Output Sets, Halting Sets and an Arithmetical Hierarchy for Ordered Subrings of the Real Numbers under Blum/Shub/Smale Computation

Rose Saint John

TR-94-035

15 August 1994

Abstract

The original exposition of Blum/Shub/Smale computation for subrings and subfields of real numbers (1989) asks how generally output and halting sets coincide. Aspects of this question were subsequently addressed by Michaux, Byerly, and Friedman/Mansfield. This document synthesizes, simplifies, and extends their answers.

Distinguishing output sets from halting sets in the reals and subrings of the reals leads to a natural arithmetical hierarchy of non-computable sets. Operators analogous to the Jump operator of classical recursion theory are used to build an arithmetical hierarchy from the empty-set. As expected, the classical arithmetical hierarchy for the natural numbers occurs as a special case. Additional special cases arise in other subrings and subfields of the real numbers.

Contents

Chapter 1

Output Sets and Halting Sets

The original exposition for computability over subrings and subfields of real numbers given by Blum, Shub, Smale [BSS] notes that the family of output sets of computable functions coincides with the family of halting sets of computable functions for rings $\mathcal{Z}$, the integers, and $\mathcal{R}$, the real numbers. Problem BSS 9.1 [BSS] asks how generally this result holds. Subsequently Michaux [Mi], Byerly [By], and Friedman/Mansfield [FM] addressed and answered aspects of this question. This chapter synthesizes, simplifies, and augments their contributions to the solution of BSS Problem 9.1.

Rings and fields, in this chapter, are considered to be (sub)rings/fields of the real numbers, $\mathcal{R}$. These are archimedian and the set $\mathcal{N} = \{0, 1, 2, 3, \ldots\}$ is decidable. For sets, the words decidable, recursive, and computable are synonymous. These are sets for which some BSS machine, M, halts with answer 1 for input value in the set and halts with answer 0 for input value outside the set.

1.1 Characterizing Output Sets and Halting Sets

Reference [BSS] describes BSS computation.

Definition 1.1.0.1 *Subset H of L^n, for ring L, is a* <u>*halting set*</u> *if there is a BSS machine M with input values $\vec{x} = (x_1, x_2, \ldots x_n) \in L^n$ with the property that: M halts on $\vec{x}$ if and only if $\vec{x} \epsilon H$. The input set is considered to be all of L^n.*

A halting set is the domain of a (partial) computable function.

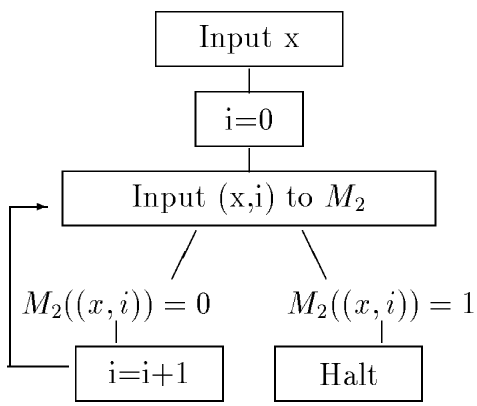
Definition 1.1.0.2 *For ring L, subset O of L^n is an* <u>*output set*</u> *if O is the range of a (partial) L-computable function.*

Definition 1.1.0.3 *An n-ary relation $S(\vec{x})$ with variables $\vec{x} = x_1, \ldots, x_n$ is* <u>*recursive*</u> *if there is a BSS machine M which takes input values $\vec{l} \epsilon L^n$ and computes answer 1 when $S(\vec{l})$ holds and answer 0 when $S(\vec{l})$ fails.*

In classical recursion theory on the natural numbers, $\mathcal{N}$, each recursive enumerable set corresponds to a first-order formula with one existential quantifier, which formula defines the set. In BSS computation, a halting set may or may not be definable by a formula in ordinary first-order language. Nonetheless, an analogous correspondence occurs.

Theorem 1.1.0.1 *For ring L, where $\mathcal{Z} \subseteq L \subseteq \mathcal{R}$, halting sets are exactly those which can be expressed as $\{x \mid \exists n \epsilon \mathcal{N}\ S(x,n)\}$ for recursive relation $S(x,n)$.*

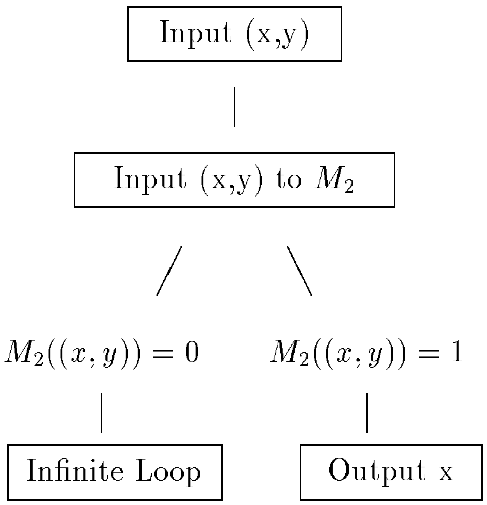
Proof: Set $A = \{x \epsilon L \mid \exists n\ \ S(x,n)\}$ is the halting set for BSS machine M, below, where M_2 is a BSS machine that decides recursive relation S.



This proves the assertion in one direction. For the other way, let H be the halting set for a BSS machine B. H=$\{x \mid \exists n$ [Machine B with input x halts after fewer than n steps]$\}$. The square brackets enclose a recursive relation on (x,n). □

Theorem 1.1.0.2 *For ring L, where $\mathcal{Z} \subseteq L \subseteq \mathcal{R}$, output sets are exactly those which can be expressed as $\{x \mid \exists y S(x,y)\}$ for recursive relation $S(x,y)$.*

Proof: Set A=$\{x \epsilon L \mid \exists y\ \ S(x,y)\}$ is the output set of machine M, below, where M_2 decides relation S.



This proves the assertion in one direction. For the other way, let O be the output set of BSS machine B. Then O=$\{x \mid \exists z \exists n$ [Machine B with input z halts in fewer than n steps and outputs x.]$\}$. Write $\vec{y} = (z, w) \epsilon L^2$ and let $p_1(\vec{y})$ be the first coordinate of $\vec{y}$ and $p_2(\vec{y})$ be the second. Unary recursive relation N distinguishes the natural numbers in L. Then O=$\{x \mid \exists \vec{y} \epsilon L^2$ [N$(p_2(\vec{y}))$ $\wedge$ B with input $p_1(\vec{y})$ halts in fewer than $p_2(\vec{y})$ steps and outputs value x]$\}$. The square brackets enclose a recursive relation on $(x, \vec{y})$. □

Theorem 1.1.0.3 *Every L-halting set is an L-output set.*

Proof: $\{x \mid \exists n S(x,n)\}$ is the set $\{x \mid \exists y (N(y) \wedge S(n,y))\}$ for unary recursive relation $N(y)$, which distinguishes the natural numbers.

Theorem 1.1.0.4 *Output sets are characterized exactly as projections of halting sets.*

Proof: Output set $\{x \mid \exists y \;\; S(x,y)\}$ is the projection of a recursive set, and recursive sets are halting sets. To prove the assertion in the other direction, consider the projection P, of a halting set H=$\{(x,z) \mid \exists n \;\; S(x,z,n)\}$. For P=$\{x \mid \exists z \exists n \;\; S(x,z,n)\}$, write $\vec{y} = (z,w) \epsilon L^2$, and then P=$\{x \mid \exists \vec{y} \;\; [N(p_2(\vec{y})) \; \wedge \; S(x, p_1(\vec{y}), p_2(\vec{y})]\}$. The square brackets enclose a recursive relation on $(x, \vec{y})$, and so by Theorem 1.1.02, P is an output set. □

1.2 Examples

Example 1.2.0.1 *Each BSS machine M is shown in [BSS] to have a code $\lceil M \rceil$ which is a finite sequence of members of the ring L. The function that M computes is denoted $\varphi_{\lceil M \rceil}$, a downward arrow $\downarrow$ denotes convergence (halting) and $\downarrow_n$ means the computation halts in n computational steps. Blum et al show the halting set $K_0 = \{\lceil M \rceil \mid \exists n \;\; \varphi_{\lceil M \rceil}(\lceil M \rceil) \downarrow_n\}$ is undecidable.*

Example 1.2.0.2 *Polynomials of degree n in m variables can be thought of as fixed-length tuples of coefficients in a ring/field L.* $P_{m,n} = \{\vec{l} \mid \exists\vec{x}\ \vec{l}(\vec{x}) = 0\}$ *is the output set consisting of such polynomials with a root in* L^m. *For the field of reals,* $P_{m,n}$ *is both a halting set and a recursive set. For the ring of integers,* $P_{m,n}$ *is a halting set that fails to be recursive* ($m > 1$). *In some fields (e.g. the field appearing in the next example) the output sets* $P_{m,n}$ *may fail to be halting sets.*

Example 1.2.0.3 *There are subsets of prime natural numbers which are not recursive enumerable in the classical thoery. Let A be such a set. Form the smallest field L containing* $\mathcal{Q} \cup \{\sqrt{p} \mid p\epsilon A\}$. *The square primes in L form an output set* $O=\{x\epsilon L \mid \exists y\ [y^2 = x \wedge N(x) \wedge\ x \text{ is prime}]\}$. *This output set fails to be a halting set. If, to the contrary, there were a BSS machine M, with machine constants* $l_0, l_1, \ldots, l_n$ *in L that takes members of L as input and halts on the square primes, O, a conversion of M produces a BSS machine over the ring* $\mathcal{Z}$ *that halts on the same set of integers as does M. The conversion is described below. Due to the conversion, the set O=A, with respect to* $\mathcal{Z}$, *is seen to be a recursive enumerable set of natural numbers, violating the original choice of A.*

Here is the idea of the conversion. Machine M is taken to have irrational constants $\sqrt{p_1}$, $\sqrt{p_s}$, $\ldots, \sqrt{p_j}$. *Form all possible products to obtain* $\{c_i\}_{i=1,2,\ldots,m}$. *For example,* c_j *might be* $\sqrt{p_1p_2}$ *or* $\sqrt{p_2p_3p_5}$. *Arithmetic performed by machine M is considered to produce numbers of the form* $\sum_{i=0}^{m} a_ic_i$ *where* a_i *are rational and* c_0*=1. Machine M is converted to do arithmetic and testing on the tuples* $\{a_i\}_{i=0,1,\ldots,m}$ *of coefficients. Converting the arithmetic and testing for equality are straightforward, assuming that computation and test nodes accept and release tuples* $\{a_i\}$ *of rationals. To test "*$0 < \sum_{i=0}^{m} a_ic_i$ *?" for some non-zero* $\{a_i\}$, *successive rational under-approximations and over-approximations of* $\sum_{i=0}^{m} a_ic_i$ *are taken until an under-approximation exceeds 0 or an over-approximation is less than 0.*

After this conversion, the new machine takes integer input, uses constants $p_1, p_2, \ldots, p_j$ *and halts on the square primes, namely the set A. A second conversion replaces the rational numbers by integers, resulting is a BSS machine with integer constants and halting set A, which contradiction proves that for the field* $\mathcal{Q}[\{\sqrt{p}\}]_{p\epsilon A}$, *output and halting sets do not coincide.*

1.3 First Order Definability

Reference [BSS] shows that the halting set of machine M is defined by a countable disjunction of quantifier-free first order formulas $\alpha_j(c_1, \ldots, c_n, x)$ where $c_1, \ldots, c_n$ are the machine constants for M and $\alpha_j(\vec{c}, x)$ holds whenever machine M halts on input x in j time steps.

Theorem 1.3.0.5 *Every halting set can be expresssed as* $\{x \mid \bigvee_{i\epsilon\mathcal{N}} \xi_{f(i)}(\vec{c}, x)\}$ *where* f *is computable and* $\{\xi_n\}_{\ n\epsilon\mathcal{N}}$ *is some fixed, effective listing of the quantifier-free formulas in an appropriate number of variables.*

Proof: This restates the previous paragraph. □

Theorem 1.3.0.6 *Every output set can be expressed as* $\{y \mid \exists x \bigvee_{i\epsilon\mathcal{N}} \xi_{f(i)}(x,y)\}$ *for computable* f.

Proof: An output set has be characterized as $\{y \mid \exists x\ R(x,y)\}$ for recursive R. The previous theorem applied to halting set $\{(x,y) \mid R(x,y)\}$ gives the result. □

Theorem 1.3.0.7 *Sets definable by quantifier-free first order formulas are exactly the time-bounded halting sets.*

Proof: Time bounded halting sets are those for which the above described countable disjunction $\bigvee_{j\epsilon\mathcal{N}} \alpha_j(\vec{c},x)$ is a finite disjunction, and thus a single quantifier-free formula. Conversely, given a single quantifier free formula, some BSS machine determines within a fixed number of timesteps whether or not a specified value satisfies the formula. □

1.4 Coincidence of Output Sets and Halting Sets

The notation O=H means output and halting sets coincide. Uniform O=H is distinguished from O=H in this section. Hovever, Theorem 2.2.2.2 in Chapter 2 shows conditions O=H and uniform O=H are equivalent.

Definition 1.4.0.4 *Uniform O=H holds when O=H and there is a computable function* f *without constants, which for output set* O *of machine* M*, computes index* $\lceil M_2 \rceil = f(\lceil M \rceil)$ *where machine* M_2 *halts on* O.

Note that the proof of $H \subseteq O$ actually shows that $H \subseteq O$ uniformly. Condition O=H (uniform O=H) for a ring or a field can be characterized in several useful ways.

1.4.1 Useful Characterizations

Theorem 1.4.1.1 *For ring* L*, O=H if and only if* L *admits quantifier simplification for* Σ_1 *sets.*

Proof: Quantifier simplification for Σ_1 sets means that every set of the form $\{x \mid \exists y R(x,y)\}$ is also a set of the form $\{x \mid \exists n \epsilon N\ \ S(x,n)\}$ for recursive relations R and S. This theorem follows from Theorems 1.1.0.1 and 1.1.0.2.□

Theorem 1.4.1.2 *For ring* L*, O=H if and only if every set of the form* $\{x \mid \exists y R(x,y)\}$ *for recursive relation* R*, is also a set of the form* $\{x \mid \bigvee_{j\epsilon\mathcal{N}} \xi_{f(j)}(\vec{c},x)\}$ *where* f *is computable and* $\{\xi_n\}_{n\epsilon\mathcal{N}}$ *is some fixed, effective listing of quantifier-free formulas in an appropriate number of variables.*

Proof: If O=H, Theorems 1.1.0.2 and 1.3.0.5 yield the desired expression. The converse holds because a set of the form $\{x \mid \bigvee_{i\epsilon\mathcal{N}} \xi_{f(i)}(\vec{c},x)\}$ with the given conditions on f and $\{\xi_n\}_{n\epsilon N}$ is indeed a halting set. □

Michaux[Mi] remarks that weak quantifier elimination is related to the coincidence of output and halting sets.

Definition 1.4.1.1 *A ring or field L enjoys* <u>*effective weak quantifier elimination*</u> *if formula* $\exists y\, \xi(x,y)$ *for quantifier-free* $\xi(x,y)$ *is equivalent in L to a countable disjunction* $\bigvee_{j\epsilon\mathcal{N}} \xi_{f(j)}(\vec{c},x)$ *of quantifier-free formulas where* f *is L-computable and* $\{\xi_n\}_n\epsilon\mathcal{N}$ *is an effective listing of first-order quantifier-free formulas in the appropriate number of variables (corresponding to* $\vec{c},x$*) for the theory of L.*

Definition 1.4.1.2 *A ring or field L enjoys* <u>*uniform weak quantifier elimination*</u> *if there is a computable function* g *for which* $\exists y\, \xi(\vec{c},x,y) \equiv_L \bigvee_{j\epsilon\mathcal{N}} \xi_{g(\lceil\xi\rceil,j)}(\vec{c},x)$ *where* $\{\xi_n\}_{n\epsilon\mathcal{N}}$ *is an effective listing of quantifier-free formulas in constants* $\vec{c}$ *and one free variable.*

Theorem 1.4.1.3 *A ring or field L for which O=H enjoys effective weak quantifier elimination.*

Proof: For formula $\exists y\, \xi(\vec{c},x,y)$, output set $O = \{x \mid \exists y\, \xi(\vec{c},x,y)\}$ is a halting set for some machine. Thus $O = \{x \mid \bigvee_{j\epsilon\mathcal{N}} \alpha_j(\vec{d},x)\}$ for this machine with constants $\vec{d}$ where $\alpha_j(\vec{d},x)$ states that the machine halts on input x in j timesteps. □

Theorem 1.4.1.4 *For ring L, O=H uniformly if and only if the first order theory of L admits uniform weak quantifier elimination.*

Proof: For formula $\exists y\, \xi(\vec{c},x,y)$, where ξ is quantifier-free, output set $O = \{x \mid \exists y\, \xi(\vec{c},x,y)\ \}$ for machine M is the halting set for machine $M_{f(\lceil M\rceil)}$. Then $O = \{x \mid \bigvee_{j\epsilon\mathcal{N}} \alpha_j(\vec{c},x)\}$ where $\vec{c}$ are constants of both machines M and $M_{f(\lceil M\rceil)}$ and $\alpha_j(\vec{c},x)$ states that machine $M_{f(\lceil M\rceil)}$ halts in j timesteps for input x. Thus uniform weak quantifier elimination holds. Conversely, suppose uniform weak quantifier elimination holds and O is an output set for some machine M with constants $\vec{c}$. Output set $O = \{y \mid \exists x \exists j\ \alpha_j(\vec{c},x,y)\}$ where $\alpha_j(\vec{c},x,y)$ states that machine M halts on input x with answer y in j timesteps. Thus $O = \{y \mid \bigvee_{j\epsilon\mathcal{N}} \exists x\, \alpha_j(\vec{c},x,y)\}$ and by uniform weak quantifier elimination, $O = \{y \mid \bigvee_{j\epsilon\mathcal{N}} \bigvee_{i\epsilon\mathcal{N}} \xi_{g(\lceil\alpha_j\rceil,i)}(\vec{c},y)\}$ which is a halting set. The index of a machine which halts on O is computable from the index of M, the original machine with output set O. viz. indices $\lceil\alpha_j\rceil$ are computable, g is computable and the listing $\{\xi_n\}$ is fixed and effective. □

Corollary 1 *For a ring or a field,*
uniform weak quantifier elimination $\iff$
$O = H$ *uniformly* $\iff$
$O = H \implies$
effective weak quantifier elimination.

Proof: Theorem 2.2.2.2 in Chapter 2 completes this proof. □

Corollary 2 *If L is a real closed field then O=H uniformly.*

Proof: Since the first order theory of a real closed field uniformly admits quantifier elimination, it admits, a fortiori, uniform weak quantifier elimination and so $O = H$ uniformly. □

Theorem 1.4.1.5 *(Michaux)For ring L, O=H if and only if projections of halting sets are halting sets.*

Proof: Since output sets are characterized in Section 1.1 as projections of halting sets, this is immediate. □

1.4.2 Rings That Can Be Listed

Michaux [Mi] notes that rings which are finitely generated over $\mathcal{Q}$ or $\mathcal{Z}$ enjoy a recursive enumeration which is a stronger property than O=H.

Definition 1.4.2.1 *A* <u>*listing*</u> *for ring L is an L-computable function f with domain $\mathcal{N}$, the natural numbers and with range all of L.*

Every member of L appears of the list {f(1), f(2), f(3),...}.

Theorem 1.4.2.1 *If ring L can be listed, then O=H for L.*

Proof: An L-output set $\{x \mid \exists y R(x,y)\}$ is halting set$\{x \mid \exists n R(x, f(n))\}$. □

Theorem 1.4.2.2 *Rings of real numbers that can be listed are exactly those rings which are finitely generated over $\mathcal{Q}$ or $\mathcal{Z}$.*

Proof: Let L be listed by f. A BSS machine M computes f and M has machine constants $\{c_1, c_2, ...c_m\}$. Each member of L is $f(k)$ for some $k\epsilon\mathcal{N}$ and so if formed by ring (field) operations on $\{k, c_1, c_2, ..., c_m\}$. Thus L is generated by $\{1, c_1, c_2, ..., c_m\}$.

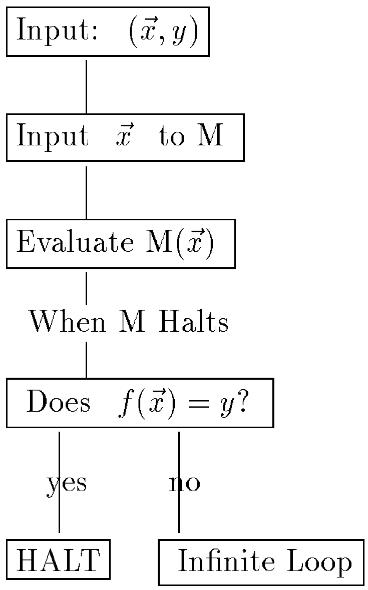
If L is a ring or field finitely generated by $\{1, a_1, a_2, ..., a_n\}$, then a BSS machine can be defined, using those same constants, for a listing of L. □

1.4.3 Function Graphs as Halting Sets

A well-known theorem for computable functions over $\mathcal{N}$ states that a partial function is computable if and only if its graph is recursive enumerable (a halting set). This theorem holds in exactly those rings which can be listed.

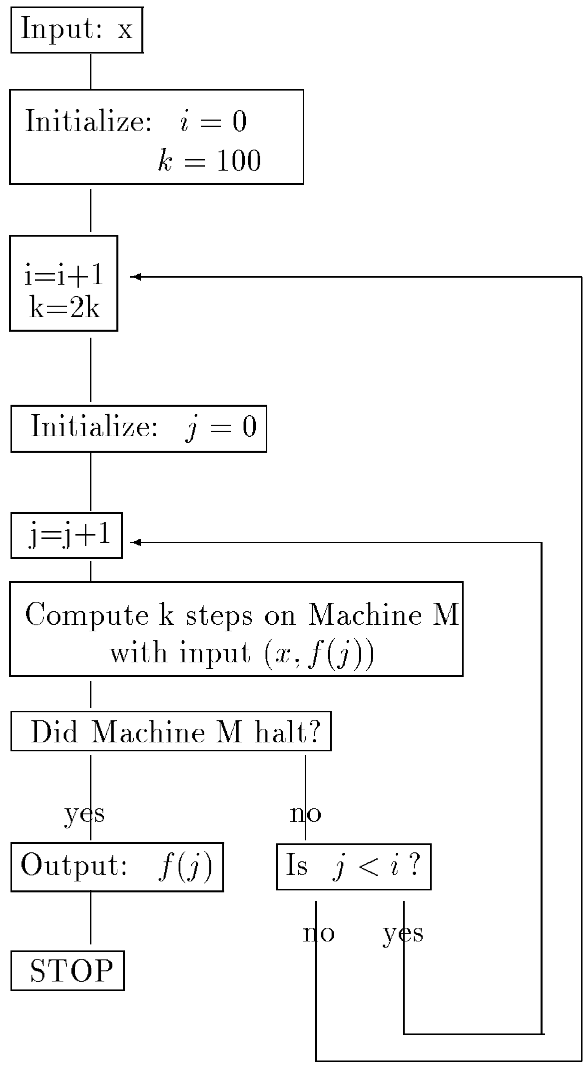
Theorem 1.4.3.1 *The graph G of a partial recursive function $f(x)$ is a relation in L which is a halting set.*

Proof: Partial recursive function f is computed by machine M. Here is the machine that halts on G, the graph of f.



Theorem 1.4.3.2 *For a ring L of reals that can be listed, the graph G of partial function* φ *is a halting set if and only if* φ *is computable.*

Proof: Let G be the graph of a funciton φ and the halting set of machine M. Let f be a listing for L. Here is the BSS machine that computes φ. The computation works because f maps $\mathcal{N}$ onto L.



Theorem 1.4.3.3 *For ring $L \subseteq \mathcal{R}$, suppose graph G for partial function f is a halting set if and only if f is computable. Then ring L is finitely generated over $\mathcal{Q}$ or $\mathcal{Z}$.*

Proof: The first part of the proof shows that the algebraic subring of L is finitely generated. Function $f(\vec{x}, m, n) = z$, where $\vec{x}$ are integer or rational coefficients of an m-degree polynomial for which z is the n^{th} real root when (all) real roots are listed in increasing order, has graph G which is a halting set in L. By hypothesis, f is a (partial) computable function in L. Some BSS machine M with constants $\{c_i\}_{i=1,2,...k}$ computes f. Then every algebraic element of L belongs to the ring (or field) generated by $\{1, c_1, c_2, ...c_k\}$.

The second part of the proof shows that L is of finite transcendence degree. Suppose to the contrary that L is of infinite transcendence degree. Let function g compute the positive square root of members of L. By hypothesis, g is computable by some machine M with finitely many machine constants. Choose transcendental t, independent of the machine constants, and use M to compute $g(t^2)$. This gives t as an algebraic expression in t^2 and the machine constants of M, which violates the choice t, proving that the transcendence degree of L is finite. □

1.4.4 Rings That Can be Coded in $\mathcal{N}$

Byerly [By] investigated the O=H question in rings of reals whose members can be recursively named with well-behaved names in $\mathcal{N}$.

Definition 1.4.4.1 *A coding for an ordered ring/field L is an L-computable ring/field isomorphism of L onto a subset of $\mathcal{N}$ with L-recursive ring/field operations and ordering $\oplus, \odot, \lhd$ in $\mathcal{N}$.*

Definition 1.4.4.2 *The image of a member l of L under coding π is called the name of l.*

Theorem 1.4.4.1 *Rings that can be listed can be coded.*

Proof: Let f be a listing for ring L. Define $\pi(l)$ to be the least k such that f(k)=l. Map π is L-computable, well-defined, and one-to-one. The following definitions of $\oplus, \odot$, and $\lhd$ in $\pi(L) \subset \mathcal{N}$ guarantee that π is an isomorphism.
$m \oplus n \equiv^{def} \pi(f(m) \oplus f(n))$
$m \odot n \equiv^{def} \pi(f(m).f(n))$
$m \lhd n \; iff \; \pi(m) < \pi(n)$ in L. □

A subring (field) L of $\mathcal{R}$, finitely transcendental over $\mathcal{Q}$ enjoys a "canonical" coding η, which is defined by first defining η on $\hat{L}$, the real closure of L, and then restricting η to L. Each member of $\hat{L}$ is a root of a polynomial $p = \sum_{i=0}^{m} c_i x^i$ where each c_i is generated using ring (field) operations from $\{1, \beta_1, \ldots, \beta_j\}$ where the $\{\beta_i\}_{i \leq j}$ are fixed, independent transcendentals. Each c_i is coded by $\lceil c_i \rceil$, a natural number; polynomial p is coded as its sequence of coefficients; and $\eta(\hat{l})$ is the smallest natural number $\lceil m, \lceil p \rceil \rceil$ where $\eta(\hat{l})$ is considered to be the m th real root of polynomial p, when its real roots are listed in increasing order. Notation $\lceil \; \rceil$ represents some canonical way of coding a sequence of natural numbers into one natural number.

This coding η is computable using a BSS machine with constants $\{1,\beta_1, \ldots, \beta_j\}$ and $\oplus, \odot, \lhd$ are defined in $\mathcal{N}$ so as to be L-computable. The restriction of η to L is a coding for L.

When subring L of $\mathcal{R}$ can be coded, and machine M computes π, the coding, then M can be converted to machine $\pi(M)$, with input $\pi(l)$ for $l \epsilon L$, and machine constants $\pi(c_1), \ldots, \pi(c_n)$, where $c_1, \ldots, c_n$ are machine constants for M. Arithmetic and testing by $\pi(M)$ are done using $\oplus, \odot$, and $\lhd$. Then $\pi(l) = k$ if and only if $\varphi_{\pi(M)}(\pi(l)) = \pi(k)$.

Theorem 1.4.4.2 *Subrings of the reals, $\mathcal{R}$ that can be coded are exactly those subrings of finite transcendence degree over $\mathcal{Q}$.*

Proof: The canonical coding η has been described for subrings (fields) finitely transcendental over $\mathcal{Q}$. Conversely, let L be a subring (field) of $\mathcal{R}$ coded by π. Since π is computable, BSS machine M with machine constants $c_1, \ldots, c_n$ computes π. If L were infinitely transcendental over $\mathcal{Q}$, choose β algebraically independent of $c_1, \ldots, c_n$. When β is input to M, natural number $\pi(\beta) =$ k is computed, contradicting the choice of β, which contradiction proves that L is finitely transcendental over $\mathcal{Q}$. □

Since a coding π for ring L is L-recursive, image $\pi(L)$ is an L-output set. This image may or may not be an L-halting set. If it is an L-halting set it may or may not be L-recursive.

Theorem 1.4.4.3 *(Byerly) If a subring L of real numbers is coded by π then image $\pi(L)$ is an L- halting set in $\mathcal{N}$ if and only if O=H in L.*

Proof: When image $\pi(L)$ is a halting set, a BSS machine M with constants in L halts on those natural numbers which are in $\pi(L)$. Consider any output set O, the output set of some BSS machine M_2. $O = \{y \mid \exists x \, \varphi_{\lceil M_2 \rceil}(x) = y\} = \{y \mid \exists n \, \varphi_{\lceil \pi(M_2) \rceil}(n) = \pi(y) \wedge n \epsilon \pi(L)\} = \{y \mid \exists n \exists m [\varphi_{\lceil \pi(M_2) \rceil}(n) = \pi(y) \wedge \varphi_{\lceil M \rceil}(n) \downarrow_m]\}$. Thus every output set is a halting set, using the hypothesis that $\pi(L)$ is a halting set. Conversely, if all output sets are halting sets, then output set $\pi(L)$ is a halting set. □

Theorem 1.4.4.4 *The real closure $\hat{L}$ of codable ring (field) L has L-recursive image $\eta(\hat{L})$, where η is the previously described canonical coding.*

Proof: Given natural number k, here is a procedure for deciding whether or not k$\epsilon\eta(\hat{L})$. Decode k into a root number and coefficient codes. Using transcendence base $\{\beta_1, \ldots, \beta_m\}$, reconstruct the polynomial and see if it has a sufficient number of real roots. If so, check all natural numbers smaller than k to confirm that k is the least one that codes the root being considered. This procedure is done by a BSS machine with constants in L. □

Theorem 1.4.4.5 *For canonical coding η of field L into $\mathcal{N}$, image $\eta(L)$ is L-recursive if and only if L is an $\hat{L}$-recursive subset of its real closure $\hat{L}$.*

Proof: If BSS machine M decides whether or not $n \epsilon \eta(L)$ for any $n \epsilon \mathcal{N}$, a machine that computes $\varphi_{\lceil M \rceil}(\eta(\hat{l}))$ can be used to determine whether or not $\hat{l} \epsilon L$. Conversely, let M be an $\hat{L}$-BSS machine that decides L$\subseteq \hat{L}$, and M_2 be a BSS machine with constants in L that decides $\eta(\hat{L}) \subseteq \mathcal{N}$ according to Theorem 1.4.4.4. A machine M_3 when given $k \epsilon \mathcal{N}$ evaluates both $\varphi_{\lceil M_2 \rceil}(k)$ and $\varphi_{\lceil \eta(M) \rceil}(k)$ can determine whether or not $k \epsilon \eta(L)$. □

1.4.5 Infinite Transcendence Degree over $\mathcal{Q}$

The condition O=H is especially strong in rings of real numbers of infinite transcendence degree over $\mathcal{Q}$.

Theorem 1.4.5.1 *(Michaux) A subring R of real numbers of infinite transcendence degree over $\mathcal{Q}$ for which O=H, is a field.*

Proof: Consider output sets: $U = \{x \mid \exists y \;\; x \cdot y = 1\}$ and $D = \{(x, y) \mid x \, divides \, y\}$. If U were to contain an open interval I, then since every non-zero $l \epsilon R$ has a rational multiple $r \cdot l \;\; \epsilon I \subseteq U$, there is some y for which $(r \cdot l)y = 1$, which means that l is a unit. U, a halting set, will contain an open set I if any $\beta \epsilon R$,transcendental over $\mathcal{Q}[l_1, l_2, \ldots, l_n]$, belongs to U. The $l_1.l_2, \ldots, l_n$ are machine constants for a BSS machine which halts on U. Showing R is a field is accomplished by exhibiting $\beta \epsilon R$, transcendental over $\mathcal{Q}[l_1, l_2, \ldots, l_n]$. Choose β_1, β_2 transcendental over $\mathcal{Q}[l_{n+1}, \ldots, l_m]$ where $l_{n+1}, \ldots l_m$ are machine constants for a BSS machine halting on D. Since $(\beta_1, \beta_1 \cdot \beta_2) \epsilon D$ there are open intervals I_1, I_2 with $\beta_1 \epsilon I_1$, $\beta_1 \cdot \beta_2 \epsilon I_2$, such that $I_1 \mathrm{x} I_2 \subseteq D$. Choose rational $r \epsilon I_2$and $\beta \epsilon I_1$ transcendental over $\mathcal{Q}[l_1, \ldots, l_n]$ so that $\beta \mid r$.Thus for some $y \epsilon R$, $\beta \cdot y = r$; and since r is a unit, $\beta \epsilon U$. Whence R is a field. □

Theorem 1.4.5.2 *A field L of real numbers of infinite transcendence degree over $\mathcal{Q}$ is real closed if and only if L is an $\hat{L}$-halting set.*

Proof: For $L = \hat{L}$ the result is clear. For L r.e. in $\hat{L}$ let machine M with machine constants $l_1 \ldots, l_n$ halt on L. Choose some $\beta \epsilon L$ transcendental over $\mathcal{Q}[l_1 \ldots, l_n]$ which belongs to an interval $I \subseteq \hat{L}$ consisting of points of L. Since L and $\hat{L}$ agree on an interval of $\hat{L}$, $L = \hat{L}$. □

Theorem 1.4.5.3 *A real field L of infinite transcendence degree over $\mathcal{Q}$, for which O=H , is closed under radicals.*

Proof: Consider output sets $H_k = \{x \mid \exists y \;\; y^k = x\}$ where $k = 1, 2, 3, \ldots$. For fixed k_0 ,let $c_1, c_2, \ldots, c_n$ be machine constants for a BSS machine halting on H_{k_0}. Choose β transcendental over $\mathcal{Q}[c_1, c_2, \ldots, c_n]$ and since $\beta^{k_0} \epsilon H_{k_0}$,some open interval I belongs to H_{k0}. Every $l \epsilon L$ has a multiple $r^{k_0} \cdot l$ which belongs to I, for rational r. Hence, for some y, $r^{k0} \cdot l = y^{k0}$. Thus l is a k_0th power and a member of H_{k0}. □

Definition 1.4.5.1 *A monic polynomial p in one variable with coefficients in L, a subfield of real numbers, is a* <u>*good polynomial*</u> *if any root of p occurring in $\hat{L}$ already occurs in L.*

The set of good polynomials G_n of degree n is an L-output set. Members of G_n are considered to be n-tuples of coefficients. Thus $G_n \subseteq L^n$.

Theorem 1.4.5.4 *For a real field L of infinite transcendence degree over $\mathcal{Q}$, if O=H then L^n with the product interval topology has an open set of good polynomials of degree n. This holds for all n.*

Proof: Since G_n is an output and a halting set, it contains an open set if it contains one n-tuple with entries transcendental over $\mathcal{Q}[l_1, \ldots, l_m]$ where $l_1 \ldots l_m$ are machine constants for a machine which halts on G_n. Choose $\beta_1, \beta_2, \ldots, \beta_n$ transcendental over $\mathcal{Q}$ $[l_1, \ldots, l_m]$ and let $c_1, c_2, \ldots c_n$ be coefficients of monic polynomial $(x-\beta_1)(x-\beta_2)\ldots(x-\beta_n)$. This gives $(c_1 \ldots c_n)\epsilon G_n$ where entries $c_1 \ldots c_n$ are appropriately transcendental, proving the existence of an open set in L^n of good, degree n, polynomials. □

Michaux[Mi] essentially uses the algebraic fact that if a real field has one bad (i.e. not good) polynomial, then, for some n, the bad polynomials of degree n are dense in L^n. This fact, along with the preceding theorem characterize those real fields of infinite transcendence degree over $\mathcal{Q}$ for which O=H.

Theorem 1.4.5.5 *(Michaux) A subring L of $\mathcal{R}$ of infinite transcendence degree over $\mathcal{Q}$ has property O=H if and only if L is a real closed field.*

A detailed proof is given in reference [Mi].

1.4.6 Summary For Rings and Fields in $\mathcal{R}$

Analysis of three distinct families of subrings of real numbers has given three distinct answers to the question 'When does O=H ?'. The technique of computable listing applies for recursion theoretic results to rings finitely generated over $\mathcal{Z}$ or $\mathcal{Q}$. The technique of computable coding yields recursion theoretic results for rings of finite transcendence degree over $\mathcal{Q}$ but not finitely generated over $\mathcal{Q}$. Rings infinitely transcendental over $\mathcal{Q}$ are amenable to powerful techniques, algebraic, topological, and computational.

Chapter 2

Hierarchy of Non-Computable Sets

Classical recursion theory over the natural numbers $\mathcal{N}$ or integers $\mathcal{Z}$ ranks undecidable sets in an "arithmetical" hierarchy. First order formulas are associated with degrees in this computability hierarchy. An associated formula has a specific quantifier configuration which corresponds to definability of the sets of that degree in the first order language.

Similarly, for BSS computability theory over rings and fields, a natural hierarchy of undecidable sets occurs with the classical arithmetical hierarchy as a special case. Undecidable sets are associated with, and ranked according to quantifier configuration. Dissimilarly however, these undecidable sets are generally not definable in ordinary first-order language. Other differences occur.

Rings and fields in this chapter, usually denoted L, are (sub)rings/fields of the real numbers $\mathcal{R}$. The archimedian property is heavily exploited as $\mathcal{N}$={1,2,3,.....} recursive in L.

2.1 Relative Computability

Concepts of relative computability and reducibility for BSS computation agree exactly with their traditional counterparts.

2.1.1 Reducibility Via a Computable Function

Definition 2.1.1.1 *For sets $A \subseteq L^n$ and $D \subseteq L^m$, set A is* $\underline{many-one\ reducible}$ *to D (written $A \leq_m D$) if there is a BSS machine M such that $\vec{l} \epsilon A$ if and only if $\varphi_{\lceil M \rceil}(\vec{l}) \epsilon D$.*

Definition 2.1.1.2 *Sets A,D, as above, are* $\underline{m-equivalent}$ *(written $\equiv_m$) if both $A \leq_m D$ and $D \leq_m A$.*

Relation $\equiv_m$ is an equivalence relation.

Theorem 2.1.1.1 *For* $A \leq_m B$

(i) If B is recursive then A is recursive.

(ii) If B is a halting set then so is A.

(iii) If B is an output set then so is A.

(iv) If B is both Σ_1 *and* Π_1 *then so is A.*

Proof: Assume A≤B via computable function f, so that $x \epsilon A \iff f(x) \epsilon B$.
(i) B recursive implies A recursive.
(ii) B={x| $\exists n\ S(x,n)$} implies A={x| $\exists n\ S(f(x),n)$}.
(iii) B={x| $\exists y\ S(x,y)$} implies A={x| $\exists y\ S(f(x),y)$}.
(iv) B={x| $\exists y S(x,y)$} = {x| $\forall y R(x,y)$} implies A={x| $\exists y S(f(x),y)$} = {x| $\forall y R(f(x),y)$}.
□

Definition 2.1.1.3 *A halting set C is* <u>*m − complete*</u> *if* $H \leq_m C$ *for every L-halting set H. An output set C is* <u>*m − complete*</u> *if* $O \leq_m C$ *for every L-output set O. A* Π_1 *set C*={$\vec{x} \mid \forall \vec{y}\ S(\vec{x},\vec{y})$}*, where S is a recursive relation in L, is* <u>*m − complete*</u> *if* $P \leq_m C$ *for each L-*Π_1 *set P.*

2.1.2 Reducibility Via an Oracle

A generic BSS machine (see [BSS]) has input/output nodes, computation/test and branch nodes, and load/store (fifth) nodes. This configuration can be modified by introducing a finite number of A-oracle nodes, where $A \subseteq L^m$ for some $m \leq \omega$. An A-oracle node responds yes or no to the question "Is this value a member of A?". Notation $\varphi^A_{\lceil M \rceil}$ stands for the function computed by BSS machine M with A-oracle node(s), indexed by $\lceil M \rceil$.

Definition 2.1.2.1 *For sets* $D \subseteq L^n$ *and* $A \subseteq L^m$ *set D is* <u>*Turing reducible*</u> *to set A, written* $D \leq_T A$, *if there is a BSS machine M with A-oracle node(s) such that*

$$\varphi^A_{\lceil M \rceil}(\vec{x}) = \begin{cases} 1 & for\ \vec{x} \epsilon D \\ 0 & for\ \vec{x} \notin D \end{cases}$$

That is, $\varphi^A_{\lceil M \rceil}$ *is the characteristic function for D.*

As expected, m-reducibility implies Turing reducibility, but not conversely.

Definition 2.1.2.2 *Set D is* <u>*recursive enumerable in A*</u> *if D is the halting set of a machine M with A-oracle node(s).*

Definition 2.1.2.3 *Set D is an* <u>*output set in A*</u> *if D is the output set of machine M with A-oracle node(s).*

2.2 Non-Computable Sets

2.2.1 Index Sets and Rice's Theorem

The s-m-n Theorem and Rice's Theorem hold for BSS computability and perform their traditional role in proving undecidability for most index sets. The canonical non-recursive halting set, K_0, described first in [BSS] and also in Example 1.2.01 of Chapter 1, is used in this and ensueing sections.

Theorem 2.2.1.1 (Kleene s-m-n Theorem) *For every $m, n \geq 1$ there exists a BSS machine M_n^m such that for BSS machine B and $l_1, l_2, \ldots, l_m, x_1, x_2, \ldots, x_n \epsilon L$* $\varphi_{\lceil M_n^m(\lceil B\rceil, l_1, l_2, \ldots, l_m)\rceil}(x_1, x_2, \ldots, x_n) = \varphi_{\lceil B\rceil}(l_1, l_2, \ldots, l_m, x_1, x_2, \ldots, x_n)$.

Proof: For BSS machine B with arguments $\vec{l}\epsilon L^m$ and $\vec{x}\epsilon L^n$, machine $M_n^m(\lceil B\rceil, \vec{l})$ with input $\vec{x}$ evaluates B at $(\vec{l}, \vec{x})$. This is effective in $\lceil B\rceil$ and $\vec{l}$ and so M_n^m can be formally described as a BSS machine. □

Definition 2.2.1.1 *A set A of sequences in L is an <u>index set</u> if every sequence in A encodes a BSS machine and* $\forall \vec{x}, \vec{y}\ [\vec{x}\epsilon A \wedge\ \varphi_{\vec{x}} = \varphi_{\vec{y}}] \implies [\vec{y}\epsilon A]$.

Theorem 2.2.1.2 *If A is a non-trivial index set then either $K_0 \leq_m A$ or $K_0 \leq_m \tilde{A}$ (read A complement).*

Proof: Let M_0 be a machine that fails to halt on each input. Suppose that $\lceil M\rceil\epsilon A$. The proof works just as well if $\lceil M\rceil\epsilon\tilde{A}$. Let B be a machine with index in $\tilde{A}$. The s-m-n Theorem gives machine M($\vec{x}$) such that

$$\varphi_{\lceil M(\vec{x})\rceil}(y) = \begin{cases} \varphi_{\lceil B\rceil}(\mathrm{y}) & if\ \vec{x}\epsilon K_0 \\ undefined & otherwise \end{cases}$$

Then $[\vec{x}\epsilon K_0] \implies [\varphi_{\lceil M(x)\rceil} = \varphi_{\lceil B\rceil}] \implies [\lceil M(x)\rceil\epsilon\tilde{A}]$
and $[\vec{x} \notin K_0] \implies [\varphi_{\lceil M(x)\rceil} = \varphi_{\lceil M_0\rceil}] \implies [\lceil M(x)\rceil\ \epsilon A]$. □

Theorem 2.2.1.3 (Rice) *For any set A of BSS machine indices, A is an L-recursive set of sequences in L if and only if $A = \phi$ or A has each BSS machine index as a member.*

Proof: This follows from the previous theorem and the undecidability of K_0. □
Rice's Theorem provides examples of L-undecidable sets.

2.2.2 Output Sets and Halting Sets

Although the following three index sets are m-equivalent in classical recursion theory for $\mathcal{N}$, this is not so for BSS computability over rings and fields.

$K_0 = \{\lceil M\rceil \mid\ \varphi_{\lceil M\rceil}(\lceil M\rceil)\downarrow\}$

$K \; = \{(\ulcorner M \urcorner, \vec{x}) \mid \; \varphi_{\ulcorner M \urcorner}(\vec{x}) \downarrow\}$

$K_1 = \{\ulcorner M \urcorner \mid \; \exists x \; \varphi_{\ulcorner M \urcorner}(x) \downarrow\}$

Theorem 2.2.2.1 *In a subring L of the real numbers,*

(i) K_0 *is m-complete for halting sets;*

(ii) K_1 *is m-complete for output sets;*

(iii) $K_0 \equiv_m K$*;*

(iv) $K_0 \leq_m K_1$*;*

(v) When O=H, $K_0 \equiv_m K_1$;

(vi) When O≠H, $K_0 <_m K_1$.

Proof:
(i) For arbitrary halting set H={x| $\exists n \, S(x,n)$} the s-m-n Theorem gives computable function f where

$$\varphi_{f(x)}(y) = \begin{cases} 1 & if \;\; S(s,n) \;\; holds \;\; for \;\; some \;\; n \\ undefined & otherwise. \end{cases}$$

Then, $x \not\in H \implies \varphi_{f(x)} \equiv_m 1 \implies \varphi_{f(x)}(f(x)) \downarrow \implies f(x) \epsilon K_0$. Likewise, $x \not\in H \implies \varphi_{f(x)}(f(x)) \uparrow \implies f(x) \not\in K_0$.
(ii) For arbitrary output set O={x| $\exists y \, S(x,y)$}, the s-m-n Theorem gives computable f where

$$\varphi_{f(x)}(y) = \begin{cases} 1 & if \;\; S(x,y) \;\; holds \\ undefined & otherwise. \end{cases}$$

Then, $x \epsilon O \implies \exists y \, \varphi_{f(x)}(y) = 1 \implies f(x) \epsilon K_1$.
Also, $x \not\in O \implies \varphi_{f(x)} \equiv_m \uparrow \implies f(x) \not\in K_1$.
(iii) Since K is a halting set, from (i), conclude $K \leq_m K_0$. The s-m-n Theorem gives computable function f where

$$\varphi_{f(x)}(y) = \begin{cases} 1 & if \;\; \varphi_x(x) \downarrow \\ undefined & otherwise. \end{cases}$$

For g(x)=(f(x),f(x)), $x \epsilon K_0 \iff g(x) \epsilon K$ which proves that $K_0 \leq_m K$.
(iv) For computable f given in (iii) above $x \epsilon K_0 \iff f(x) \epsilon K_1$, proving that $K_0 \leq_m K_1$.
(v) If K_1 is a halting set, which occurs whenever O=H, by (i) $K_1 \leq_m K_0$ and by (iv), $K_0 \equiv_m K_1$.
(vi) The contrapositive is proved. Suppose $K_1 \leq_m K_0$. By (ii), for any output set,O, $O \leq_m K_1 \leq_m K_0$ and by Theorem 2.1.1.1 part (ii), O is a halting set since K_0 is. □

Theorem 2.2.2.2 *In a subring L of real numbers, O=H is equivalent to uniform O=H.*

Proof: Suppose that O=H and that A is the output set for BSS machine B. That is, A=$\{x \mid \exists y\ B(y) = x\}$. Set K_1 is m-complete for output sets and is itself an output set. Let M be a machine that halts on K_1. As in the proof of (ii) in Theorem 2.2.2.1, consider computable function f where $x \epsilon A \iff f(x) \epsilon K_1 \iff$ M halts on $f(x)$ and $f(x)$ is the index of a machine which takes y as input, evaluates $B(y)$ and outputs 1 if $B(y) = x$ and is otherwise undefined. Thus the value $f(x)$ is computed as $g(x, \lceil B \rceil)$ for computable function g. Machine M_3 takes input x, computes $g(x, \lceil B \rceil)$ and inputs this value to machine M. Machine M_3 has halting set A and its index is (uniformly) effective in $\lceil B \rceil$. □

2.3 Jump and Skip Operators

Jump and skip operators in BSS computation are analogous to the jump operator in classical recursion theory.

2.3.1 Basics of Jumps and Skips

Definition 2.3.1.1 *For L, a subring of the real numbers, $\mathcal{Z} \subseteq L \subseteq \mathcal{R}$ and subset $A \subseteq L^m$, $m \leq \omega$, A-Jump $=\{\lceil M \rceil \mid \exists n \epsilon \mathcal{N}\ \varphi^A_{\lceil M \rceil}(\lceil M \rceil) \downarrow_n\}$, and A-Skip $= \{\lceil M \rceil \mid \exists y \epsilon L, n \epsilon \mathcal{N}\ \varphi^A_{\lceil M \rceil}(y) \downarrow_n\}$.*

Recall the definitions of sets K_0 and K_1 in section 2.2.2, and note that K_0=$\emptyset$-Jump and K_1=$\emptyset$-Skip. Just as K_0 and K_1 are, respectively, a canonical halting set and output set for a BSS machine without an oracle, so A-Jump and A-Skip are a canonical halting set and output set, respectively, for BSS machines with oracle A.

Theorem 2.3.1.1 *For BSS machine M, with oracle A, halting set H and output set O,*
(i) $H \leq_m$ A-Jump.
(ii) $O \leq_m$ A-Skip.

Proof of (i): The s-m-n Theorem, relativized for oracle A, gives computable function f so that

$$\varphi^A_{f(x)}(y) = \begin{cases} 1 & if\ M\ halts\ on\ x \\ undefined & otherwise \end{cases}$$

Then $x \epsilon H \implies \varphi^A_{f(x)}(y) \equiv_m 1 \implies f(x) \epsilon$ A-Jump.
Likewise $x \not\epsilon H \implies \varphi^A_{f(x)}(y) \equiv_m \uparrow \implies f(x) \not\epsilon$ A-Jump.
Proof of (ii): The s-m-n Theorem, relativized for oracle A, gives computable function f so that

$$\varphi^A_{f(x)}(y) = \begin{cases} 1 & if\ M\ halts\ on\ input\ y\ and\ \varphi_{\lceil M \rceil}(y) = x. \\ undefined & otherwise. \end{cases}$$

Then $x \epsilon O \implies$ there is some y_0 such that $\varphi_{\lceil M \rceil}(y_0) = x \implies \varphi^A_{f(x)}(y_0)\downarrow \implies f(x) \epsilon$A-Skip.
Likewise, $x \not\epsilon O \implies \varphi^A_{f(x)}(y) \equiv_m \uparrow \implies f(x) \not\epsilon$A-Skip. □

Theorem 2.3.1.2 *For set A, $A <_m$ A-Jump $\leq_m$ A-Skip and $\tilde{A} <_m$ A-Jump $\leq_m$ A-Skip.*

Proof: The usual diagonal argument is relativized to produce a halting set in A which is not A-decidable, for the first inequality. For the second, the (relativized) s-m-n Theorem produces recursive function f where

$$\varphi^A_{f(x)}(y) = \begin{cases} 1 & if\ \varphi^A_x(x)\downarrow \\ undefined & otherwise \end{cases}$$

which gives $[x \epsilon$A-Jump$] \iff$ [f(x)ϵ A-Skip]. Also, the (relativized) s-m-n Theorem gives recursive function g where

$$\varphi^A_{g(x)}(z) = \begin{cases} 1 \text{ if } & x \notin A \\ undefined & otherwise. \end{cases}$$

Then, $x \epsilon \tilde{A} \iff g(x) \epsilon$ Jump A. □

Theorem 2.3.1.3 *For $A \leq_m B$, A-Jump $\leq_m$ B-Jump.*

Lemma: For $A \leq_m B$ via computable function f, there is a computable function g such that for all y, $\varphi^A_x(y)\downarrow \iff \varphi^B_{g(x)}(y)\downarrow$.
Proof of lemma: At each oracle node of BSS machine with index x, consult oracle B regarding f(z), in lieu of consulting oracle A regarding z. Let g(x) be the index of the modified machine. For all z, oracle B gives the answer for f(z) that oracle A gives for z, so that φ^A_x and $\varphi^B_{g(x)}$ compute the same (partial) function. □
Proof of theorem: The (relativized) s-m-n theorem gives computable function h where

$$\varphi^B_{h(x)}(y) = \begin{cases} 1 & if\ \varphi^B_{g(x)}(x)\downarrow \\ undefined & otherwise \end{cases}$$

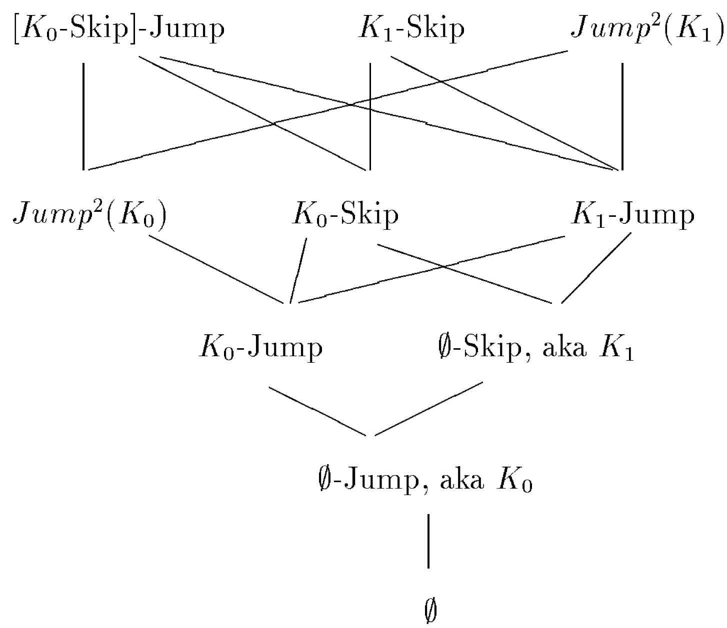
Computable functions g and f are described in the above lemma. Then,
$[x \epsilon$A-Jump$] \implies [\varphi^A_x(x)\downarrow] \implies [\varphi^B_{g(x)}(x)\downarrow] \implies [\varphi^B_{h(x)} \equiv 1] \implies [h(x) \epsilon$B-Jump$]$.
Also, $[x \notin$A-Jump$] \implies [\varphi^A_x(x)\uparrow] \implies [\varphi^B_{g(x)}(x)\uparrow] \implies [\varphi^B_{h(x)} \equiv \uparrow]$
$\implies$[h(x)$\notin$B-Jump]. □

Theorem 2.3.1.4 *For $A \leq_m B$, A-Skip $\leq_m$B-Skip.*

Proof: [xϵA-Skip] $\implies [\varphi^A_x(y_0)\downarrow] \implies [\varphi^B_{g(x)}(y_0)\downarrow] \implies [g(x) \epsilon$ B-Skip].
Also, [x$\notin$A-Skip] $\implies [\varphi^A_x \equiv \uparrow] \implies [\varphi^B_{g(x)} \equiv \uparrow] \implies [g(x) \notin$B-Skip].
Computable function g is described in the lemma which stands between the statement and proof of the previous theorem. □

Sets obtained from the empty set, $\emptyset$, by a succession of jumps and skips are partially ordered according to the theorems in this section. In the diagram below, a straight line connecting two sets indicates m-comparability and the comparable higher set is higher on the page. Abbreviation aka means "also known as".

JUMP SKIP COMPARABILITY DIAGRAM



Contrast the full Jump/Skip Comparability Diagram with the corresponding comparisons in the classical theory for $\mathcal{N}$. For the ring of integers, skips and jumps coincide and the diagram thins from a bush to a stalk.

2.3.2 Non-Computable m-Complete Sets

Recall from Theorem 2.2.2.1 that K_0 is of the form {x| $\exists n$ S(x,n)} and dominates sets of its form in the partial ordering $\leq_m$. Likewise, K_1 is of the form {x| $\exists y$ S(x,y)} and dominates sets of its form. Jumps and skips of K_0 and K_1 are m-complete as well for their respective forms.

Theorem 2.3.2.1 *Jumps and skips of K_0 and K_1 are expressed as follows.*

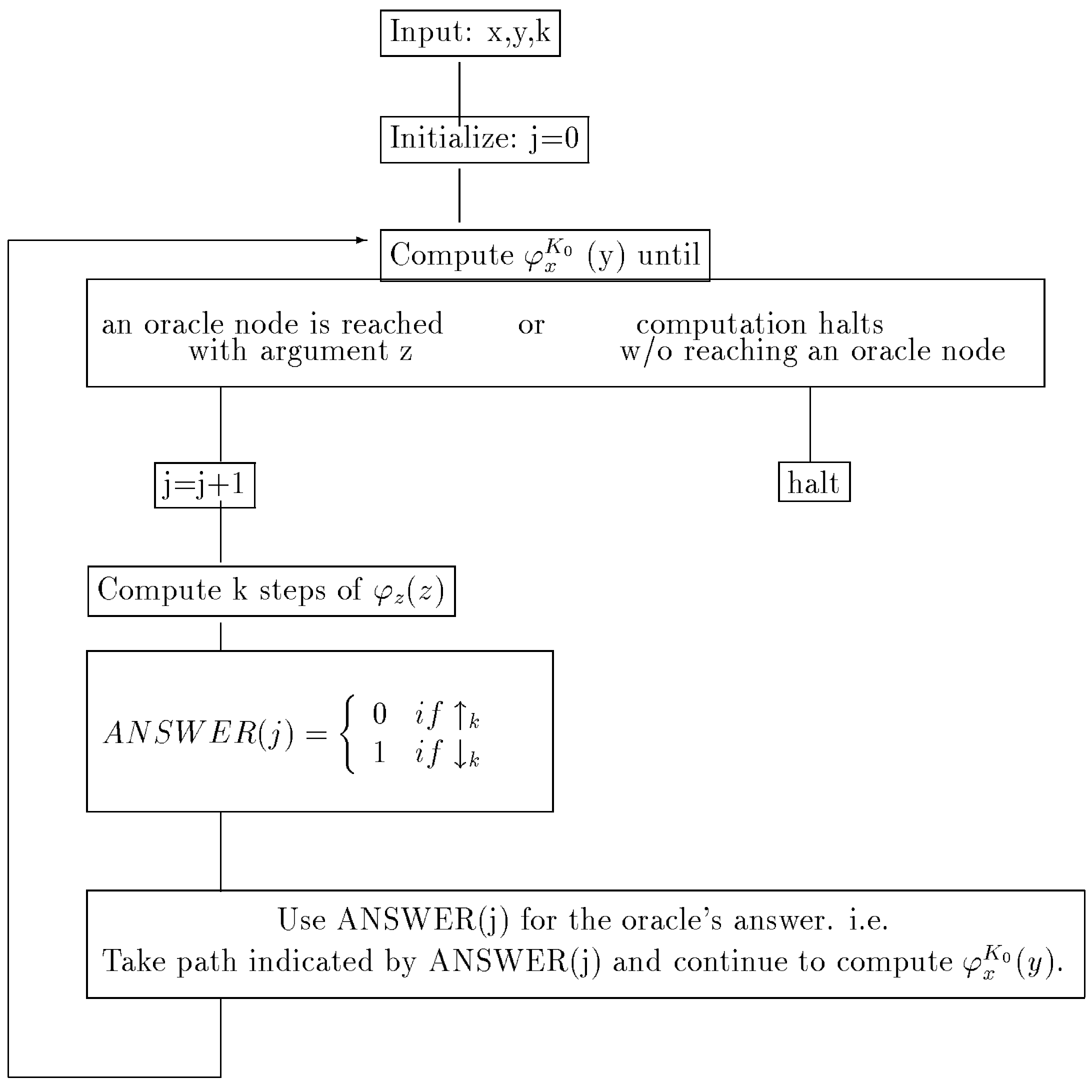
(i) K_0-Jump is of the form $\{x|\ \exists n \epsilon \mathcal{N}\ \forall k \epsilon \mathcal{N}\ \ S(x,n,k)\}$ *for recursive S.*

(ii) K_0-Skip is of the form $\{x|\ \exists \vec{y} \forall k \epsilon \mathcal{N}\ S(x,\vec{y},k)\ \}$ *for recursive S.*

(iii) K_1-Jump is of the form $\{x|\ \exists n[\exists \vec{y}\ S_1(x,n,\vec{y})\ \wedge\ \forall \vec{z}\ S_2(x,n,\vec{z})]\}$
for recursive sets S_1,S_2.

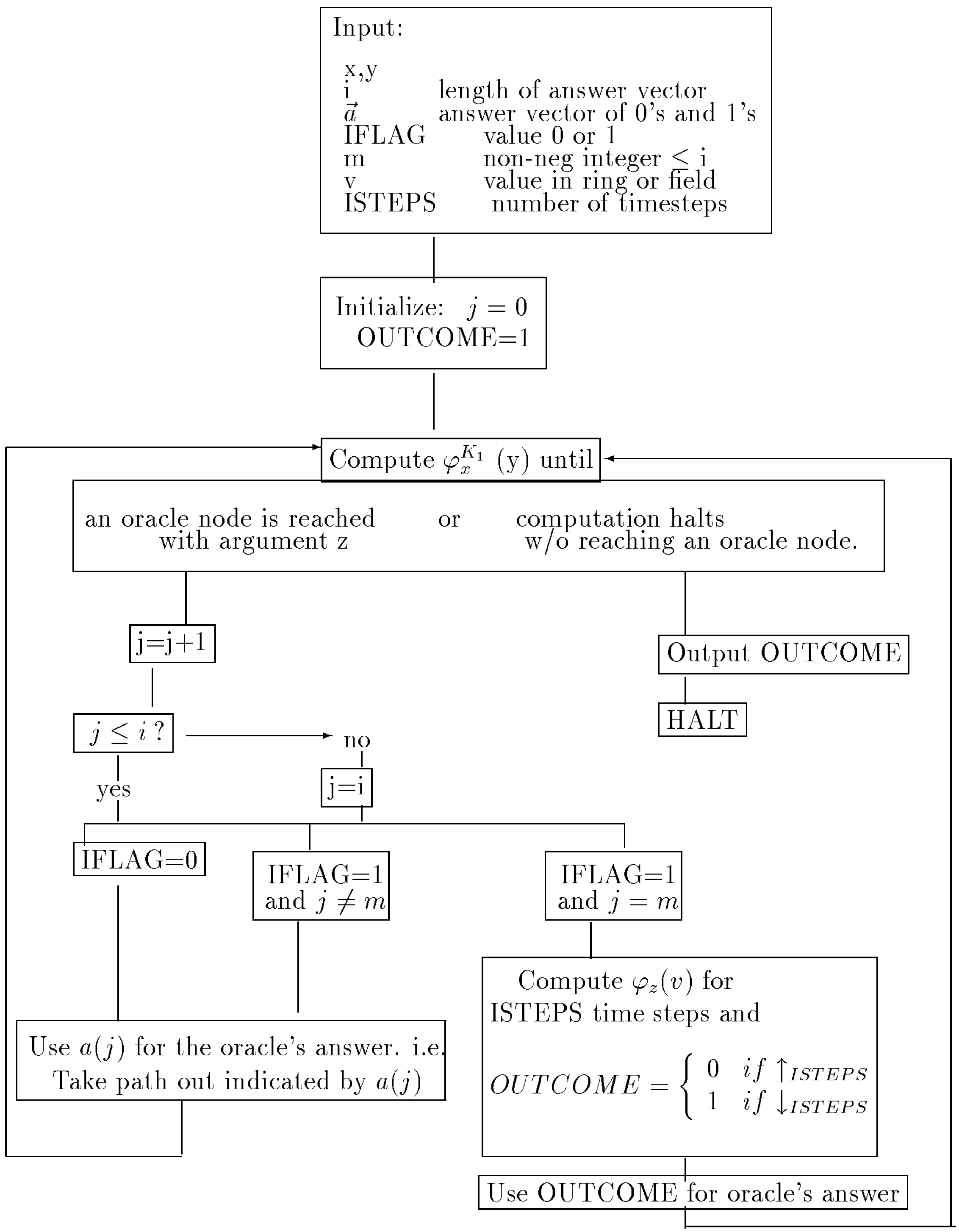
(iv) K_1-Skip is of the form $\{x|\ \exists\vec{y}\forall\vec{z}\, S(x,\vec{y},\vec{z})\}$ *for recursive S.*

Proof of (i) and (ii): Computation $\varphi_x^{K_0}(y)$ is approximated by machine M with input (x,y,k), $k\epsilon\mathcal{N}$. Machine M has no oracle. If $\varphi_x^{K_o}(y)$ halts, then for some j, exactly j calls to oracle K_0 are made during the computation. Each such oracle call results in an answer 0=no or 1=yes to the question "Does argument z_j belong to set K_0?" This chart defines BSS machine M.



Computable function g(x,y) gives the index of machine M for parameters x and y. Then $x \epsilon K_0$-Jump $\iff \varphi_x^{K_0}(x)\downarrow \iff \exists n \exists m\, \forall k \geq m\, [\varphi_{g(x,y)}(m) \downarrow_n \wedge \varphi_{g(x,y)}(m) = \varphi_{g(x,y)}(k)]$. Also, $x \epsilon K_0$-Skip $\iff \exists y\, \varphi_x^{K_0}(y)\downarrow \iff \exists y \exists n \exists m\, \forall k \geq m\, [\varphi_{g(x,y)}(m) \downarrow_n \wedge \varphi_{g(x,y)}(m) = \varphi_{g(x,y)}(k)]$. Thus K_0-Jump is written in the form $\{x|\ \exists n \forall k\, S(x,n,k)\}$ and K_0-Skip is written in the form $\{x|\ \exists \vec{y} \forall k\, S(x,\vec{y},k)\}$.

Proof of (iii) and (iv): For parameters x and y, computation $\varphi_x^{K_1}(y)$ is used to define a new BSS machine M which approximates its computation and has no oracle. Input for M consists of an answer vector $\vec{a}$ of length i, with 0 and 1 entries, a flag IFLAG, to test, and index $m \leq i$, used to mark one entry in $\vec{a}$, some new value v, and a number of timesteps, ISTEPS. Parameters x and y are also input to M. A flowchart for M is given and computable function h(x,y) is taken to be the index of machine M for parameters x,y. Then K_1-Jump and K_1-Skip are shown to be as stated in the theorem, using index function h. Here is the flowchart for M.


Input:
x,y
i length of answer vector
$\vec{a}$ answer vector of 0's and 1's
IFLAG value 0 or 1
m non-neg integer ≤ i
v value in ring or field
ISTEPS number of timesteps
Initialize: $j = 0$
OUTCOME=1
Compute $\varphi_x^{K_1}$ (y) until
an oracle node is reached with argument z
or
computation halts w/o reaching an oracle node.
j=j+1
Output OUTCOME
HALT
$j \leq i$?
no
yes
j=i
IFLAG=0
IFLAG=1 and $j \neq m$
IFLAG=1 and $j = m$
Compute $\varphi_z(v)$ for ISTEPS time steps and
$OUTCOME = \begin{cases} 0 & if \uparrow_{ISTEPS} \\ 1 & if \downarrow_{ISTEPS} \end{cases}$
Use $a(j)$ for the oracle's answer. i.e.
Take path out indicated by $a(j)$
Use OUTCOME for oracle's answer

Function $\varphi_{h(x,y)}(i, \vec{a}, IFLAG, m, v, ISTEPS)$ is computable using no oracle and is exactly $\varphi_{\lceil M \rceil}(x, y, i, \vec{a}, IFLAG, m, v, ISTEPS)$. Then $x \epsilon K_1$-Jump $\Longleftrightarrow \varphi_x^{K_1}(x) \downarrow \Longleftrightarrow$ $\exists n \exists i \exists \vec{a} [[\varphi_{h(x,x)}(i, \vec{a}, 0, 0, 0, 0) \downarrow_n] \wedge \forall j \leq i \; [a_j = 0 \longrightarrow \forall v \forall k \, \varphi_{h(x,x)}(i, \vec{a}, 1, j, v, k) \downarrow_{n+k}$ $\wedge OUTCOME = 0] \wedge \forall j \leq i [\; a_j = 0 \longrightarrow \exists v \exists k \varphi_{h(x,x)} i, \vec{a}, 1, j, v, k) \downarrow_{n+k} \; \wedge OUTCOME = 1]]$. Likewise, $x \epsilon K_1 - Skip \iff \exists y \, \varphi_x^{K_1}(y) \downarrow \iff \exists y \exists n \exists i \exists \vec{a} \; [[\; \varphi_{h(x,y)}(i, \vec{a}, 0, 0, 0, 0) \downarrow_n$ $\wedge \forall j \leq i [\; a_j = 0 \longrightarrow \forall v \forall k \, \varphi_{h(x,y)}(i, \vec{a}, 1, j, v, k) \downarrow_{n+k} \;\wedge$ OUTCOME=0] $\wedge \forall j \leq i [$ $a_j = 0 \longrightarrow \exists v \exists k \; \varphi_{h(x,y)}(i, \vec{a}, 1, j, v, k) \downarrow_{n+k} \; \wedge OUTCOME = 1]]$. Thus K_1-Jump is written as $\{$x$| \; \exists n [\forall \vec{z} \, S_2(n, x, \vec{z}) \; \wedge \; \exists \vec{z} \, S_1(n, x, \vec{z}]\}$ and K_1-Skip is written as $\{$x$| \; \exists \vec{y} \forall \vec{z} \, S(x, \vec{y}, \vec{z})\}$. □

Theorem 2.3.2.2 *For any recursive relations S, S_1, S_2,*

(i) K_0-Jump $\geq_m \; \{x| \; \exists n \forall k \; S(x, n, k)\}$.

(ii) K_0-Skip $\geq_m \; \{x| \; \exists \vec{y} \forall k \; S(x, \vec{y}, k)\}$.

(iii) K_1-Jump $\geq_m \; \{x| \; \exists n \; [\exists \vec{y} \, S_1(x, n, \vec{y}) \; \wedge \; \forall \vec{z} \, S_2(x, n, \vec{z})]\}$.

(iv) K_1-Skip $\geq_m \; \{x| \; \exists \vec{y} \forall \vec{z} \; S(x, \vec{y}, \vec{z})\}$.

Proof:
(i) Let A $=\{$x$| \; \exists n \forall k \; S(x, n, k)\}$. The relativized s-m-n Theorem gives computable function f such that

$$\varphi_{f(x)}^{K_0} = \begin{cases} 1 & if \; \exists n \forall k \, S(x, n, k) \\ undefined & otherwise \end{cases}$$

Then $x \epsilon A \iff f(x) \epsilon K_0$-Jump and so $A \leq_m K_0$-Jump.
(ii) Let A$=\{$x$| \; \exists \vec{y} \; \forall k \; S(x.\vec{y}, k\}$. The relativized s-m-n Theorem give somputable function f such that

$$\varphi_{f(x)}^{K_0}(\vec{y}) = \begin{cases} 1 & if \; \forall k \, S(x, \vec{y}.k) \\ undefined & otherwise. \end{cases}$$

Then $x \epsilon A \iff f(x) \epsilon K_0$-Skip and so $A \leq_m K_0$-Skip.
(iii) Let A$=\{$x$| \; \exists n \; [\exists y \; S_1(n, x, y) \; \wedge \; \forall z \; S_2(n, x, z)]\}$. There is a BSS machine M with oracle K_1 and halting set A. For this M, using the (relativized) s-m-n Theorem, computable function f is defined by:

$$\varphi_{f(x)}^{K_1}(v) = \begin{cases} 1 & if \;\; \varphi_{\lceil M \rceil}^{K_1}(x) \downarrow \\ undefined & otherwise \end{cases}$$

Then, $x \epsilon A \implies \varphi_{f(x)}^{K_1} \equiv 1 \implies f(x) \epsilon \, K_1$-Jump. Likewise, $x \not\epsilon A \implies \varphi_{f(x)}^{K_1} \equiv \uparrow$ $\implies f(x) \not\epsilon \, K_1$-Jump. Thus $A \leq_m K_1$-Jump.
(iv) Given set A$=\{$x$| \; \exists y \forall z \; S(x, y, z)\}$, there is a machine M with oracle K_1 such that

$$\varphi_{\lceil M \rceil}^{K_1}(x, y) = \begin{cases} 1 & if \; \forall z S(x, y, z) \\ undefined & otherwise. \end{cases}$$

Let computable function f be given by the (relativized) s-m-n Theorem so that $\varphi^{K_1}_{f(x)}(y) = \varphi^{K_1}_{\lceil M\rceil}(x,y)$. But then $A \leq_m K_1$-Skip via the function f. □

Other more natural index sets are m-complete for these quantifier configurations. The following sets are undecidable by Rice's Theorem, and are shown to be m-complete.

DOMN = $\{\lceil M\rceil \mid \forall n \exists k\ \varphi_{\lceil M\rceil}(n)\downarrow_k\}$

TOT = $\{\lceil M\rceil \mid \forall x \exists k\ \varphi_{\lceil M\rceil}(x)\downarrow_k\}$

PARAMNULL = $\{\lceil M\rceil \mid$ there are parameters $\vec{y}$ which make $\varphi_{\lceil M(\vec{y})\rceil}(\vec{l})$ the empty function.$\}$ (A remark regarding this notation follows the next definition.) PARAMNULL $= \{x \mid \exists \vec{y} \forall z \forall k\ [\varphi_x(\vec{y},\vec{z})\uparrow_k]\}$

INTPARAM = $\{\lceil M\rceil \mid \varphi_{\lceil M\rceil}$ has domain $\mathcal{N}^k$xL^m x{0,1} and range L. There is some integer parameter(s) $\vec{n}\epsilon\mathcal{N}^k$ that makes $\varphi_{\lceil M(\vec{n})\rceil}(\vec{z},0) \equiv\uparrow$ and $\varphi_{\lceil M(\vec{n})\rceil}(\vec{z},0) \not\equiv \uparrow$.$\}$

For $\varphi_{\lceil M\rceil}$ with arguments $(y_1, y_2, \ldots, y_j, l_{j+1}, \ldots l_m)$ notation $\varphi_{\lceil M(\vec{y})\rceil}$ is the function computed by machine M with fixed parameters $\vec{y}$ and arguments $\vec{l}$.

Theorem 2.3.2.3 *Index sets* $\widetilde{DOMN}$, $\widetilde{TOT}$, $INTPARAM$ *and* $PARAMNULL$ *are m-complete for their respective forms.*

(i) $\widetilde{DOMN} \equiv_m K_0$*-Jump.*

(ii) $\widetilde{TOT} \equiv_m K_0$*-Skip.*

(iii) $INTPARAM \equiv_m K_1$*-Jump.*

(iv) $PARAMNULL \equiv_m K_1$*-Skip.*

Proof: (i) For set A={x| $\forall n \exists k\, S(n,k,x)$} the s-m-n Theorem gives computable function f where

$$\varphi_{f(x)}(y) = \begin{cases} 1 & if\ \ N(y) \wedge \exists k\, S(y,k,x) \\ undefined & otherwise \end{cases}$$

Then $[x\epsilon A \iff f(x)\epsilon\, DOMN]$. *i.e.* $[x\epsilon\tilde{A} \iff f(x)\epsilon\widetilde{DOMN}]$. Hence $\widetilde{DOMN}$ m-dominates any set of the form {x| $\exists n \forall k\, S(n,k,x)$}; and so $\widetilde{DOMN} \geq_m K_0$-Jump. Theorem 2.3.1.2 shows $\widetilde{DOMN} \leq_m K_0$-Jump and hence equivalence holds.
(ii) For set A={x| $\forall y \exists n$ S(n,x,y)} the s-m-n Theorem gives computable f where

$$\varphi_{f(x)}(y) = \begin{cases} 1 & if\ \exists n\, S(n,x,y) \\ undefined & otherwise \end{cases}$$

Then [$x\epsilon A \iff f(x)$ is total]. *i.e.* $[x\epsilon\tilde{A} \iff f(x)\epsilon\widetilde{TOT}]$. Hence $\widetilde{TOT}$ m-dominates any set of the form {x| $\exists y \forall n$ S(n,x,y)} for recursive S and thus $\widetilde{TOT} \geq_m K_0$-Skip.

Theorem 2.3.1.2 shows $\widetilde{TOT} \leq_m K_0$-Skip, and so equivalence holds.
(iii) Let B={x| $\exists n[\exists y\ S_1(x,n,y) \quad \wedge \quad \forall z\ S_2(x,n,z)]$}. The s-m-n Theorem gives computable f and g such that

$$\varphi_{f(x)}(n,z) = \begin{cases} 1 & if\ S_2(x,n,z)\ fails\ to\ hold \\ undefined & otherwise. \end{cases}$$

$$\varphi_{g(x)}(n,z) = \begin{cases} 1 & if\ S_1(x,n,z)\ holds \\ undefined & otherwise. \end{cases}$$

Then let $\varphi_{h(x)}(n,z,0) = (\varphi_{f(x)}(n,z)$ and $\varphi_{h(x)}(n,z,1) = \varphi_{g(x)}(n,z)$. For computable $h(x)$,
$[x \epsilon B \iff h(x) \epsilon$ INTPARAM] which gives $B \leq_m$INTPARAM and
K_1-Jump$\leq_m$ INTPARAM. By Theorem 2.3.1.2, K_1-Jump$\equiv_m$ INTPARAM.
(iv) For B={x| $\exists y \forall z$ S(x,y,z)}, the s-m-n Theorem gives computable f where

$$\varphi_{f(x)}(y,z) = \begin{cases} 1 & if\ S(x,y,z)\ fails \\ undefined & otherwise. \end{cases}$$

Then $[x \epsilon B] \iff$[For some parameter y_0, $\varphi_{f(x)}(y_0,z) \equiv \uparrow$] $\iff$[f(x) ϵ PARAMNULL]. This means $B \leq_m$PARAMNULL and K_1-Skip $\leq_m$ PARAMNULL. Theorem 2.3.1.2 yeilds K_1-Skip $\equiv_m$ PARAMNULL. □

2.4 Rings and Fields in the Real Numbers

The hierarchy of non-computable sets built from $\emptyset$ by a succession of jumps and skips assumes a shape and configuration appropriate to the specific subring of $\mathcal{R}$ to which it corresponds.

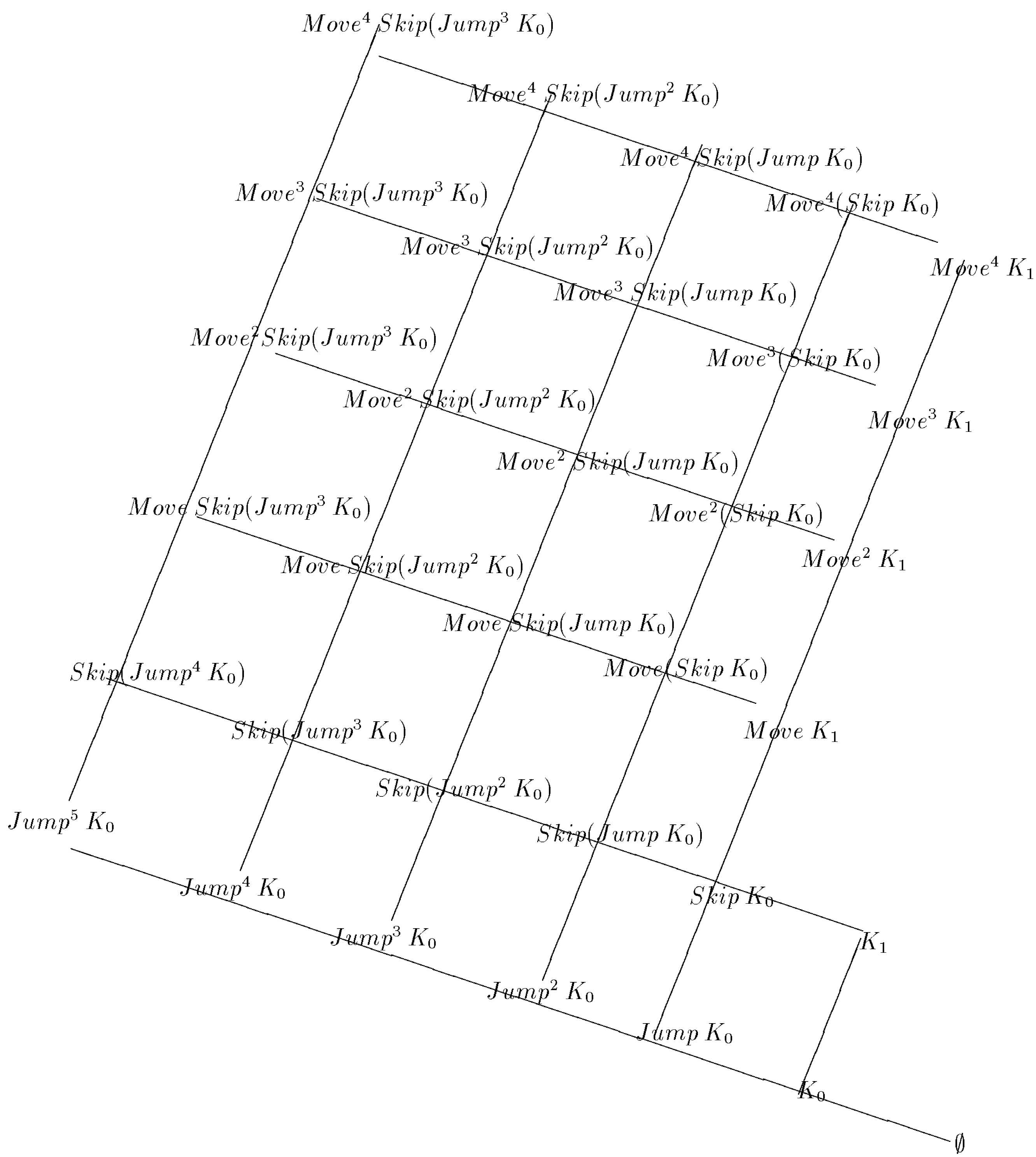
2.4.1 Finite Transcendence Over $\mathcal{Q}$

A subring L of reals of finite transcendence degree over $\mathcal{Q}$ enjoys a coding π into $\mathcal{N}$. Coding π is an L-computable isomorphism with appropriate ring operations and order in $\mathcal{N}$.

For BSS machine M, notation $\pi(M)$ is the BSS machine with input $\pi(x)$, $x \epsilon L$ and all computations done with π images in $\mathcal{N}$. If M has oracle A, then machine $\pi(M)$ has oracle $\pi(A)$. Machine M halts on x if and only if $\pi(M)$ halts on $\pi(x)$. Notation $\lceil \pi(M) \rceil$ is the code for machine $\pi(M)$ and $\varphi_{\lceil \pi(M) \rceil}$ is the partial function that $\pi(M)$ computes.
Lemma: For subring L of $\mathcal{R}$, finitely transcendental over $\mathcal{Q}$, and any set, $A \subseteq L$, $\pi(Skip(A)) \leq_m Skip(A)$ where π is a coding for L.
Proof: Skip(A) is m-complete for output sets with oracle A and is itself an output set with oracle A. The image of an output set under computable function π is an output set. □

Theorem 2.4.1.1 *For subring L of $\mathcal{R}$, finitely transcendental over $\mathcal{Q}$, and any set $A \subseteq L$, $Jump^n(Skip\,A) \equiv_m Skip^n(Skip\,A)$ for n=0, 1, 2, ...*
Proof: The proof is by induction. $Skip^2\,A = \{y \mid \exists x\ \varphi^{Skip\,A}_{\lceil M \rceil}(x) = y\ \}$, an output set for a machine M, with oracle Skip(A).
$Skip^2\,A = \{y \mid \exists n\ [n \epsilon \pi(L) \wedge \varphi^{\pi(Skip\,A)}_{\lceil \pi(M) \rceil}(n) = \pi(y)]\}$. Since $\pi(L) \leq_m K_1 = Skip(\emptyset) \leq_m Skip(A)$, and $\pi(Skip(A)) \leq_m Skip(A)$ by the lemma, the square brackets enclose a relation recursive in Skip(A). Thus $Skip^2(A) \leq_m$ Jump (Skip(A)) and the theorem holds for n=1. Now suppose the theorem holds for 1,2,3,$\ldots, n$. $Skip^{n+1}(SkipA) = Skip(Skip^{n+1}(A)) = \{y \mid \exists x \varphi^{Skip^{n+1}(A)}_{\lceil M \rceil}(x) = y\} = \{y\mid \exists n[n \epsilon \pi(L) \wedge \varphi^{\pi(Skip^{n+1}(A))}_{\lceil \pi(M) \rceil}(n) = \pi(y)]\ \} \leq_m Jump\,Skip^{n+1}(A) \equiv_m Jump\,Skip^n(Skip\,A) \equiv_m Jump\ Jump^n(Skip(A)) \equiv_m Jump^{n+1}(Skip(A))$.□
Corollary: For ring L of reals finitely transcendental over $\mathcal{Q}$,
$Skip^n(K_1) \equiv_m Jump^n(K_1)$.
Notation: Notation "Move" stands for Jump or Skip when a Jump is m-equivalent to a Skip. For example, the result of the corollary is written $Skip^n(K_1) \equiv_m Jump^n(K_1) \equiv_m Move^n(K_1)$.

When jumps and skips of $\emptyset$ are organized using this corollary and theorem, the following orderly grid occurs. Straight lines indicate $\leq_m$ and the set higher on the page is higher in the order $\leq_m$.


$Move^4\,Skip(Jump^3\,K_0)$
$Move^4\,Skip(Jump^2\,K_0)$
$Move^4\,Skip(Jump\,K_0)$
$Move^4(Skip\,K_0)$
$Move^4\,K_1$
$Move^3\,Skip(Jump^3\,K_0)$
$Move^3\,Skip(Jump^2\,K_0)$
$Move^3\,Skip(Jump\,K_0)$
$Move^3(Skip\,K_0)$
$Move^3\,K_1$
$Move^2\,Skip(Jump^3\,K_0)$
$Move^2\,Skip(Jump^2\,K_0)$
$Move^2\,Skip(Jump\,K_0)$
$Move^2(Skip\,K_0)$
$Move^2\,K_1$
$Move\,Skip(Jump^3\,K_0)$
$Move\,Skip(Jump^2\,K_0)$
$Move\,Skip(Jump\,K_0)$
$Move(Skip\,K_0)$
$Move\,K_1$
$Skip(Jump^4\,K_0)$
$Skip(Jump^3\,K_0)$
$Skip(Jump^2\,K_0)$
$Skip(Jump\,K_0)$
$Skip\,K_0$
K_1
$Jump^5\,K_0$
$Jump^4\,K_0$
$Jump^3\,K_0$
$Jump^2\,K_0$
$Jump\,K_0$
K_0
$\emptyset$

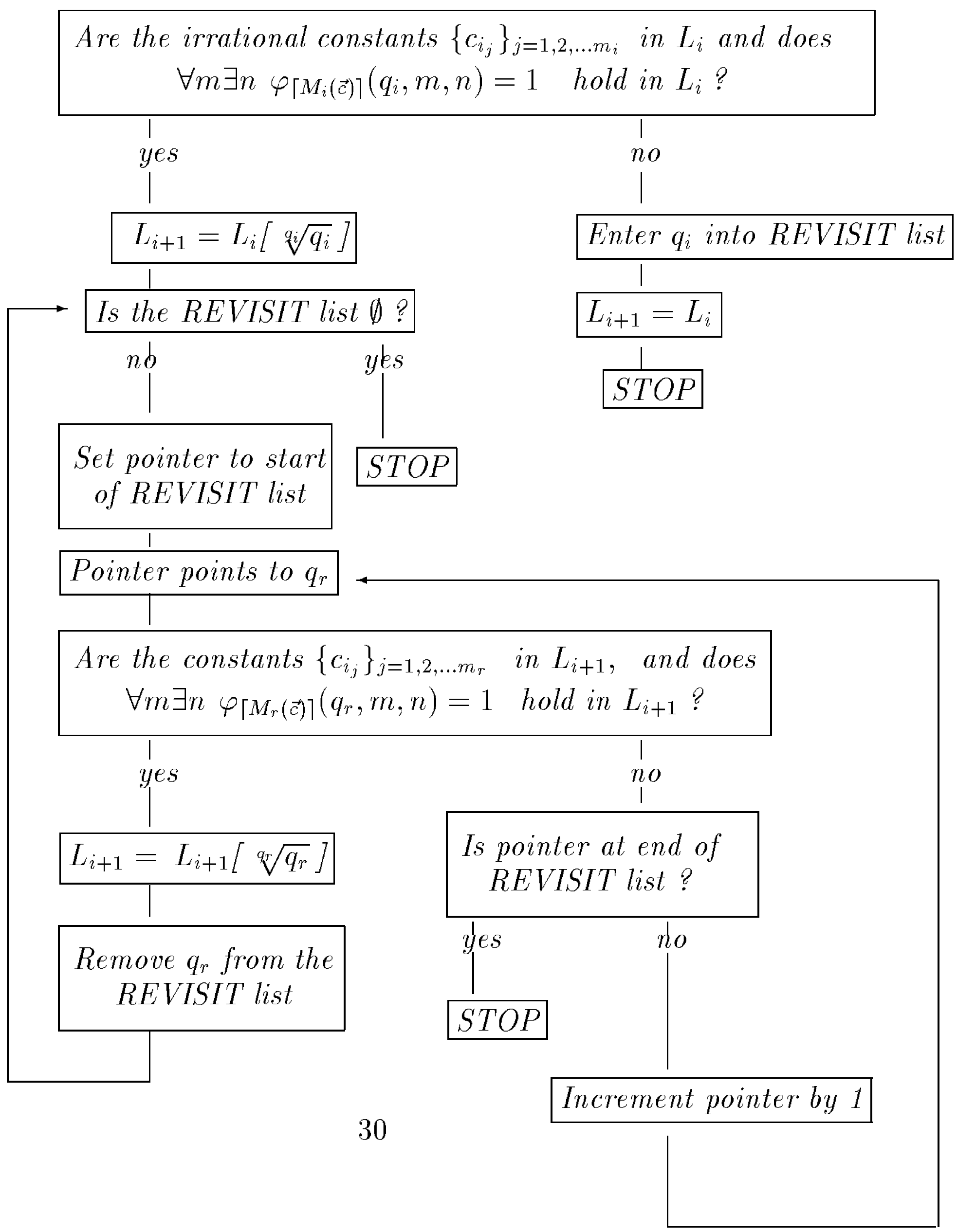
Examples show that distinct sets on the grid are indeed distinct. The following examples are illustrative.

Example 2.4.1.1 *In the ring of integers,* $Jump K_0 \not\leq_m K_1$*. Indeed, for the integers,* $K_1 =_0$ *and* $Jump\, K_0 \not\leq_m K_0$*.*

Example 2.4.1.2 *A field of algebraic reals is constructed for which* $K_1 \not\leq_m Jump\, K_0$*.*

List the prime numbers $\{q_i\}_{i\epsilon N}$ *in increasing order and list the BSS machines* $\{M_i\}_{i\epsilon N}$ *with three input arguments and output set* $\{0,1\}$*, with rational constants and irrational constants* $\{c_{i_j}\}_{j=1,2,...,m_i}$*, where each* c_{i_j} *is the* p *th root of prime* p*, with operations* $+, -, \cdot, \div$*, and tests* $<, =$*.*

A nested sequence of fields, $\mathcal{Q}=L_0 \subseteq L_1, \subseteq L_2 \ldots$*, is defined and the example field* L *is* $\bigcup L_i$*. Here is the procedure for obtaining* L_{i+1} *from* L_i*.*



Consider the set $S = \{l \epsilon L \mid l$ *is prime and* $\forall y\ y^l \neq l\} = \{$*primes p missing their p th root*$\}$. *Then* $\tilde{S} = \{l \epsilon L \mid l$ *is not prime or l has its l th root.*

If $K_1 \leq_m Jump\, K_0$, *an egregious contradiction occurs. viz:* $[K_1 \leq_m Jump\, K_0]$ *gives* $[\tilde{S} \leq_m K_1 \leq_m Jump K_0]$ *and* $[S \leq_m \widetilde{Jump}\, K_0]$. *Write* $S = \{l \mid \forall m \exists n\, R(l,m,n)\}$ *where R is a relation recursive in L. There is BSS machine* M_{i_0} *which computes*

$$\varphi_{\lceil M_{i_0} \rceil}(l,m,n) = \begin{cases} 1 & if\ \ R(l,m,n)\ \ holds \\ 0 & if\ \ R(l,m,n)\ \ fails \end{cases}$$

where M_{i_0} *has irrational machine constsnts* $\{\vec{c}\} \subseteq L$. *Here is the punch question. Is* $\sqrt[q_{i_0}]{q_{i_0}}\, \epsilon L$ *?*

Case 1: Yes! Then at some stage of the construction, $\forall m \exists n\, \varphi_{\lceil M_{i_0}(\vec{c}) \rceil}(q_{i_0}, m, n) = 1$. *Thus* $q_{i_0} \epsilon S = \{l \mid \forall m \exists n\ R(l,m,n)\}$ *and so* $\forall y\, y^{q_{i_0}} \neq q_{i_0}$ *by the original definition of S. This is impossible.*

Case 2: No! Then $q_{i_0} \epsilon S$ *by the original definition of S and so* $\forall m \exists n\, R(l,m,n)$. *Then* $\forall m \exists n\ \varphi_{\lceil M_{i_0}(\vec{c})}(q_{i_0}, m, n) = 1$ *where machine constants* $\vec{c}$ *are in L and thus are in some* L_j. *For* $t = max(i_0, j)$, *at the t th stage of construction,* $\sqrt[q_{i_0}]{q_{i_0}}$ *is adjoined. i.e.* $\sqrt[q_{i_0}]{q_{i_0}}\, \epsilon L$. *Impossible!*

Since the two cases are exhaustive and each is impossible, S cannot be written as $\{l \mid \forall m \exists n\, R(l,m,n)\}$ *and* $K_1 \not\leq_m Jump\, K_0$.

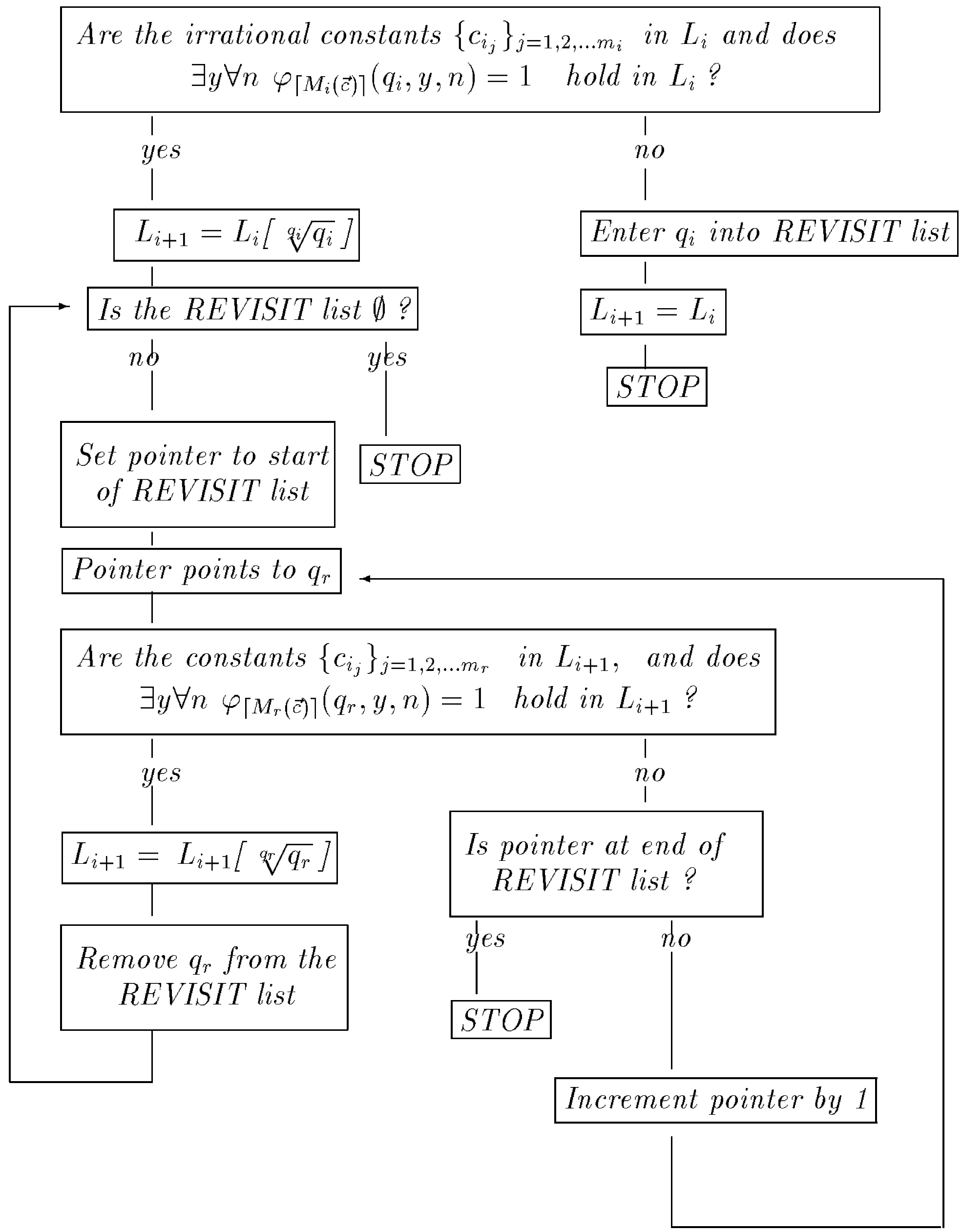
Example 2.4.1.3 *In the field constructed above,* $K_1 \not\leq_m K_0$. *Indeed, if so, then,* $K_1 \leq_m K_0 \leq_m Jump\, K_0$, *contradicting the choice of field L.*

Example 2.4.1.4 *In the field constructed above,* $Skip\, K_0 \not\leq_m Jump\, K_0$. *Indeed, if so,* $K_1 \equiv_m Skip\, \emptyset \leq_m Skip\, K_0 \leq_m Jump\, K_0$.

Example 2.4.1.5 *A field of algebraic reals is constructed for which* $Jump\, K_1 \not\leq_m Skip\, K_0$. *This is accomplished by constructing a field for which* $\widetilde{K_1} \not\leq_m Skip\, K_0$ *and recalling that* $\widetilde{K_1} \leq_m Jump\, K_1$, *by Theorem 2.3.1.2.*

List the prime numbers $\{q_i\}_{i \epsilon N}$ *in increasing order and list the BSS machines* $\{M_i\}_{i \epsilon N}$ *with three input arguments and output set* $\{0,1\}$, *with rational constants and irrational constants* $\{c_{i_j}\}_{j=1,2,\ldots,m_i}$, *where each* c_{i_j} *is the p th root of prime p, with operations* $+, -, \cdot, \div$, *and tests* $<, =$.

A nested sequence of fields, $\mathcal{Q} = L_0 \subseteq L_1, \subseteq L_2 \ldots$, *is defined and the example field L is* $\bigcup L_i$. *Here is the procedure for obtaining* L_{i+1} *from* L_i.


Are the irrational constants $\{c_{i_j}\}_{j=1,2,\ldots m_i}$ in L_i and does $\exists y \forall n \ \varphi_{\lceil M_i(\vec{c})\rceil}(q_i, y, n) = 1$ hold in L_i ?
yes
no
$L_{i+1} = L_i[\ \sqrt[q_i]{q_i}\ ]$
Enter q_i into REVISIT list
Is the REVISIT list $\emptyset$?
$L_{i+1} = L_i$
no
yes
STOP
Set pointer to start of REVISIT list
STOP
Pointer points to q_r
Are the constants $\{c_{i_j}\}_{j=1,2,\ldots m_r}$ in L_{i+1}, and does $\exists y \forall n \ \varphi_{\lceil M_r(\vec{c})\rceil}(q_r, y, n) = 1$ hold in L_{i+1} ?
yes
no
$L_{i+1} = L_{i+1}[\ \sqrt[q_r]{q_r}\ ]$
Is pointer at end of REVISIT list ?
Remove q_r from the REVISIT list
yes
no
STOP
Increment pointer by 1

Consider the set $S = \{l \epsilon L \mid l$ *is prime and* $\forall y\ y^l \neq l\} = \{primes\ p\ missing\ their$ *p th root*$\}$.

If $\widetilde{K_1} \leq_m Skip\, K_0$*, an egregious contradiction occurs. viz:* $[\widetilde{K_1} \leq_m Skip\, K_0]$ *gives* $[S \leq_m Skip\, K_0]$*. Write* $S = \{l \mid \exists y \forall n\ R(l,y,n)\}$ *where R is a relation recursive in L. There is BSS machine* M_{i_0} *which computes*

$$\varphi_{\lceil M_{i_0} \rceil}(l,y,n) = \begin{cases} 1 & if\ \ R(l,y,n)\ \ holds \\ 0 & if\ \ R(l,y,n)\ \ fails \end{cases}$$

where M_{i_0} *has irrational machine constsnts* $\{\vec{c}\} \subseteq L$*. Here is the punch question. Is* $\sqrt[q_{i_0}]{q_{i_0}}\ \epsilon L$ *?*

Case 1: Yes! Then at some stage of the construction, $\exists y \forall n\ \varphi_{\lceil M_{i_0}(\vec{c}) \rceil}(q_{i_0}, y, n) = 1$*. Thus* $q_{i_0} \epsilon S = \{l \mid \exists y \forall n\ R(l,y,n)\}$ *. But by the original definition of S,* $q_{i_0} \notin S$*.*

Case 2: No! Since $q_{i_0} \epsilon S$*, using the* $Skip\, K_0$ *definition of S,* $\exists y \forall n\ R(q_{i_0}, y, n)$ *holds in L. Thus, for some t, at the t th stage,* $\exists y \forall n \varphi_{\lceil M_{i_0} \rceil}(q_{i_0}, y, n) = 1$ *holds and so* $\sqrt[q_{i_0}]{q_{i_0}}$ *is adjoined, giving* $q_{i_o} \notin S$*.*

Since the two cases are exhaustive and each is impossible, S cannot be written as $\{l \mid \exists y \forall n\, R(l,y,n)\}$ *and* $\widetilde{K_1} \not\leq_m Skip\, K_0$*. Therefore,* $Jump\, K_1 \not\leq_m Skip\, K_0$*.*

Example 2.4.1.6 *In the field constructed above,* $Jump\ K_1 \not\leq_m Jump\ K_0$*, and* $Skip\, K_1 \not\leq_m Jump\, K_0$*. (Note that Jump-*$K_1 \equiv_m$ *Skip-*K_1*.)*

Example 2.4.1.7 *In the field constructed above,* $Skip\, K_1 \not\leq_m Skip\, K_0$*.*

The grid of sucessive jumps and skips of $\emptyset$ collapses to a stalk if the image, $\pi(L)$, is (classically) Σ_1 or Σ_2 as a subset of $\mathcal{N}$. These are special cases (j=1,2) of the following theorem and its corollaries.

Theorem 2.4.1.2 *For subring L of* $\mathcal{R}$*, finitely transcendental over* $\mathcal{Q}$*, with coding* π*, for all* $j \epsilon \mathcal{N}$ *, if* $[\pi(L) \leq_m Jump^j\, \emptyset]$ *then* $[K_1 \leq_m Jump^j\, \emptyset]$*.*

Proof: Since K_1 is an output set, it suffices to show that any L-output set O is below $Jump^j\, \emptyset$ in the partial ordering $\leq_m$. Let O=$\{$y$\mid \exists x\ S(x,y)\}$ for recursive relation S. By hypothesis, $\pi(L) = \{m \mid \exists k\ \ S_1^{Jump^{j-1}\emptyset}(m,k)\}$ where relation S_1 is recursive in (oracle) $Jump^{j-1}\, \emptyset$. Then, $O = \{y \mid \exists n\ \ [n \epsilon \pi(L) \wedge \mathrm{S}_\pi(n, \pi(y))]\}$= $\{$y $\mid \exists n$ $[\exists k\ S_1^{Jump^{j-1}\emptyset}(n,k)\ \wedge\ S_\pi(n,\pi(y))]\}$= $\{y \mid \exists n \exists k\ \ S_2^{Jump^{j-1}\emptyset}(n,k,y)\}$ for relation S_2, recursive in $Jump^{j-1}\emptyset$. Thus $O \leq_m Jump(Jump^{j-1}\emptyset)$, and $K_1 \leq_m Jump^j\, \emptyset$.□

Corollary: $[\pi(L) \leq_m K_0] \implies [K_1 \equiv_m\ K_0]$.

Proof: This is Byerly's result. The theorem is restated for j=1. □

Corollary: If $\pi(L)$ is Σ_j as a subset of $\mathcal{N}$ in classical recursion theory then $K_1 \leq_m Jump^j\, \emptyset$ in L-recursion theory.

Proof: By induction, if subset A of $\mathcal{N}$ is Σ_j in classical recursion theory then in L-recursion theory, $A \leq_m \mathrm{Jump}^j\, \emptyset$. For j=1, note that A is a halting set for a BSS machine with constants in $\mathcal{N}$ and is thus a halting set for a BSS machine over L.

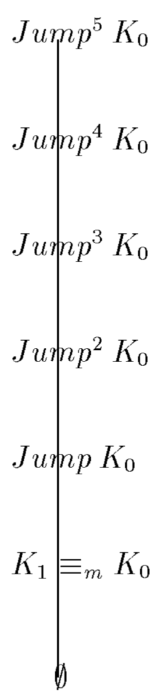
Suppose the assertion holds for 1,2,3,...,j and A is Σ_{j+1} in $\mathcal{N}$. Then A is a halting set set for a machine with constants in $\mathcal{N}$ and with oracle B which is Σ_j in $\mathcal{N}$. By induction hypothesis, $B \leq_m Jump^j\,\emptyset$ and so A is a halting set over L with oracle $Jump^j\,\emptyset$. i.e. $A \leq_m Jump^{j+1}\,\emptyset$. The theorem applied to $A = \pi(L)$ yields the corollary. □

Corollary: If $\pi(L)$ is either an L-halting set or a Σ_1 subset of $\mathcal{N}$ in classical recursion theory, then $Jump^n(K_0) \equiv_m Skip^n(K_0) \equiv_m Jump^n(K_1) \equiv_m Skip^n(K_1)$.
Proof: The assertion follows from Theorem 2.4.1.1 and the first corollary following Theorem 2.4.1.2.

Corollary: For subring L of $\mathcal{R}$, finitely transcendental over $\mathcal{Q}$, and coding π of L into $\mathcal{N}$, if $\pi(L)$ is either an L-halting set or a classical Σ_1 subset of $\mathcal{N}$, then the successive jumps and skips of $\emptyset$ form this linearly ordered stalk.

Finite Transcendence Degree Over $\mathcal{Q}$

Special Case I



The ring of integers and those subrings of $\mathcal{R}$ which can be listed have these Jump/Skip relationships. For these subrings, coding π may be taken so that image $\pi(L)$ is all of $\mathcal{N}$.

Theorem 2.4.1.3 *For subring L of $\mathcal{R}$, finitely transcendental over $\mathcal{Q}$, and for coding π, if image, $\pi(L)$ is either Σ_2 in $\mathcal{N}$ or $\leq_m Jump^2\,\emptyset$ in L, then $Move^{k-1}\,K_1 \leq_m Jump^k\,K_0 \leq_m Skip(Jump^{k-1}\,K_0)$ for k=1,2,3,....*

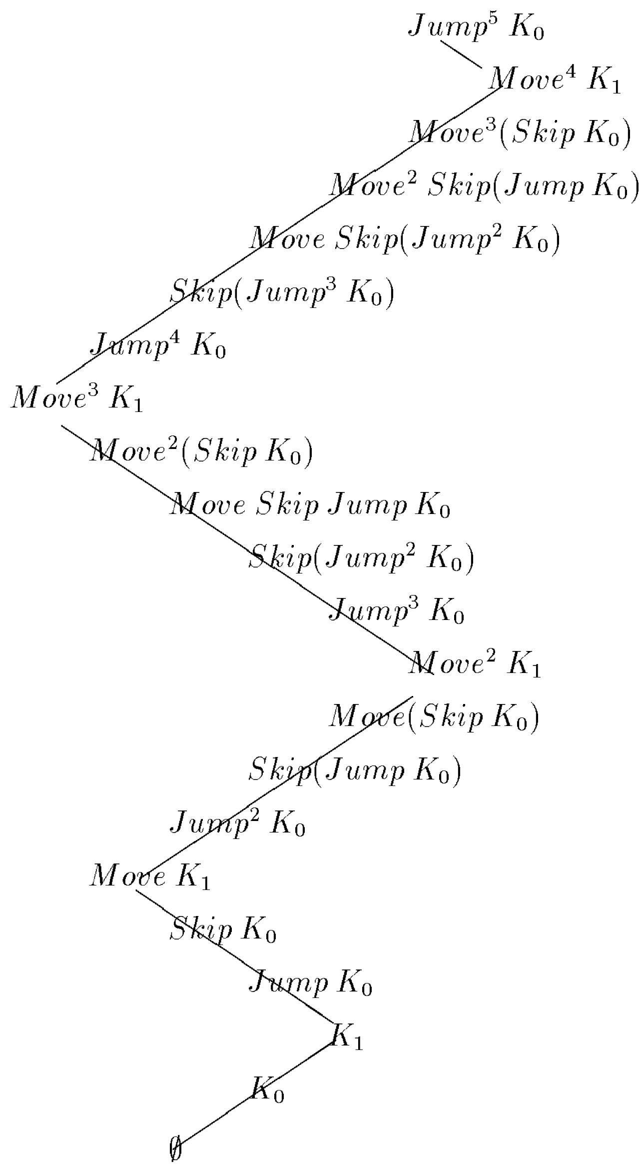
Proof: Theorem 2.4.1.2 gives $K_1 \leq_m Jump^2\,\emptyset \equiv_m Jump\,K_0$. The first corollary of Theorem 2.4.1.1 gives $Move^{k-1}\,K_1 \leq_m Jump^{k-1}\,K_1$. Together these imply $Move^{k-1}\,K_1 \leq_m Jump^{k-1}(Jump\,K_0)$. Since the skip of any set exceeds or equals its jump, $Move^{k-1}\,K_1 \leq_m Jump^k\,K_0 \leq_m Skip(Jump^{k-1}\,K_0)$.□

Corollary: For subring L of $\mathcal{R}$, finitely transcendental over $\mathcal{Q}$, and for coding π, if image, $\pi(L)$ is either Σ_2 in $\mathcal{N}$ or $\leq_m Jump^2\,\emptyset$ in L, then the successive jumps and skips of $\emptyset$ are linearly ordered in the ordering $\leq_m$.

Proof: The relationships for sets given in Theorem 2.4.1.3 along with relationships previously proved for jumps and skips yield the Jump/Skip diagram below.

Finite Transcendence Degree Over $\mathcal{Q}$

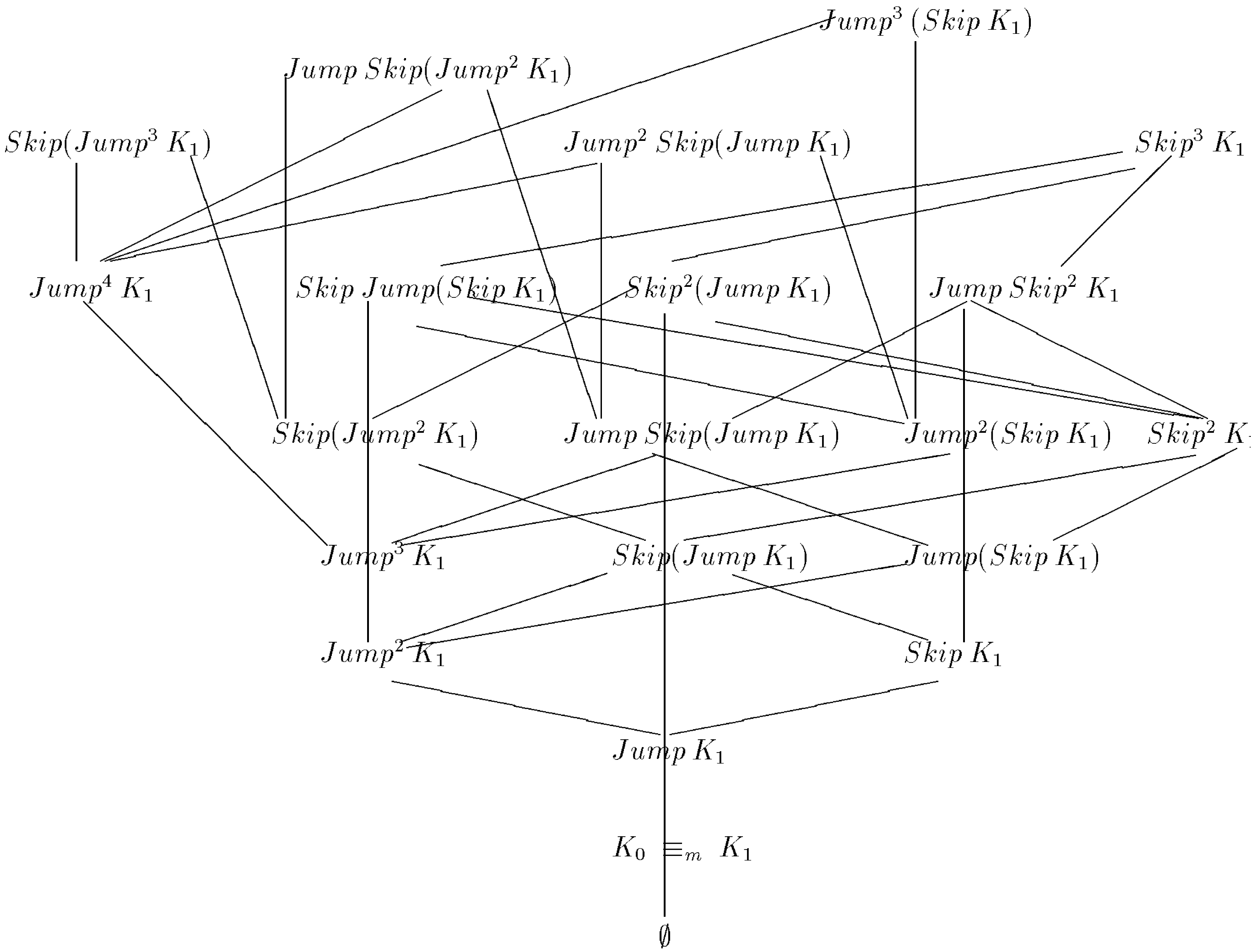
Special Case II



2.4.2 Infinite Transcendence Over $\mathcal{Q}$

Rings infinitely transcendental over $\mathcal{Q}$ for which 0=H have been shown to be real closed fields. In $\mathcal{R}$, skips and jumps are generally distinct. That is, it matters whether quantification applies to $\mathcal{R}$ or to $\mathcal{N}$. Successive jumps and skips from $\emptyset$ are diagrammed below for real closed fields, for up to four quantifiers.

Real Closed Fields, Infinite Transcendence Degree Over $\mathcal{Q}$



Rings and fields in $\mathcal{R}$ which fail to be real closed may not enjoy collapsing of the full Jump/Skip hierarchy.

2.4.3 Conclusion

The full hierarchy of non-computable sets built from $\emptyset$ by successive jumps and skips collapses into specific configurations for specific subrings (fields) of real numbers. Subrings finitely generated over $\mathcal{Z}$ or $\mathcal{Q}$ have a hierarchy consisting merely of jumps which corresponds, for example, to the jump hierarchy of classical recursion theory over the natural numbers. Some other subrings (fields) of finite transcendence degree over $\mathcal{Q}$ enjoy a linearly ordered Jump/Skip hierarchy considerably richer than the classical hierarchy for $\mathcal{N}$.

The Jump/Skip hierarchy for subrings (fields) of infinite transcendence degree over $\mathcal{Q}$ collapses in the case of real closed fields due to the coincidence of output and halting sets, but otherwise remains full. Rings and fields of reals of infinite transcendence degree over $\mathcal{Q}$ which fail to be real closed may retain the full Jump/Skip configuration.

Acknowledgements

This author and her work have enjoyed the kind support of Lenore Blum, Mike Shub, and Steve Smale. I am especially grateful for the unwavering generosity and eclectic intellect of Leo Harrington.

Bibliography

[BSS] L.Blum, M.Shub, and S.Smale. *On a Theory of Computation and Complexity over the Real Numbers: NP Completeness, Recursive Functions and Universal Machines.* Bull.Amer.Math.Soc.(N.S.) **21** (1989) pp.1-46.

[By] R.E.Byerly. *Ordered Subrings of the Reals in which Output Sets are Recursively Enumerable.* Proc.Amer.Math.Soc. Volume **118**, Number 2 (1993) pp.597-601.

[FM] H.Friedman, and R.Mansfield. *Algorithmic Procedures.* Volume **332**,Number 1, (1992) pp.297-312.

[Mi] C.Michaux. *Ordered Rings over which Output Sets are Recursively Enumerable.* Proc.Amer.Math.Soc. **112** (1991) pp.569-575.

[So] R.I.Soare. *Recursively Enumerable Sets and Degrees.* Springer-Verlag, Berlin.(1987).