= *yes*) and 0 (= *no*).

(b) $\varphi_M(y, y') = 1$ only if $y \in Y_{yes}$ and,

(c) for each $y \in Y_{yes}$, there is a $y' \in \overline{I}'$ such that $\varphi_M(y, y') = 1$ and $C_M(y, y') \leq c(\text{size}(y))^q$.

The $y' \in \overline{I}'$ are the guesses as in the standard NP-completeness theory (see *Garey-Johnson*). It is easy to see that without loss of generality we can assume, in (c) that the size $(y') \leq c'$ size $(y)^q$, for some fixed $c' \in \mathbf{Z}^+$. Again, if $R = \mathbf{Z}$, our class NP coincides with the classical one.

PROPOSITION 2. *$NP \supset P$ over R.*

The proof goes by using the machine for P and ignoring the guess.

The following two examples illustrate our notion of NP over the real numbers $\mathbf{R}$.

PROPOSITION 3. *The Traveling Salesman decision problem over $\mathbf{R}$ is in NP.*

(See Example 7 in §1.)

PROOF. We write a flow chart for the NP machine as follows (Figure 10):

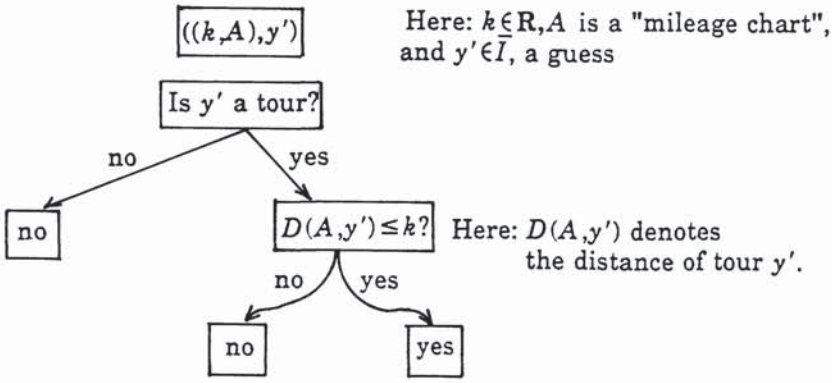

FIGURE 10

The proof now focuses on the subroutine, "Is y' a tour?" Suppose mileage charts and tours are represented as in §4. Let n be the number of cities. To check if y' represents a tour of n cities just check if each integer $1, \ldots, n$ appears once and only once in $(y'_1, \ldots, y'_n)$ and $y'_1 = 1$. Now it can be seen that the subroutine can be implemented by a polynomial time algorithm.

The second example is the 4-*Feasibility problem*, a certain feasibility problem for real algebraic varieties, which we write (F, F_{yes}). Here F consists of polynomials $f\colon \mathbf{R}^n \to \mathbf{R}$ of degree ≤ 4, and $f \in F_{yes}$ if there is some $x \in \mathbf{R}^n$ such that $f(x) = 0$. It remains to be specified how $F \subset \mathbf{R}^\infty$. To do this we describe the *powerfree representation*:

The polynomial $f\colon \mathbf{R}^n \to \mathbf{R}$ of degree ≤ 4 is *powerfreely represented* in $\mathbf{R}^\infty$ as $(4, n)$ followed by a sequence of (α, a_α) where $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, $\alpha_i \in [0, \ldots, n]$, $\alpha_i \leq \alpha_{i+1}$ and $a_\alpha \in \mathbf{R}$. The pair (α, a_α) stands for the monomial $a_\alpha x_{\alpha_1} x_{\alpha_2} x_{\alpha_3} x_{\alpha_4}$, with $x_0 = 1$ to allow for terms of degree less than 4. These (α, a_α) are supposed ordered by the lexicographic order on the α. Thus $f(x) = \sum_\alpha a_\alpha x_{\alpha_1} x_{\alpha_2} x_{\alpha_3} x_{\alpha_4}$. Note f can be considered as a polynomial on $\mathbf{R}^\infty$ which does not depend on x_i for $i > n$. (For each degree $d \in \mathbf{Z}^+$, it is clear how to generalize this description to get the *powerfree representation* in $\mathbf{R}^\infty$ of polynomials $f\colon \mathbf{R}^n \to \mathbf{R}$ of degree $\leq d$.)

PROPOSITION 4. *(F, F_{yes}) is in NP over* $\mathbf{R}$.

The NP machine takes guesses for $f\colon \mathbf{R}^n \to \mathbf{R}$, points y' in $\mathbf{R}^\infty$ and tests if $f(y') = 0$. Since degree $f \leq 4$, this evaluation is given by a polynomial time machine.

PROBLEM 5.2. Does $P = NP$ over $\mathbf{R}$?

6. NP-completeness and the analogue to Cook's theorem over R. Inspired by the theory of NP completeness in computer science (see *Garey-Johnson*) we say that a decision problem $(\hat{Y}, \hat{Y}_{yes})$ is *NP complete* if it is in NP and: Given any decision problem (Y, Y_{yes}) in NP, there is a map $\psi\colon Y \to \hat{Y}$ with these properties:

(a) $\psi(y) \in \hat{Y}_{yes}$ if and only if $y \in Y_{yes}$.

(b) $\psi = \varphi_M | Y$ for some machine M, and this machine is in class P. In other words ψ is polynomial time computable.

This definition is meant over a ring R and a definition of height over R is needed. In particular, this is satisfied if $R = \mathbf{Z}$ in which case our definition checks with the classical definition. Also the case of $R = \mathbf{R}$ is included, in which case, we have a new definition of NP complete. This is the case of primary interest in what follows; but the case of real closed fields should be the same.

MAIN THEOREM (ANALOGUE OF COOK'S THEOREM FOR $\mathbf{R}$). *The 4-Feasibility problem (F, F_{yes}) of the previous section is NP complete over* $\mathbf{R}$.

The proof uses the machinery and results of the last three sections.

Note first that (F, F_{yes}) has already been proved to lie in NP, Proposition 4 of the previous section. Let M be the "nondeterministic machine" in the definition of NP for the problem (Y, Y_{yes}). In this definition there is the polynomial bound $c(size(y))^q = T_0(y)$. We must describe the map $\psi: Y \to F$ with the requisite properties. Thus let $y \in Y$ and consider in the space $(R \times \overline{S})^{T+1}$, $T = T_0(y)$, the equations (a‴) for $k = 1, \ldots, T$. We adjoin to this set, the following

(2) $n_0 = 1$, $I(y, y') = x_0$. (Here y' is a free variable of length K_T.)

(3) $n_T = N$, and $O(x_T) = 1$ (yes).

LEMMA. *For $y \in Y$, this system of equations* (1a‴), *for $k = 1, \ldots, T(2)$ and* (3) *has a solution if and only if $y \in Y_{yes}$.*

PROOF. One just has to trace through all the definitions.

This system is equivalent to the time T halting equations of §4 (now with input space $\overline{I} \times \overline{I}'$) *plus* the equation $1 - x_T]_1$. Thus, it is easily modified to an equivalent *finite* polynomial system (we have already eliminated χ).

As in the Theorem in §4, we can convert this system to a single polynomial equation $f: \mathbf{R}^n \to \mathbf{R}$ of degree less than or equal to 4, to obtain using the powerfree representation, $\psi: Y \to F$. By the previous lemma, $\psi(y)$ is in F_{yes} if and only if $y \in Y_{yes}$. It remains to see that ψ is a polynomial time computable map.

For this one notes that the construction of ψ makes the computability clear. Moreover, since $T = T_0(y)$ is a polynomial in the size of y, it is only left to see that the length of (the powerfree representation of) $\psi(y)$ is a polynomial in T. Here one uses the analyses given in §§3 and 4.

COROLLARY. *Any algorithm for the feasibility problem (e.g. from Tarski-Seidenberg, see van den Dries, or the faster algorithms of Canny and Renegar) can be used to solve any problem in NP over* **R**. *If that algorithm is a polynomial time algorithm, then any problem in NP over* **R** *can be solved in polynomial time.*

This is a usual motivation for studying NP-completeness (Cook-Karp) see *Garey-Johnson.*

Our theorem above raises many questions. Among them is: what are other NP-complete problems over **R**? We only have very preliminary results in this direction as

PROPOSITION 2. *Fixing a degree d, let F'_d be the space of all semialgebraic sets defined by polynomial constraints of degree d, powerfreely represented. Let $F'_{d\,yes}$ be the nonempty ones. Then $(F'_d, F'_{d\,yes})$ is NP-complete.*

PROOF. The reduction is given by the inclusion map $F \to F'_d$.

PROPOSITION 3. *Let F'' be the set of all representations (powerfree) of polynomial systems of equations of the type $t_i t_j = t_k$, and one equation $\sum_{i \in J} t_i = c$. Let F''_{yes} be the feasible ones. Then (F'', F''_{yes}) is NP-complete.*

PROOF. This follows from the proof of Proposition 1.

REMARK. Finally note that the map $f\colon \mathbf{R}^n \to \mathbf{R}$, $f \in F$, gives a reduction to a one dimensional problem. Does $0 \in \text{Image } f$? (The image of f is an interval. Of course the description of the interval is not the standard one.) This presumably can be defined as an NP-complete problem.

7. Computable functions, normal forms and partial recursive functions over R. We deal mainly with the finite dimensional case. Suppose $l, m < \infty$ and let $f\colon R^l \to R^m$ be a partial function (map) computable over R. We denote this by writing $f \in C_R^{<\infty}$. Suppose M computes f. *Added in proof*: Without loss of generality we can assume M is finite dimensional.[4] We may also assume M is in normal form. Using the (partial) computing endomorphism $H\colon \overline{N} \times \overline{S} \to \overline{N} \times \overline{S}$ for M we get a *normal form* description for f (and φ_M): $f(x) = O(h_2(\min_t(h_1(t,x) = N), x))$ where $h_1\colon \Omega_{h_1} = \mathbf{Z}^{\geq 0} \times \overline{I} \to \overline{N}$ and $h_2\colon \Omega_{h_2} = \mathbf{Z}^{\geq 0} \times \overline{I} \to \overline{S}$ are given by $h_1(t,x) = H^t(1, I(x))]_{\overline{N}}$ and $h_2(t,x) = H^t(1, I(x))]_{\overline{S}}$ and $\min_t(h_1(t,x) = N) = \min\{t \in Z^{\geq 0} | h_1(t,x) = N\}$. (Here, $\mathbf{Z}^{\geq 0} = \mathbf{Z}^+ \cup \{0\}$ and we are using the bracket notation as before to indicate projection. Thus $]_{\overline{N}}$ means projection onto the first coordinate, and $]_{\overline{S}}$ means projection onto the remaining coordinates. Also, H^t means H composed with itself t times, H^0 is the identity.)

Inspired by this normal form description and by classical recursive function theory (e.g. see *Davis, Manin, Cutland*), we give a function-theoretic characterization of the computable functions over an arbitrary ring R.

DEFINITION. The class $P_R^{<\infty}$ of finite dimensional *partial recursive functions over R* is the smallest class of partial functions (maps) $f\colon R^l \to R^m$ $(l, m < \infty)$ together with their domains Ω_f, containing the basic functions:

(i) the polynomial (rational, if R is a field) functions $f\colon R^l \to R$ (and hence e.g. the successor, constant and coordinate projection functions), with their natural domains,

---

[4] In our original manuscript, we stipulated that M be finite dimensional in this definition, and posed the problem: Suppose a machine M over R has finite dimensional input and output space (but $\overline{S}$ may be ∞-dimensional). Is M equivalent to a finite dimensional machine (one with $\dim \overline{S} < \infty$ as well)? We have with Leo Harrington answered this affirmatively, thus the class of computable finite dimensional functions remains the same with or without the condition that M be finite dimensional. The idea of the proof is to simulate M by a finite dimensional machine M'. The state space of M' has the same initial $D_M (= \max(2l+2, 2m+2, k_M))$ coordinates as $\overline{S}_M$, then l coordinates to hold a (fixed) copy of input $x \in R^l$. In addition there is a coordinate to hold a single integer that will "code" at each stage the sequence of computations that if started on $x \in R^l$ will produce all current nonzero entries x_i of $\overline{S}_M$. The code uses prime powers and the node labels $\{1, \ldots, N\}$. At each *non* fifth node stage in M's computation, M' behaves as M on its initial coordinates, and also updates the code as necessary. At a fifth node stage in M's computation, with $(i, j, x_1, x_2, \ldots) \in \overline{S}_M$, if $j \leq D_M$ then M' puts x_i in the jth place (if $j > D_M$ no replacement is made): If $i \leq D_M$ this is done by a polynomial map P_{ij}. If $i > D_M$, then M reconstructs x_i from the (fixed) input and current code and the replacement is then made by a polynomial map p_j. (Several additional coordinates are used to construct and unravel the code and perhaps D_M more for reconstructing x_i.) The code is updated by replacing the sub code for the jth place (given e.g. by the exponent of the jth prime factor of the code) by the subcode for the ith place. Subsequent to our solution, another was given by Harvey Friedman and Richard Mansfield.

(ii) the characteristic function $\chi\colon \Omega_\chi = R \to \{0, \pm 1\}$ where

$$\chi(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0, \end{cases}$$

and closed under the operations of

(a) composition,

(b) juxtaposition,

(c) primitive recursion and

(d) minimalization.

For (a) we suppose $f\colon R^l \to R^m$ and $g\colon R^m \to R^n$ are given partial functions. Then the *composition* $g \cdot f\colon R^l \to R^n$ is defined by $g \cdot f(x) = g(f(x))$ where $\Omega_{g \cdot f} = f^{-1}\Omega_g$.

Next for (b), suppose $f_i\colon R^l \to R^{m_i}$ $(i = 1, \ldots, k)$ are given and that ψ is the natural isomorphism mapping $R^{m_1} \times \cdots \times R^{m_k}$ onto $R^{m_1 + \cdots + m_k}$. (We often identify $R^{m_1} \times \cdots \times R^{m_k}$ with $R^{m_1 + \cdots + m_k}$ without writing ψ.) Then the *juxtaposition* $F = (f_1, \ldots, f_k)\colon R^l \to R^{m_1 + \cdots + m_k}$ (of the f_i) is given by $F(x) = \psi(f_1(x), \ldots, f_k(x))$ for $x \in \Omega_F = \bigcap_{i=1}^k \Omega_{f_i}$. Note that juxtaposition of polynomial functions gives us the polynomial maps $f\colon R^l \to R^m$, $(m = m_1 + \cdots + m_k)$ and thus we also get the general projections.

For (c), we suppose we are given a partial endomorphism $g\colon R^l \to R^l$. *Primitive recursion* then defines a partial map $G\colon \mathbf{Z}^{\geq 0} \times R^l \to R^l$ by $G(0, x) = x$ and $G(t + 1, x) = g(G(t, x))$ and

$$\Omega_G = \{(n, x) \in \mathbf{Z}^{\geq 0} \times R^l \mid n = 0 \text{ or } n = t + 1, (t, x) \in \Omega_G \text{ and } G(t, x) \in \Omega_g\}.$$

So, for $t \in \mathbf{Z}^{\geq 0}$, $G(t, x) = g^t(x) = g \cdot \underbrace{\cdots}_{t} \cdot g(x)$ (g composed with itself t times applied to x).

Finally, given $F\colon \Omega_F \supseteq \mathbf{Z}^{\geq 0} \times R^l \to R$, the operation of *minimalization* (d) defines a partial function $L\colon R^l \to \mathbf{Z}^{\geq 0}$ where $\Omega_L = \{x \in R^l \mid F(t, x) = 0$ for some $t \in \mathbf{Z}^{\geq 0}\}$ and $L(x) = \min_t(F(t, x) = 0) = \min\{t \in \mathbf{Z}^{\geq 0} \mid F(t, x) = 0\}$.

Now suppose M is finite dimensional. Then the computing endomorphism for M is given by $H(n, x) = (\beta(n, \chi(x]_1)), g_n(x))$ where β and $g(n, x) = g_n(x)$ are polynomial (rational, if R is a field). (See §3.) Recall that the coefficients of β and g are in the ring generated by R and $\mathbf{Q}$. Since R need not contain $\mathbf{Q}$, some modifications are necessary. We first define a partial recursive function over R which for $x, y \in \mathbf{Z}^{\geq 0}$, $y \neq 0$ gives the "greatest integer in" x/y: $\lfloor x/y \rfloor = \min_t(F(t, x, y) = 0)$ where $F(t, x, y) = \chi(y(t + 1) - x) - 1$. Now, for $n \in \overline{N} = \{1, \ldots, N\}$ let

$$\overline{a}_n(y) = \prod_{\substack{j \neq n \\ j \in \overline{N}}} \left\lfloor \frac{(y - j)\chi(y - j)}{(n - j)\chi(n - j)} \right\rfloor.$$

Let

$$\overline{\beta}(y,\sigma) = \sum_{n\in\overline{N}-B} \overline{a}_n(y)\beta(n) + \left(\left\lfloor \frac{\sigma(\sigma+1)}{2}\right\rfloor + (\sigma+1)(1-\sigma)\right)\sum_{n\in B}\overline{a}_n(y)\beta^+(n) + \left\lfloor \frac{\sigma(\sigma-1)}{2}\right\rfloor \sum_{n\in B}\beta^-(n).$$

(As before $B = \{\text{branch nodes of } M\}$.) Now let $\overline{g}(y,x) = \sum_{n\in\overline{N}} \overline{a}_n(y)g_n(x)$. All the above functions are partial recursive over R. Thus (using the basic functions, composition, juxtaposition and the above), $\overline{H}\colon R\times\overline{S}\to R\times\overline{S}$ defined by $\overline{H}(y,x) = (\overline{\beta}(y,\chi(x]_1)), \overline{g}(y,x)$ is partial recursive over R. And also, $\overline{H}|_{\overline{N}\times\overline{S}} = H$.

Thus $\overline{H}$ is a partial recursive endomorphism over R. So (using primitive recursion and appropriate projections in addition to juxtaposition, composition and noting that I and O are basic), we see that h_1 and h_2 are partial recursive over R, each with domain $\mathbf{Z}^{\geq 0}\times\overline{I}$. It follows (using minimalization and the normal form description) that the input-output map φ_M is partial recursive over R. Thus, any finite dimensional computable functions over R is partial recursive over R.

The converse follows by reasonably straightforward programming which we sketch schematically below.

First, it is clear that the basic functions are computable. Thus, we need only show that the computable functions are closed under the prescribed operations.

(a) COMPOSITION (BY JOINING MACHINES). Suppose f and g are computable via M_f and M_g and $f(\Omega_f)\subset\Omega_g$. Construct $M = M_{g\cdot f}$ as follows:

Let $\overline{I} = \overline{I}_f$ the input space of M_f, $\overline{O} = \overline{O}_g$ the output space of M_g, and $\overline{S} = R^m$ where m is the maximum dimension of all spaces occurring in M_f and M_g. Then schematically, for $x\in\Omega_{g\cdot f}$, M looks like (Figure 11).

(b) JUXTAPOSITION (BY "PARALLEL" PROCESSING, SEQUENTIALLY AS NECESSARY). Suppose f_i are computable via M_{f_i} and $\overline{I}_{f_i} = R^n (i = 1,\ldots,k)$. Construct $M = M_{(f_1,\ldots,f_k)}$: Let $\overline{I} = R^n$, $\overline{O} = \psi(\overline{O}_{f_1}\times\cdots\times\overline{O}_{f_k})$ and $\overline{S} = R^{km}$ where m is the maximum dimension of all spaces occurring in each M_{f_i}. Then, for $x\in\Omega_{(f_1,\ldots,f_k)}$ (see Figure 12).

(c) PRIMITIVE RECURSION (BY LOOPING WITH COUNTDOWN). Suppose $g\colon R^l\to R^l$ is computable via M_g. Let $\overline{I}_g$, $\overline{O}_g$, $\overline{S}_g$ be the corresponding input, output, and state spaces. We construct $M = M_G$ as follows: Let $\overline{I} = R\times\overline{I}_g$, $\overline{O} = \overline{O}_g$ and $\overline{S} = R\times\overline{I}_g\times\overline{S}_g\times\overline{O}_g$. For $z\in\overline{S}$, let $z = (z_1, z_2, z_3, z_4)$ where $z_1\in R$, $z_2\in\overline{I}_g$, $z_3\in\overline{S}_g$ and $z_4\in\overline{O}_g$. Then for $(t,x)\in R\times R^l$, M looks like (Figure 13).

(d) MINIMALIZATION (EVALUATE AND COUNT-UP). Finally, suppose $F\colon \Omega_F = \mathbf{Z}^{\geq 0}\times R^l\to R$ is computable via M_F and let $L(x) = \min_t(F(t,x) = 0)$. To construct $M = M_L$, let $\overline{I} = R^l$, $\overline{O} = R$ and $\overline{S} = \overline{I}_F\times\overline{S}_F\times\overline{O}_F$. For $z\in\overline{S}$ let $z = (z_1, z_2, z_3, z_4)$ where $z_1\in Rz_2\in R^l$ (so $(z_1,z_2)\in\overline{I}_F$), $z_3\in\overline{S}_F$ and $z_4\in\overline{O}$. Schematically, M looks like (Figure 14).

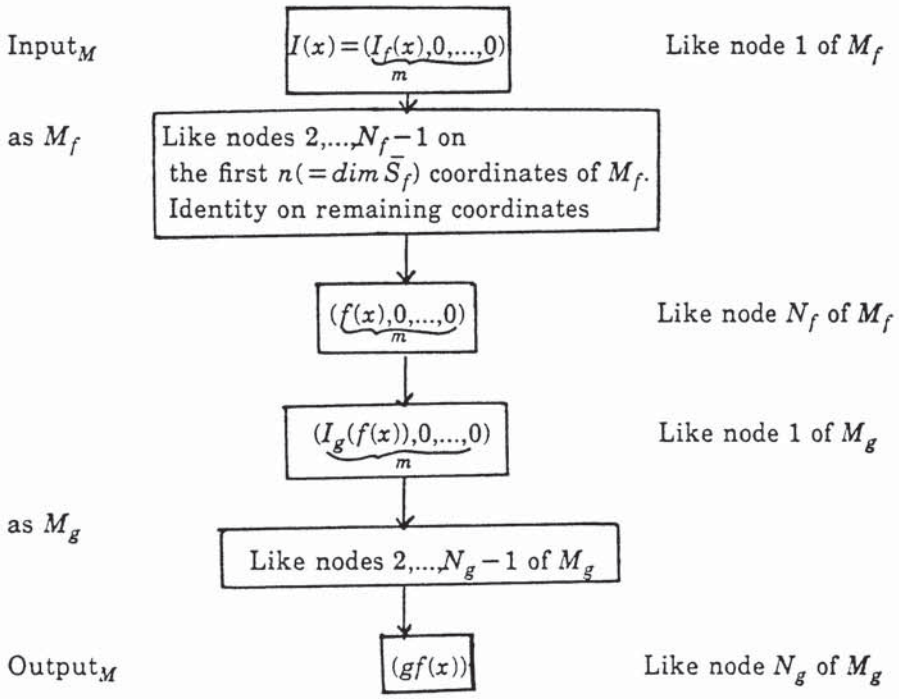

FIGURE 11

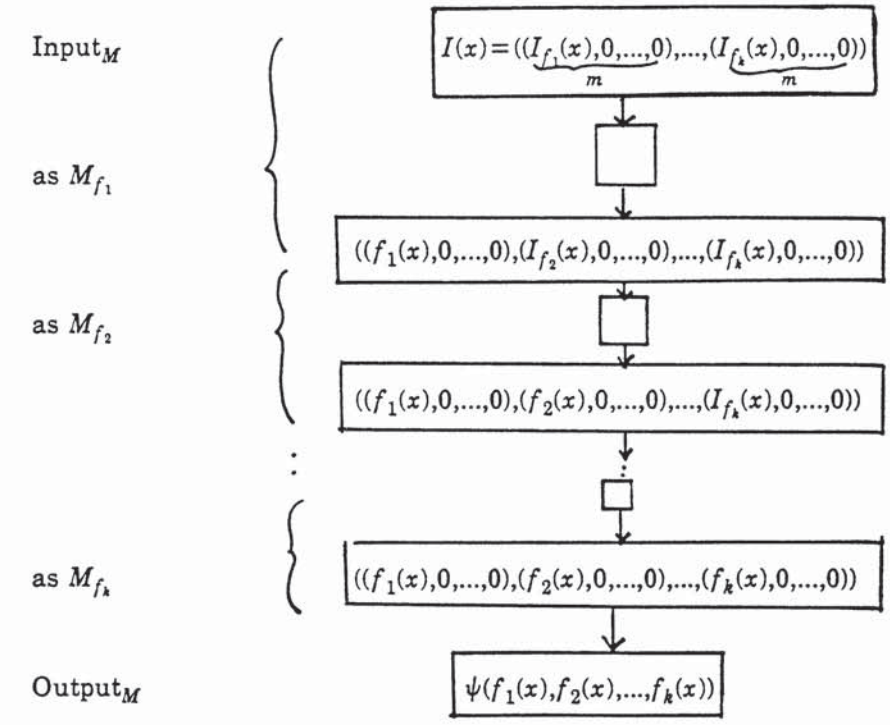

FIGURE 12

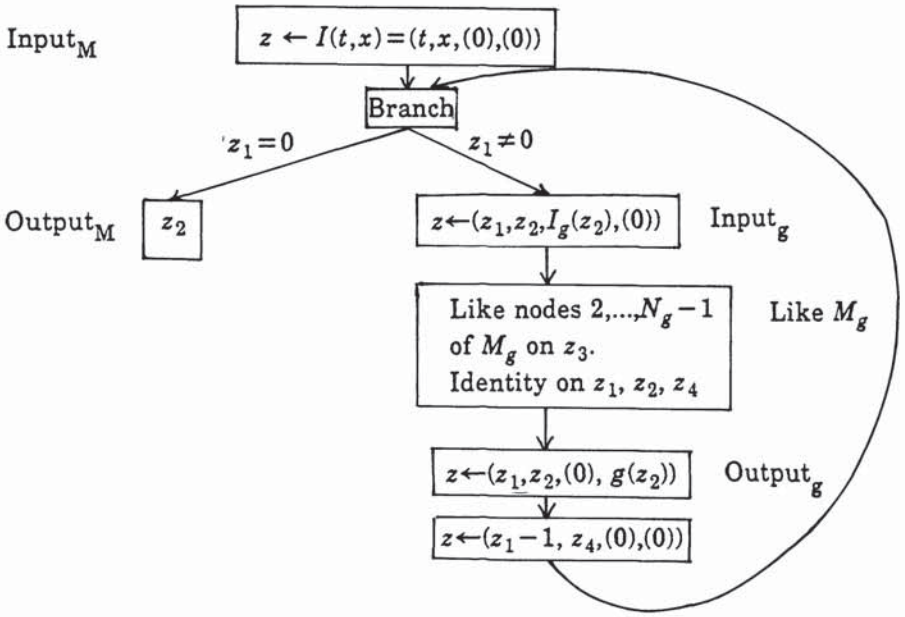

FIGURE 13

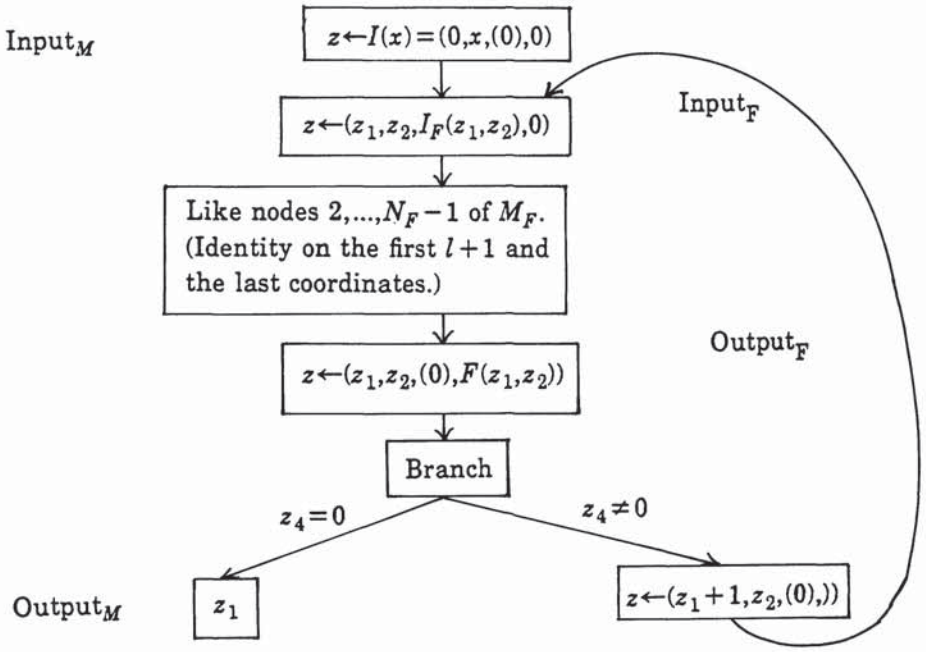

FIGURE 14

Thus we have

THEOREM. $P_R^{<\infty} = C_R^{<\infty}$ *i.e. the finite dimensional partial recursive functions over R are the finite dimensional computable functions over R.*

REMARK. In classical recursive function theory, the partial recursive functions are defined over the natural numbers $\mathbf{Z}^{\geq 0}$, as the smallest class of partial functions $f\colon (\mathbf{Z}^{\geq 0})^l \to (\mathbf{Z}^{\geq 0})^m$ $(l, m < \infty)$ containing the successor, zero and projection functions, and closed under composition, juxtaposition, "classical" primitive recursion (which we define below) and minimalization. It is clear, but hardly ever stated, that this definition extends naturally to the integers to define a class P which we shall call the "classical" partial recursive functions over $\mathbf{Z}$.

PROPOSITION. $P = P_{\mathbf{Z}}^{<\infty}$.

To prove this it is sufficient to define the operation of "classical" primitive recursion (cpr) and show it is equivalent to ours.

DEFINITION. The operation of *classical primitive recursion* (cpr) associates to given (partial) functions $f\colon R^l \to R^m$ and $k\colon R \times R^l \times R^m \to R^m$, a (partial) function $K\colon \mathbf{Z}^{\geq 0} \times R^l \to R^m$ where $K(0, x) = f(x)$ and $K(t+1, x) = k(t, x, K(t, x))$ with $\Omega_K = \{(n, x) \in \mathbf{Z}^{\geq 0} \times R^l | n = 0$ and $x \in \Omega_f$ or $n = t+1$ and $(t, x) \in \Omega_K$ and $(t, x, K(t, x)) \in \Omega_k\}$.

To get the classical definition one replaces R by $\mathbf{Z}^+$ (see e.g. *Manin*). Although cpr is seemingly more general than our definition of primitive recursion, we have

LEMMA. *Both definitions of primitive recursion are equivalent (in the presence of* (i) *and* (a) *and* (b) *in our definition of partial recursive function).*

PROOF. First assume cpr and suppose $g\colon R^l \to R^l$ is a (partial) endomorphism. Let $f\colon R^l \to R^l$ be the identity, and $k\colon R \times R^l \times R^l \to R^l$ be given by $k(t, x, y) = g(y)$. Then (by induction on t) the function K given by cpr is the G stipulated by our definition (c).

Conversely, suppose f and k are given as in the hypothesis of cpr and define (partial) $g\colon R \times R^l \times R^m \to R \times R^l \times R^m$ by $g(t, x, y) = (t+1, x, k(t, x, y))$. Let $G\colon \mathbf{Z}^{\geq 0} \times R \times R^l \times R^m \to R \times R^l \times R^m$ be the (partial) function prescribed by our definition of primitive recursion (c). Then letting $K(t, x) = G(t, 0, x, f(x))]_{R^m}$ we get the function stipulated by cpr. To see this use induction on t. Hence, $K(t, x)$ is gotten by the following composition:

$$(t, x) \underset{\text{juxt.}}{\to} (t, 0, x, f(x)) \underset{G}{\to} G(t, 0, x, f(x)) \underset{\text{proj. on } R^m}{\to} G(t, 0, x, f(x))]_{R^m} = K(t, x).$$

Thus, we have the above Proposition and the following

COROLLARY. $P = \overline{C}_{\mathbf{Z}}^{<\infty}$, *i.e. the finite dimensional computable functions over* $\mathbf{Z}$ *are the "classical" partial recursive functions over* $\mathbf{Z}$.

One can imagine extending the notion of partial recursive functions to the ∞ dimensional case.

8. Existence of a universal machine over a ring. The first task is to describe a machine M over R, as an element $\pi(M)$ of R^∞. Thus "coding of M" is actually a "program" for M, with M assumed in normal form. More precisely, $\pi(M)$ is the sequence of labeled instructions, $n = 1, 2, 3, \ldots, N$, $(n, t_n, \beta_n, b_n, g_n)$, entries having the following meaning. The n at the beginning refers to the node, and t_n is the type of the node. The symbol β_n stands for the next node, and b_n is the lengthof the description of the computation g_n at node n.

More specifics and constraints are: $t_n = 1, 2, 3, 4$ or 5 corresponding to (1) input, (2) computation, (3) branch, (4) output and (5) fifth nodes respectively. Thus $t_n = 1$ if and only if $n = 1$. Also, $t_n = 4$, if and only if $n = N$. If $t_n = 3$, then β_n is the pair $(\beta^-(n), \beta^+(n))$ and if $t_n = 4$, β_n is omitted.

If $t_n \neq 2$, then the pair (b_n, g_n) is omitted. If $t_n = 2$ and

$$g_n = \left(\frac{p_n^1}{q_n^1}, \frac{p_n^2}{q_n^2}, \ldots\right)$$

is of dimension k, we suppose g_n is represented by k followed by the sequence of pairs of polynomials (p_n^l, q_n^l), $l = 1, \ldots, k$ where p_n^l, q_n^l, $l = 1, \ldots, k$ are given by some standard representation (see below for the one we are using here).

At the beginning of $\pi(M)$ we might wish to insert a 0 or 1 to indicate how the various instructions are to be interpreted (0 for finite dimensionally, 1 for infinite dimensionally). However, for simplicity we will assume all machines here are infinite dimensional. This is not a serious restriction since, as can be easily seen, there is a uniform procedure to convert any finite dimensional machine to an equivalent infinite dimensional one.

The following is the flow chart of the Universal Machine U. (See Figure 15.) If it takes as input the pair $(\pi(M), u)$, $u \in \Omega_M$, it will yield as output $\varphi_M(u)$.

Note that enough information is given in $\pi(M)$, so that given a node n, one knows where $(n, t_n, \ldots)$ is in the sequence, and an appropriate subroutine using the fifth node will be able to access that instruction.

The state space $\overline{S}_{\overline{U}}$ of U is presumed decomposed as

$$(\mathbf{Z}^+ + \mathbf{Z}^+) + (R^\infty) + (R + R + R^\infty) + (R^\infty) + (R^\infty).$$

The $\pi(M)$, supposed entered into the fourth of these five components, remains fixed throughout the computation of the universal machine. The fifth component is to provide "work space" needed in some of the subroutines. The second component will hold the "current instruction of M;" the third will be used to simulate the "current state of M."

In the flow chart we are assuming various programming devices in each box that check the well formedness of the current state of U with respect to the operations to be performed.

The universal polynomial evaluator is a subroutine which has input a polynomial (or rational) map g in some standard representation and a state $z \in \overline{S}$. The output is $g(z)$ in $\overline{S}$.

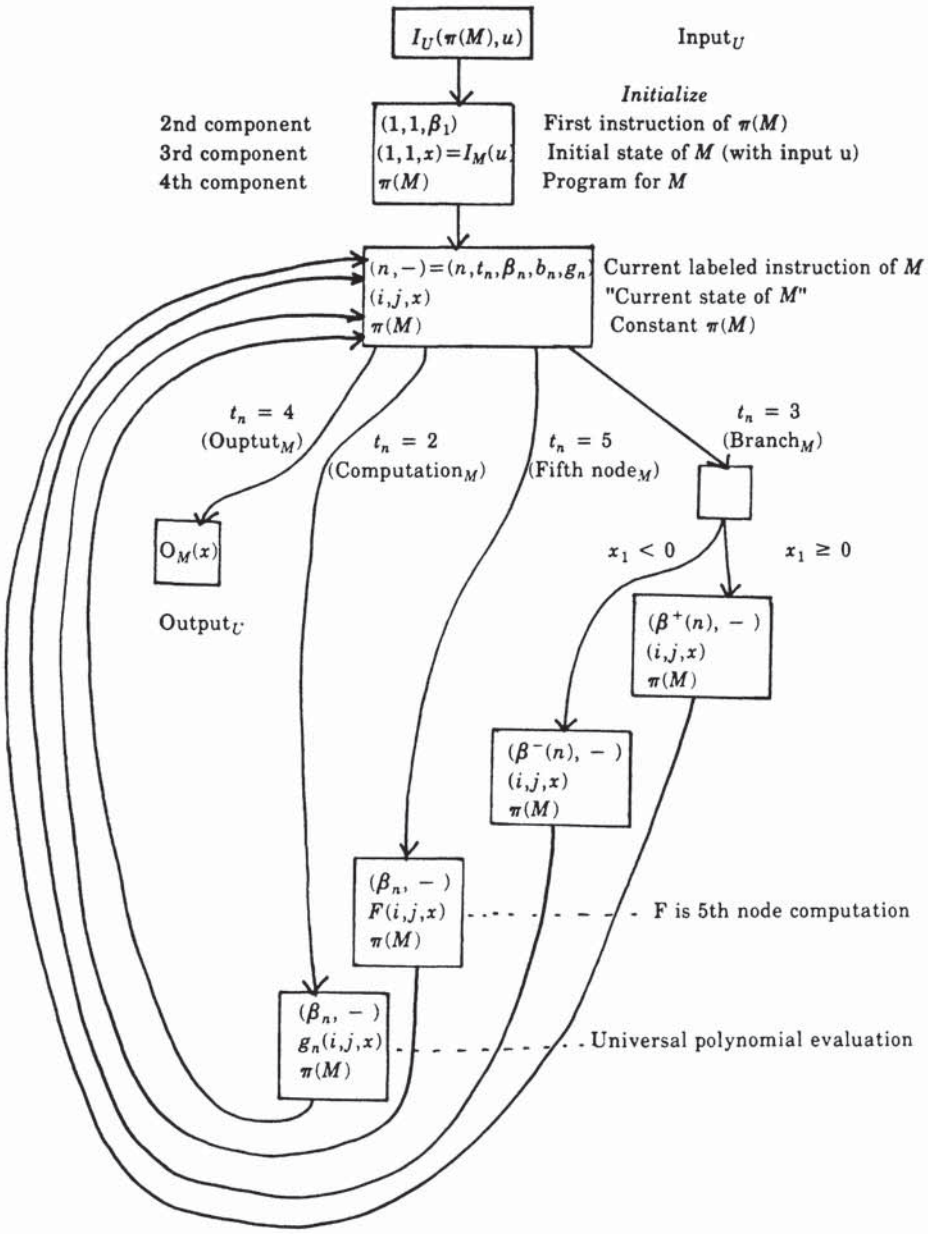

FIGURE 15

The universal polynomial evaluator is described briefly as follows. First we write a routine which sends

$$y = (1, 1, n, x_1, \ldots, x_n, \ldots) \text{ into } y'$$

$$= (1, 1, x_n, x_1, \ldots, x_n, \ldots).$$

Let $$g_1(i, 1, z, \ldots) = (i + 1, 1, z - 1), \ldots)$$

and $g_2(i, 1, z, \ldots) = (1, 1, z, \ldots)$. (See Figure 16.)

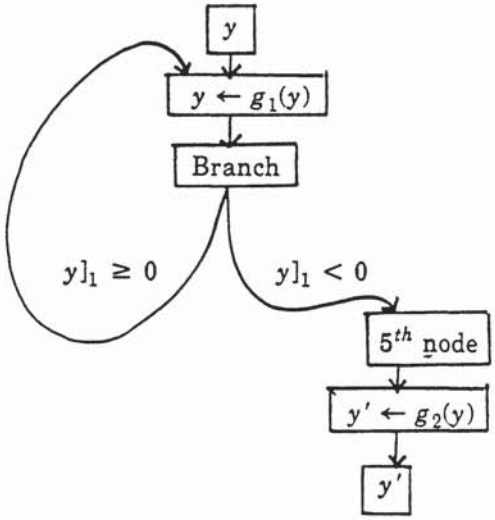

FIGURE 16

A second routine, easily written sends $(\dots, \alpha, t, \dots)$ into $(\dots, t^\alpha, \dots)$ for $\alpha \in \mathbf{Z}^{\geq 0}$. More generally, writing $\alpha = (\alpha_1, \dots, \alpha_k)$, $\alpha_i \in \mathbf{Z}^{\geq 0}$, $t = (t_1, \dots, t_k)$ and $t^\alpha = t_1^{\alpha_1} \cdots t_k^{\alpha_k}$, a third subroutine takes $(\dots, \alpha, t, \dots)$ to $(\dots, t^\alpha, \dots)$. Let a polynomial $f\colon R^k \to R$ of degree d be *standardly represented* as $(k, d, (\alpha, a_\alpha))$ lexicographically in α with $\sum \alpha_i \leq d$ and with $a_\alpha \in R$. Then using the above, one easily constructs

$$(f, t) \to \sum_\alpha a_\alpha t^\alpha.$$

Polynomial maps and rational maps are constructed similarly.

A final remark on the use of the 5th node should be made. In the universal machine, the 5th node is used in

(a) Universal polynomial evaluation,

(b) To access a labeled instruction,

(c) The 5th node computation.

For (a) and (b) we may start (i, j) in $\mathbf{Z}^+ + \mathbf{Z}^+$ with $(1, 1)$. Therefore when the 5th node is used in (a) or (b), the first step is to start with a routine which stores the current (i, j), replace them by $(1, 1)$. Then after (a) or (b) is done, $(1, 1)$ is replaced by the original (i, j). These routines are easily done.

The existence of the universal machine may be used to construct R.E. subsets of R^∞ which are not decidable by the usual Cantor style diagonal arguments—see e.g. *Rogers*. In particular Ω_U is such an R.E. undecidable set.

9. Characterizing R.E. sets as output sets and pseudo-diophantine sets. The R.E. or halting sets over a ring R are the domains of input-output functions of machines over the ring. The output sets are the image sets. For the ring of integers $\mathbf{Z}$, the class of subsets of $\mathbf{Z}$ which are R.E. is the same as the class of output sets.

PROPOSITION 1. *For a real closed field R the output sets are the same as the R.E. sets.*

SKETCH OF PROOF. It is simple to make an R.E. set an output set, first keep track of the input element throughout the computation and output it if the computation halts.

Now we wish to make the output set of a machine M, the halting set of another machine. Let M_1 be the machine which on input T outputs the modified time T halting equations for M, i.e:

(1a‴) $\beta(n_{k-1}, x_{k-1}]_1 u_{i-1}^2) - n_i = 0,$

$x_{k-1}]_1(x_{k-1}]_1 u_{k-1}^2 + 1)(x_{k-1}]_1 u_{k-1}^2 - 1) = 0$

$g(n_{k-1}, x_{k-1}) - x_i = 0$ (for coordinates $j \leq K_T$)

for $k = 1, \ldots, T$.

(1b) $n_0 = 1$, $x_0 = I(y)$ (for coordinates $j \leq K_T$),

(1c) $n_T = N$,

and $w_j = x_T]_{2j}$ for $2j \leq K_T - 2$.

Couple this machine with the Tarski-Seidenberg machine M_2 (see van den Dries) to eliminate all the variables but w from these equations. This produces a system of equations and inequalities in w alone, which w satisfies if and only if w is output by time T. Our machine M_3 on input w at stage T verifies if w satisfies the output of M_2 coupled to M_1 on input T. If w does, M_3 outputs w; if not T is incremented to $T + 1$. A flow chart for this machine is (Figure 17).

PROBLEM 9.1. How generally (i.e., for which rings and fields) does this result hold?

Hilbert's tenth problem on the existence of an algorithm to decide if diophantine equations over $\mathbf{Z}$ have a solution was remarkably solved (*Davis-Matijasevic-Robinson*) by characterizing the R.E. sets of integers as precisely the diophantine sets.

We recall the definition of diophantine set for a commutative ring with identity R. A subset $\Omega \subset R^l$ is *diophantine* over R if there is a $k \geq l$ and

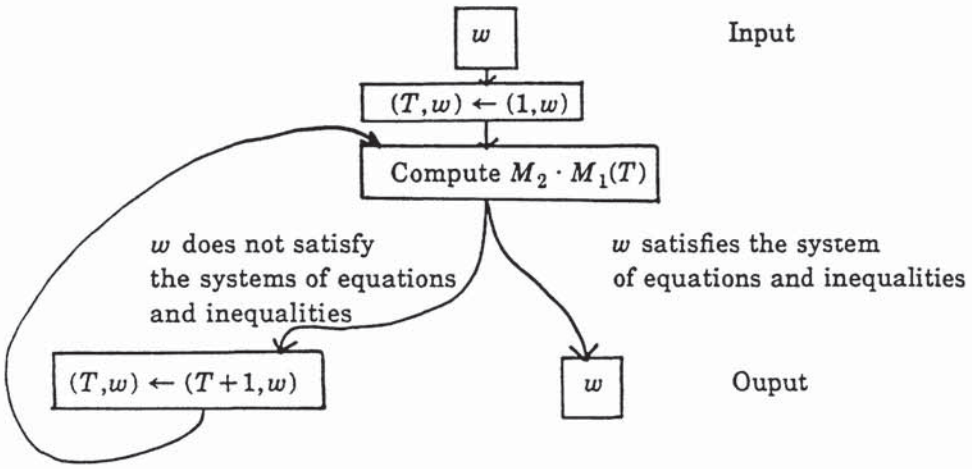

FIGURE 17

a polynomial $P \in R[x_1, \ldots, x_k]$ such that $(x_1, \ldots, x_l) \in \Omega$ iff there exist $x_{l+1}, \ldots, x_k$ in R such that $P(x_1, \ldots, x_l, x_{l+1}, \ldots, x_k) = 0$.

Over $\mathbf{R}$, diophantine sets of reals are R.E., even decidable (by *Tarski*), but not conversely. As we have already seen, there exist R.E. sets of reals that are not decidable over $\mathbf{R}$. But even more, it is rather simple to construct a decidable set over $\mathbf{R}$ with a countable number of connected components, for example $\{x \in \mathbf{R} | \exists n \in \mathbf{Z} \text{ and } |n - x| < \frac{1}{2}\}$. Such sets cannot be diophantine over $\mathbf{R}$, for diophantine sets over $\mathbf{R}$ are semialgebraic and thus can have only finitely many components.

Nevertheless, motivated by *Davis-Putnam* and *Denef* 1, 2, for simple machines we may try to put a diophantine-like structure on the equations which describe the halting sets of machines. Since polynomial rings over any of our rings have many properties similar to the integers, we can hope for some measure of success.

For a ring R and a finite (or countable) set of variables T, subsets $S_1, \ldots, S_k \subset T$ and positive integers $l_1, \ldots, l_k$, we say the set $\Omega \subset R[S_1]^{l_1} \times \cdots \times R[S_k]^{l_k}$ is *pseudo-diophantine* over $R[T]$ iff there exist subsets $S_{k+1}, \ldots, S_m \subset T$, positive integers $l_{k+1}, \ldots, l_m$, and a polynomial P over $R[T]$,

$$P\colon R[S_1]^{l_1} \times \cdots \times R[S_k]^{l_k} \times R[S_{k+1}]^{l_{k+1}} \times \cdots \times R[S_m]^{l_m} \to R[T]$$

such that Ω is the image of the projection on $R[S_1]^{l_1} \times \cdots \times R[S_k]^{l_k}$ of $P^{-1}(0)$. We say a relation is pseudo-diophantine if its graph is.

First we show that composition (or evaluation) is pseudo-diophantine.

PROPOSITION 2. *Let $T = (t_1, \ldots, t_n)$ and $T' = (t'_1, \ldots, t'_m)$. Then the relation $K = F \circ G$ for $K \in R[T']^s$, $F \in R[T]^s$ and $G \in R[T']^n$ is pseudo-diophantine over $R[T \cup T']$. (Here we are thinking of G as a polynomial mapping $G\colon R^m \to R^n$.)*

PROOF. We claim there exists an $s \times n$ matrix A over $R[T \cup T']$ such that

$$(**) \qquad F(T) - K(T') = A(T, T')(T - G(T'))$$
$$\text{if and only if } K(T') = F \cdot G(T').$$

Now for any monomial $a_I t_{i_1}^{n_{i_1}} \cdots t_{i_k}^{n_{i_k}}$, there exist f_i such that $a_I t_{i_1}^{n_{i_1}} \cdots t_{i_k}^{n_{i_k}} - a_I g_{i_1}^{n_{i_1}} \cdots g_{i_k}^{n_{i_k}} = \sum_{j=1}^{k} f_{i_j}(t_{i_j} - g_{i_j})$. This is easily seen by induction on the degree. For degree one it is clear. For degree greater than one we have

$$a_I t_{i_1}^{n_{i_1}} \cdots t_{i_k}^{n_{i_k}} - a_I g_{i_1}^{n_{i_1}} \cdots g_{i_k}^{n_{i_k}} = a_I (t_{i_1}^{n_{i_1}} - g_{i_1}^{n_{i_1}})(g_{i_2}^{n_{i_2}} \cdots g_{i_k}^{n_{i_k}})$$
$$+ a_I t_{i_1}^{n_{i_1}} (t_{i_2}^{n_{i_2}} \cdots t_{i_k}^{n_{i_k}} - g_{i_2}^{n_{i_2}} \cdots g_{i_k}^{n_{i_k}}).$$

Thus there exists an $s \times n$ matrix L over $\mathbf{Z}[T \cup T']$ such that $F(T) - F \cdot G(T') = L(T, T')(T - G(T'))$. On the other hand if $F(T) - K(T') = A(T, T')(T - G(T'))$ add and subtract $F \cdot G(T')$ to obtain

$$F \cdot G(T') - K(T') = (A + L)(T - G(T')).$$

Substituting $G(T')$ for T the right-hand side is identically zero, so $F \cdot G(T') = K(T')$.

Note, the above holds for rings with the property that any polynomial in many variables over R whose values are all zero, is the zero polynomial.

Now we deal with an endomorphism.

PROPOSITION 3. *Let $T = (t_1, \ldots, t_n)$ and $T' = (t'_1, \ldots, t'_n)$ and R be a ring as above. Then the relation $H = F \cdot G$, $H, F \in R[T]^s$ and $G \in R[T]^n$ is pseudo-diophantine over $R[T \cup T']$. (Here we are thinking of G as a polynomial endomorphism $G\colon R^n \to R^n$.)*

PROOF. We claim there exist $K \in R[T']^s$ and $s \times n$ matrices A, B over $R[T \cup T']$ such that

$$(**) \qquad F(T) - K(T') = A(T - G(T')),$$

$$(***) \qquad H(T) - K(T') = B(T - T'),$$

if and only if $H(T) = F \cdot G(T)$. Moreover $K(T') = F \cdot G(T')$. There exist K, A satisfying $(**)$ if and only if $K(T') = F \cdot G(T')$. Now there exist B such that $(***)$ if and only if the values of H and K agree when $T = T'$. Thus $H(T) = F \cdot G(T)$. Similarly $K(T') = F \cdot G(T')$.

Next we will show that dynamics is pseudo-diophantine. First we need two lemmas which we could derive from Denef 1 in a stronger form, but which we prove in our context. For $P \in \mathbf{Z}[t]$ let P' be the derivative of P.

LEMMA 1. *The set $\{(P, P') \in \mathbf{Z}[t] \times \mathbf{Z}[t]\}$ is pseudo-diophantine over $\mathbf{Z}[t, s, t', s']$.*

PROOF. $Q = P'$ if and only if there exist $H, K \in \mathbf{Z}[t, s]$ such that $H(t, s) = P(t) + sQ(t) + s^2K(t, s)$ and $H(t, s) = P(t + s)$. The latter is a pseudo-diophantine relation over $\mathbf{Z}[t, s, t', s']$.

LEMMA 2. *The set $\{(k, t^k)\}_{k \in \mathbf{Z}^{\geq 0}} \subset \mathbf{Z} \times \mathbf{Z}[t]$ is pseudo-diophantine over $\mathbf{Z}[t, s, t', s']$.*

PROOF. Let $g \in \mathbf{Z}[t]$. Then $g = t^k$ if and only if $tg' = kg$ and $g(1) = 1$.

THEOREM. *Let $G \in \mathbf{Z}[T]^n$, $T = (t_1, \ldots, t_n)$ be a polynomial endomorphism $G\colon \mathbf{Z}^n \to \mathbf{Z}^n$ with Zariski dense image (i.e., only the zero polynomial vanishes on it). Then the set $\{(k, G^k)\}_{k \in \mathbf{Z}^{\geq 0}} \subset \mathbf{Z} \times \mathbf{Z}[T]^n$ is pseudo-diophantine over $\mathbf{Z}[S]$ for a finite set $S \supset T$.*

PROOF. Let $T = (t_1, \ldots, t_n)$, $T' = (t'_1, \ldots, t'_n)$, $\hat{T} = (t_1, \ldots, t_n, s)$, $\hat{T}' = (t'_1, \ldots, t'_n, s')$. Let $\hat{G} = G \times \mathrm{id}_s$. Then we claim $G^k = J$ for $J \in R[T]^n$ if and only if there exist $F, H \in R[\hat{T}]^n$, $K \in R[\hat{T}']^n$ and there exist $n \times (n + 1)$ matrices A, B over $R[\hat{T} \cup \hat{T}']$ such that

$$(*) \qquad H(\hat{T}) - F(\hat{T}) = s^k J(T) - T$$

$$(**) \qquad F(\hat{T}) - K(\hat{T}') = A(\hat{T}, \hat{T}')(\hat{T} - \hat{G}(\hat{T}')$$

$$(***) \qquad H(\hat{T}) - K(\hat{T}') = B(\hat{T}, \hat{T}')(\hat{T} - \hat{T}').$$

For suppose F, H, G, J, A, B satisfy $(*)$, $(**)$, $(***)$. Let $F = \sum_{i=1}^{d} f_i(T)s^i$ with $f_d \not\equiv 0$. Then $F \cdot \hat{G} = \sum f_i(G(T))s^i$ and since $H = F \cdot \hat{G}$ by $(**)$

and $(***)$,

$$sH - F = (f_d \cdot G(T))s^{d+1} + \sum_{i=0}^{d-1}(f_i \cdot G(T)) - f_{i+1}(T)s^{i+1} - f_0 = s^k J(T) - T.$$

Since $f_d \cdot G(T) \not\equiv 0$ (the image of G is Zariski dense), $k = d + 1$. $f_0 = T$ and $f_{i+1} = f_i \cdot G$ which proves that $J = G^k(T)$. In fact we have also shown $F = \sum_{i=0}^{k-1} G^i s^i$.

On the other hand, if $J = G^k(x)$. Let

$$F = \sum_{i=0}^{k-1} G^i(T)s^i, \quad H = \sum_{i=0}^{k-1} G^{i+1}(T)s^i.$$

Then $(*)$ is satisfied and there exist A, B, K by Proposition 3.

Now since $\{(k, s^k)\}_{k\in\mathbf{Z}^+} \subset \mathbf{Z} \times \mathbf{Z}[s]$ is pseudo-diophantine, by squaring and adding all the equations we are done.

Note that this Theorem applies to complex analytic iterations. Given $g(z) = z^z + c$, we may consider c as a variable and define $G(c, z) = (c, z^z + c)$.

More generally, consider a finite dimensional machine M with one branch node over a ring R (see Figure 18).

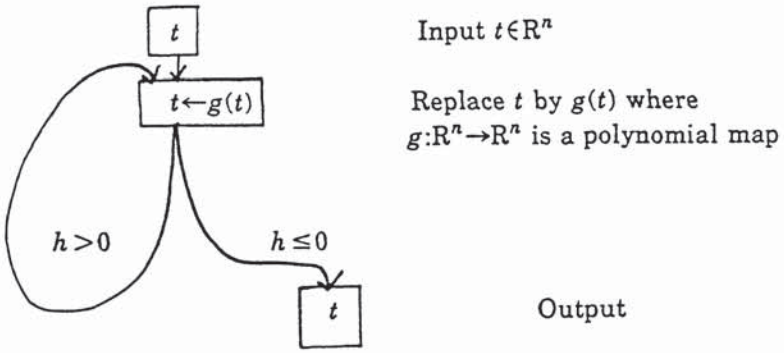

FIGURE 18

where $h\colon R^n \to R$ is given by $h(t) = \sum b_I t^I$ and $g = (g_1, \ldots, g_n)$ is given by $g_i = \sum c_{I,i} t^I$.

For each nonzero coefficient $c_{I,i}$ of g assign a variable $C_{I,i}$; and each nonzero coefficient b_I of h a variable B_I. Suppose there are p new variables $C_{I,i}$ and q new variables B_I.

Let $\tilde{g}_i = \sum C_{I,i} t^I \in \mathbf{Z}[C, T]$ and $\tilde{h} = \sum B_I t^I \in \mathbf{Z}[B, T]$. Here $C = (C_{I,i})$, $T = (t_1, \ldots, t_n)$ and $B = (B_I)$. Define $G \in \mathbf{Z}[C, T] \times \mathbf{Z}[C, T]$, $G\colon R^p \times R^n \to R^p \times R^n$ by $G = I_d \times \tilde{g}$ where $\tilde{g} = (\tilde{g}_1, \ldots \tilde{g}_n)$. Suppose that the image of G is Zariski dense. Then the set $(k, \hat{G}^k)$ is pseudo-diophantine.

The element $t = (t_1, \ldots, t_n)$ is in Ω_M if and only if specializing $C_{I,i}$ to $c_{I,i}$ and B_I to b_I, we have $hG^k(t) \leq 0$ for some k. Thus the set $(k, \tilde{h}, G^k)$

is pseudo-diophantine and to find the halting set of M, we need only specialize the coefficients and evaluate these polynomials.

PROBLEM 9.2. Suppose $\Omega \subset \mathbf{Z}[T, U]$, $T = (t_1, \ldots, t_n)$, $U = (u_1, \ldots, u_m)$ is pseudo-diophantine. For each $u \in R^m$, let

$$X_{\Omega,u} = \{x \in R^n | \exists p \in \Omega \text{ with } p(x, u) \leq 0\}.$$

Then $X_{\Omega,u}$ is R.E. over R. Do all R.E. sets over R arise this way?

10. Most Julia sets are undecidable. In this section we investigate which Julia sets of rational maps $g = p/q$ of the Riemann sphere ($\hat{\mathbf{C}}$) into itself can be R.E. over R.

Recall briefly that the Julia sets $J = J_g$ is the set of points $z \in \hat{\mathbf{C}}$ such that the family of iterates, g^n, $n \geq 0$ of g fails to be normal on any neighborhoods of z. This set is, for degree $g \geq 2$, the same as the closure of the repelling periodic points of g. (See *Brolin* for this and other properties of Julia sets we shall be using.) In particular

(i) J is closed and fully invariant for g, i.e. $g(J) = J = g^{-1}(J)$.

(ii) For any relatively open set $U \subset J$ there is a finite $n \geq 0$ such that $g^n(U) = J$.

(iii) If J has interior it is all of $\hat{\mathbf{C}}$.

The *fixed points of g are hyperbolic* if the derivative of g has modulus different from one for each fixed point, i.e., $|g'(z)| \neq 1$ whenever $g(z) = z$.

THEOREM. *An R.E. Julia set of a rational map* $g: \hat{\mathbf{C}} \to \hat{\mathbf{C}}$ *is either,*

(a) *Empty; and g is a rotation, or a constant; or,*

(b) *a point; and g is fractional linear but not a rotation; or,*

(c) *a real analytic arc; or,*

(d) *a real analytic Jordan curve; or,*

(e) *the whole sphere* $\hat{\mathbf{C}}$.

Moreover, if the fixed points of g are hyperbolic then the arc in (c) *is actually the arc of a round circle, the Jordan curve in* (d) *is a round circle and in this last case g is conjugate by an affine map to a Blaschke product*

$$z \to a_0 z^m \prod_{i=1}^{n} \frac{z - \overline{a}_i}{1 - a_i z} \quad \textit{with} \quad |a_0| = 1 \quad \textit{and} \quad |a_i| \leq 1.$$

PROOF. From (ii) it follows that J is either connected or has uncountably many distinct components. Since J is the countable unions of semialgebraic sets each of which has finitely many components, J must be connected. If J has interior it is the whole sphere. If J is empty or a single point it follows fairly simply that degree $g = 1$ or 0. Rotations are easily seen to have empty Julia set and other fractional linear have one fixed point as their Julia set. Thus we are reduced to the case where J is the countable union of 1-dimensional semialgebraic sets, which we may assume to be closed. Now the Baire category theorem asserts that one of these semialgebraic sets has relative interior in J. Thus we may find a real analytic arc in it, I and a finite $n_0 \geq 0$ such that

$$g^{n_0}(I) = J.$$

$g^{n_0}(I)$ may have as many as two ends and a finite number of branch Y or crossing points X. But if a Julia set has branching or crossing points they must be dense by (ii). Thus J has no branch or crossing points and is either an analytic arc or an analytic Jordan curve.

This finishes the proof of parts (a)–(e) of the theorem. Now we may consider that we are in case (c) or (d) and the fixed points of g are all hyperbolic. The complement of J is then one or two fully invariant simply connected domains. By *Sullivan's* classification theorem these must be the basins of attraction of a hyperbolic fixed point and by *Brolin* (Theorem 9.1 and Lemma 9.1), the arc is the arc of a round circle, the circle is a round circle and the map in this case is conjugate to a Blaschke product.

PROBLEM 10.1. If the Julia set of a rational map of $\hat{\mathbf{C}}$ is a differentiable arc or differentiable Jordan curve is it necessarily the arc of a round circle or a round circle respectively even without the hypothesis that all fixed points are hyperbolic?

EXAMPLES. (a) If g has three attractive fixed points, J_g is not R.E.

(b) If $f(z)$ is a polynomial with at least 3 distinct roots and $N_f(z) = z - f(z)/f'(z)$ then $J_{N_f(z)}$ is not R.E.

PROOF. The basins of attraction permitted by (a)–(e) of the theorem are all of $\hat{\mathbf{C}}$ or 1 or 2 simply connected domains. If g has three attractive fixed points, its basin has at least three distinct components.

$N_f(z)$ is the Newton iteration for the zeros of f, and has an attractive fixed point at each roots of f.

PROBLEM 10.2. Classifying Julia sets as to their complexity is an important problem in complex analytic dynamics (see e.g., *Blanchard*). Can *relative decidability* be called into play? That is, define as in classical recursive function theory, a (Julia) set A to be *decidable in* (Julia) set B if a machine with an additional node for deciding B (i.e., an "oracle") can be used to decide A. Do the resulting equivalence classes and hierarchies shed any light on the Julia set classification problem?

11. Some final remarks and problems. There are a number of ways to further develop and modify the model presented in this paper.

1. For example, it would be of interest to explore an analogous theory for unordered fields such as $\mathbf{C}$, fields of finite characteristic and valued fields (such as the p-adic fields $\mathbf{Q}_p$). A way to do this is to replace the branch nodes by branch nodes which distinguish between $h(z) \neq 0$ and $h(z) = 0$. For the case of valued fields, one could incorporate branching decisions based on the values of the valuation function.

2. We have already indicated how the theory of NP-completeness could be developed for rings in general and have perhaps alluded to ways the NP-completeness of the Feasibility problem might be generalized. We think this direction looks promising.

3. It would also seem natural to further develop ideas from recursive function theory such as fixed point theorems, reducibilities and hierarchies for machines over a ring R.

4. To develop a theory of probabilistic algorithms over R, one would naturally adjoin "coin tossing" nodes.

5. Finally, to bring machines over **R** closer to the subject of numerical analysis, it would be useful to incorporate round-off error, condition numbers and approximate solutions into our development. It would also seem natural to adjoin nodes to compute limits of rapidly converging sequences, as well as other useful functions.

To conclude we indicate how one might start to develop a theory for computing approximate solutions to problems over **R**.

A *problem* over R is a subset $X \subset R^l \times R^m$. Let X_l be the image of the projection of X on R^l and X_m the image of the projection of X on R^m. A machine M with input space R^l and output space R^m *solves the problem* if on input $x_l \in X_l$, M outputs $x_m \in X_m$ and $(x_l, x_m) \in X$. If there exist $c, d \in \mathbf{Z}^+$ such that $T_M(x_l) \leq c(\text{size}\, x_l)^d$, for each "problem instance" $x_l \in X_l$, then M is a *polynomial time* machine or *algorithm* for solving the problem.

We give some examples of classical problems over **R**:

Let P_d be the space of monic complex polynomials of one variable and degree d, $P_d = \{f | f = z^d + a_{d-1}z^{d-1} + \cdots + a_0\} P_d = \mathbf{C}^d \simeq \mathbf{R}^{2d}$. Let $P_\infty = \bigcup_{d \geq 0} P_d$ be the space of all monic complex polynomials. Then

(1) $X_1^d = \{(f, \zeta) | f \in P^d, \zeta \in \mathbf{C} \text{ such that } f(\zeta) = 0\}$ is the problem of finding a root of f, and

(2) $X_{all}^d = \{(f, \zeta) | f \in P_d, \zeta = (\zeta_1, \ldots, \zeta_d) \in \mathbf{C}^d \text{ and } f = \prod_{i=1}^d (z - \zeta_i)\}$ is the problem of finding all roots of f with multiplicity.

No machine over **R** solves X_1^d or X_{all}^d. The output function of a machine is a rational function on each of the semialgebraic sets decomposing the halting set. Since the machine is to halt on all inputs, one of the semialgebraic sets must be open. Even for quadratic equations $(z^2 + a, \zeta)$, ζ is not a rational function of a on any open set of inputs. For if $(Q(a))^2 + a = 0$ for an open set and Q is rational then $Q(t)^2 + t = 0$ as functions, which is not true as can be seen by a short computation.

Thus we are naturally led to a notion of *approximate problem* (or solution). $X_\varepsilon \subset R^l \times R^m$ for $0 < \varepsilon \leq \varepsilon_0$ is an ε *approximation* to X iff $X_{\varepsilon,l} = X_l$ and $(x, y) \in X_\varepsilon$ only if there is an $(x, y_0) \in X$ with $|y - y_0| \leq \varepsilon$.

EXAMPLES.

(1) $X_{1,\varepsilon}^d = \{(f, \overline{\zeta}) | f \in P_d, \overline{\zeta} \in \mathbf{C} \text{ and } \exists \zeta \in \mathbf{C} \text{ such that } f(\zeta) = 0 \text{ and } |\zeta - \overline{\zeta}| \leq \varepsilon\}$, $X_{1,\varepsilon}^\infty = \bigcup X_{1,\varepsilon}^d$.

(2) $X_{all,\varepsilon}^d = \{(f, \overline{\zeta}) | f \in P_d, \overline{\zeta} \in \mathbf{C}^n \text{ and there is a numbering of the roots of } f, \zeta_1, \ldots, \zeta_d \text{ such that } |\zeta_i - \overline{\zeta}_i| \leq \varepsilon \text{ for all } i\}$. $X_{all,\varepsilon}^\infty = \bigcup X_{all,\varepsilon}^d$.

The complexity of a machine to solve a problem should now depend on ε. Since the distance ε scales with change of variables we judge the complexity in terms of size of the coefficients as well.

DEFINITION. A machine M *solves the approximate problem* X_ε *for all* ε, $0 < \varepsilon \leq \varepsilon_0$, in polynomial time iff there exist $c, q \in \mathbf{Z}^+$ such that M with input (x, ε), $x \in X_l$ outputs y with $(x, y) \in X_\varepsilon$ and $T_M(x, \varepsilon) \leq c(\text{size}\, x + \ln|x| + |\ln(\varepsilon)|)^q$ where $|\,|$ is some norm over R.

For P_∞, the size of $f = \sum_{i=0}^d a_i z^i$ is twice its degree and $|f| = \sup |a_i|$. The inputs for such a machine M are $(d, a_0, \ldots, a_{d-1}, 0, \ldots, 0)$ where d

is the degree and the a_i are the coefficients of f given as pairs of real numbers. A recent paper, *Kim*, describes algorithms for the solution of the ε all roots problem, $X_{1,\varepsilon}^{\infty}$. Algorithm 1 there, for example, can be seen to be given by a polynomial time machine.

PROBLEM. What about other classical problems of numerical analysis?

REFERENCES

F. Abramson, *Effective computation over the real numbers*, Proceedings of the 12th Annual (IEEE) Symposium on Switching and Automata Theory, 1971, pp. 33-37.

A. Aho, J. Hopcroft, and J. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Reading, Mass., 1979.

E. Becker, *On the real spectrum of a ring and its application to semialgebraic geometry*, Bull. Amer. Math. Soc. (N.S.) **15** (1986), 19–60.

M. Ben-Or, *Lower bounds for algebraic computation trees*, Proceedings 15th ACM STOC, 1983, pp. 80–86.

P. Blanchard, *Complex analytic dynamics of the Riemann sphere*, Bull. Amer. Math. Soc. (N.S.) **11** (1984), 85–141.

A. Borodin, *Structured vs. general models in computational complexity*, Logic and Algorithmic, Monographie, no. 30, de L'enseignement Mathematique, Geneva, 1982, pp. 47–65.

H. Brolin, *Invariant sets under iteration of rational functions*, Ark. Mat. **6** (1965) 103–144.

J. Canny, *Some algebraic and geometric computations in PSPACE*, 20th Annual ACM Symposium on Theory of Computing, 1988, pp. 460–467.

S. A. Cook, *The complexity of theorem proving procedures*, Proceedings 3rd ACM STOC 1971, pp. 151–158.

N. Cutland, *Computability*, Cambridge Univ. Press, Cambridge, 1980.

M. Davis, *Computability and unsolvability*, Dover, New York, 1982.

M. Davis, Y. Matijasevic, and J. Robinson, *Hilbert's tenth Problem. Diophantine equations: positive aspects of a negative solution*, Mathematical Development Arising from Hilbert Problems, Proc. Sympos. Pure Math., vol. 28, Amer. Math. Soc., Providence, R.I., 1976, pp. 323–378.

M. Davis, and H. Putnam, *Diophantine sets over polynomial rings*. III, J. Math. **7** (1963), 251–256.

J. Denef, *I, Diophantine sets over* $\mathbf{Z}[T]$, Proc. Amer. Math. Soc. **69** (1978), 148–150.

J. Denef, *2, The Diophantine problem for polynomial rings and fields of rational functions*, Trans. Amer. Math. Soc. **242** (1978), 391–399.

B. C. Eaves and U. G. Rothblum, *A theory on extending algorithms for parametric problems*, Department of O.R., Stanford Univ., 1985.

S. Eilenberg, *Automata, languages, and machines*, vol. A, Academic Press, New York, 1974.

E. Engeler, *Algorithmic approximations*, J. Comput. System Sci. **5** (1971), 67–82.

H. Friedman, *Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory*, Logic Colloquium 1969, (R. O. Gandy and Yates, eds.) C.M.E., North-Holland, Amsterdam, 1971, pp. 361–390.

H. Friedman and K. Ko, *Computational complexity of real functions*, J. Theoret. Comput. Sci. **20** (1982), 323–352.

N. Friedman, *Some results on the effects of arithmetic comparisons*, Proceedings of the 13th Annual (IEEE) Symposium on Switching and Automata Theory, 1972, pp. 139–143.

M. Garey and D. Johnson, *Computers and intractability*, Freeman, New York, 1979.

D. Y. Grigorev and N. N. Vorobojov, *Solving systems of polynomial inequalities in subexponential time*, J. Symbolic Computation, **5** nos. 1 and 2 (1988), 37–64.

L. Harrington, M. Morley, A. Seedrov, and S. Simpson (eds.), *Harvey Friedman's Research on the Foundations of Mathematics*, North-Holland, Amsterdam, 1985.

G. T. Herman and S. D. Isard, *Computability over arbitrary fields*, J. London Math. Soc. (2) **2** (1970), 73–79.

H. J. Hoover, *Feasibly constructive analysis*, Ph.D. Thesis, Department of Computer Science, Univ. of Toronto, 1987.

J. P. Jones and Y. V. Matijasevic, *Register machine proof of the theorem on exponential diophantine representation of enumerable sets*, J. Symbolic Logic **49** (1984), 818–829.

M. H. Kim, *Topological complexity of a root finding algorithm*, preprint.

C. Kreitz and K. Weihrauch, *Complexity theory on real numbers and functions*, Lecture Notes in Computer Sci., 145, Theoretical Computer Science, (A. B. Cremers and H. P. Kreigel eds.), Springer-Verlag, Berlin and New York, 1982, pp. 165–174.

L. Lovász, *An algorithmic theory of numbers, graphs and complexity*, CBMS-NSF Series 50, SIAM, Phil., Pa, 1986.

M. Machtey and P. Young, *An introduction to the general theory of algorithms*, North-Holland, New York, 1978.

X. I. Manin, *A course in mathematical logic*, Springer-Verlag, Berlin and New York, 1977; 1984 (2nd printing).

Z. Manna, *Mathematical theory of computation*, McGraw-Hill, New York, 1974.

B. Mazur, *Arithmetic on curves*, Bull. Amer. Math. Soc. (N.S.) **14** (1986), 207–259.

J. Milnor, *On the Betti-numbers of real varieties*, Proc. Amer. Math. Soc. **15** (1964), 275–280.

M. Minsky, *Computation: Finite and infinite machines*, Prentice-Hall, Englewood Cliffs, New Jersey, 1967.

Y. N. Moschovakis, *Foundations of the theory of algorithms*. I, draft 1986.

V. Ya Pan, *Methods of computing values of polynomials*, Russian Math. Surveys **21** no. 1 (1966), 105–136.

M. B. Pour-El and I. Richards, *Computability and noncomputability in classical analysis*, Trans. Amer. Math. Soc. **275** (1983), 539–560.

F. P. Preparata and M. I. Shamos, *Computational geometry*, Springer-Verlag, Berlin and New York, 1985.

M. Rabin, 0.1, *Computable algebra, general theory and theory of computable fields*, Trans. Amer. Math. Soc. **95** (1960), 341–360.

M. Rabin, 0.2, *Proving simultaneous positivity of linear forms*, J. Comput. and System Sci. **6** (1972), 639–650.

J. Renegar, *A faster* PSPACE *algorithm for deciding the existential theory of the reals*, Technical Report No. 792, School of Operations Research and Industrial Engineering, Cornell, April 1988. (Also, Proceedings. of the 29th Annual Symposium on Foundations of Computer Science, 1988, pp. 291–295.)

H. Rogers, Jr., *Theory of recursive functions and effective computability*, McGraw-Hill, New York, 1967.

A. Shamir, *Factoring numbers in* $O(\log n)$ *arithmetic steps*, Inform. Process. Lett. **8** no. 1 (1979), 28–31.

J. C. Shepherdson and H. E. Sturgis, *Computability of recursive functions*, J. Assoc. Comput. Mach. **10** (1963), 217–255.

A. Schönhage, *On the power of random access machines*, Proc. 6th ICALP, Lecture Notes in Computer Science, no. 71, Springer-Verlag, Berlin and New York, 1979, pp. 520–529.

S. Smale, *On the topology of algorithms*. I, J. Complexity **3** (1987), 81–89.

E. Sontag, *Polynomial response maps*, Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin and New York, 1979.

M. Steele and A. Yao, *Lower bounds for algebraic decision trees*, J. Algorithms **3** (1982), 1–8.

V. Strassen, *Algebraische berechnungskomplexität*, Perspectives in Mathematics, (Basel), Birkhäuser-Verlag, 1984.

D. Sullivan, *Quasi-conformal homomorphisms and dynamics*. III, IHES preprint.

A. Tarski, *A decision method for elementary algebra and geometry*, 2nd ed., Berkeley and Los Angeles, 1951, vol. 1, 63 pp.

R. Thom, *Sur l'homologie des variétés algébriques réelles*, Differential and Combinatorial Topology, Princeton Univ. Press, Princeton, New Jersey, 1965, pp. 255–265.

J. Tiuryn, *A survey of the logic of effective definitions*, Lecture Notes in Computer Sci., 125, Logic of Programs, (E. Engeler, ed.), Springer-Verlag, Berlin New York, 1979, pp. 198–245.

J. F. Traub and H. Wozniakowski, *Complexity of linear programming*, Oper. Res. Lett. **1** no. 2, (1982), 59–62.

L. Valiant, *Completeness classes in algebra*, Proc. 11th Ann ACM STOC, 1979, pp. 249–261.

L. van den Dries, *Alfred Tarski's elimination theory for real closed fields*, J. Symbolic Logic **1** (1988), 7–19.

J. von zur Gathen, *Algebraic complexity theory*, Technical Report No. 207/88, Department of Computer Science, University of Toronto, January 1988, 37 pp.

S. Winograd, *Arithmetic complexity of computations*, SIAM Regional Conf. Ser. Appl. Math. **33** (1980), 93pp.

INTERNATIONAL COMPUTER SCIENCE INSTITUTE, 1947 CENTER STREET, BERKELEY, CALIFORNIA 94704

IBM T. J. WATSON RESEARCH CENTER, YORKTOWN HEIGHTS, NEW YORK 10598-0218

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, BERKELEY, CALIFORNIA 94720