Notre Dame Journal of Formal Logic
Volume 39, Number 4, Fall 1998

An Undecidable Linear Order That Is n-Decidable for All n

JOHN CHISHOLM and MICHAEL MOSES

Abstract A linear order is *n-decidable* if its universe is $\mathbb{N}$ and the relations defined by Σ_n formulas are uniformly computable. This means that there is a computable procedure which, when applied to a Σ_n formula $\varphi(\bar{x})$ and a sequence $\bar{a}$ of elements of the linear order, will determine whether or not $\varphi(\bar{a})$ is true in the structure. A linear order is *decidable* if the relations defined by *all* formulas are uniformly computable.

These definitions suggest two questions. Are there, for each n, n-decidable linear orders that are not $(n+1)$-decidable? Are there linear orders that are n-decidable for all n but not decidable? The former was answered in the positive by Moses in 1993. Here we answer the latter, also positively.

1 Introduction The study of computable algebraic structures has a long and by now widely known history. Beginning in the "finitistic" demands of the algebraists of the late 1800s and early 1900s, it hit its stride in the papers of Fröhlich and Shepherdson [4] and Rabin [7] which set the tone for much of what was to follow. A good introduction to such a computable analysis as applied to linear orders is provided in the final chapter of Rosenstein [8]; a more current and comprehensive, almost encyclopedic treatment in Downey [1].

A linear order is *computable* if its universe is $\mathbb{N}$ (we will take this to be the case from here on) and the quantifier-free formulas uniformly denote computable relations. This means that there is a computable procedure, which, when applied to a quantifier-free formula $\varphi(\bar{x})$ and a sequence $\bar{a}$ of elements of the linear order, will determine whether or not $\varphi(\bar{a})$ is true in the structure. (Quite clearly this is equivalent to demanding just that the order relation be computable.) A linear order is *decidable* if *all* formulas uniformly denote computable relations. Between these two concepts lies that of an *n-decidable* linear order, defined to be one in which the Σ_n formulas uniformly denote computable relations.

Moses [6] answered the first of the questions in our abstract by constructing, for each n, a linear order that is n-decidable but not $(n+1)$-decidable and, in fact, has no

Received January 7, 1999; revised November 15, 1999

$(n+1)$-decidable copy. In this paper we answer the second by constructing a linear order that is n-decidable for all n but is not decidable and, in fact, has no decidable copy. To make the linear order n-decidable we will arrange that the Σ_n formulas uniformly denote computable relations. To ensure it has no decidable copy we will arrange that the set of *sentences* true in our linear order is not computable. These conflicting requirements, that truth can be effectively determined for Σ_n formulas, but that this cannot be done uniformly in n, not even for sentences, produces the tension in our construction. This paper then will be devoted to establishing the following.

Theorem 1.1 *There is a linear order $\mathfrak{L}$ that is n-decidable for all n but has no decidable copy.*

Our terminology will be standard, as presented for instance in [8], or will be obvious from the context. We will sometimes replace a sub- or superscript by the wildcard symbol $*$ to allow easy reference to the structures denoted by the range of the sub/superscript: L_*, for instance, will represent (any and every) one of $L_1, L_2, \ldots, L_k$. Note that we will mean L_* to denote any and every one of $L_1, L_2, \ldots, L_k$ *every* time we use it, that is, repeated usage, even in the same sentence, is intended to denote (possibly) different linear orders. If we wish to refer repeatedly to a specific one of them we will use the more standard L_i. We mean the labels to denote (classical) order types; we will not use individual labels for the several separate copies of each order type. We believe, or at least hope, that this conservation of symbols will not cause confusion.

2 Ehrenfeucht-Fraïssé Games We extend the notation $L_1 \equiv L_2$ (elementary equivalence) to $L_1 \equiv_n L_2$, meaning that the two linear orders satisfy the same Σ_n sentences. To establish $L_1 \equiv_n L_2$ we will use a modification of the Ehrenfeucht-Fraïssé Games. Consider a two-person game played on the linear orders L_1 and L_2 by the players **P1** and **P2**. Two numbers are set before the game begins: n, the number of moves each player will make, and k, whose usage will be described below. The players will move in turn, with **P1** playing first. At each move **P1** will select a sequence of at most k elements wholly contained in *either* one of the linear orders and **P2** will select a sequence of the same length in the other. **P2**'s aim is to arrange that the (finite) suborder of L_1 that consists of all the elements selected so far (by **P1** *and* **P2**) is isomorphic to the corresponding set of elements in L_2 via the mapping that identifies the sequences chosen by the two players at each move. If **P2** is able to match **P1**'s choice for n moves, *extending* the isomorphism to include the new sequence each time, we say **P2** has won the *n-k* **E-F Game**; otherwise **P1** is the victor.

Note that the number of moves and the maximum length of the sequences selected at each move are established in advance. Notice also that the sequence **P1** selects may come from *either* linear order and that **P2**'s sequence must produce an isomorphism that *extends* the existing isomorphism. The original Ehrenfeucht-Fraïssé Games, introduced implicitly in Fraïssé [3] and explicitly in Ehrenfeucht [2], and applied extensively to linear orders in [8], is the restriction of our games caused by fixing k at 1. The obvious modification of the arguments presented there establishes this:

> *For each n, $L_1 \equiv_n L_2$ if, for every k,* **P2** *can win every n-k* **E-F Game** *played on L_1 and L_2.*

Allow us to reiterate: the relation $\equiv_n$, corresponding to our version of the Ehrenfeucht-Fraïssé Games, is different from the usual one, most often denoted $\sim_n$, which, in our notation, would require only that **P2** can win every n-1 **E-F Game**, and implies only that L_1 and L_2 satisfy the same sentences of *quantifier depth n*. This is not strong enough for our purposes.

3 Shuffles The *shuffle* of linear orders $L_1, L_2, \dots, L_k$, denoted $\sigma(L_1, L_2, \dots, L_k)$, is the linear order produced by partitioning η, the dense linear order without endpoints, into k subsets, each of which is dense in η, and replacing each point in the ith of these subsets with a copy of L_i.

We will construct our linear order $\mathfrak{L}$ as the limit of a sequence of linear orders $L_1, L_2, L_3, \dots$. Each L_i will be a shuffle (enclosed within a pair of endpoints) of linear orders $L^1_{i-1}, L^2_{i-1}, \dots, L^k_{i-1}$, produced at the previous stage, the first of which will be L_{i-1}. We begin with $L_0 = \mathbf{1} + \sigma(\mathbf{2}, \mathbf{3}, \mathbf{4}) + \mathbf{1}$, the shuffle of the two, three, and four point linear orders, enclosed by a pair of points. From then on, to build L_i from L_{i-1}, we consider the different shuffles produced by shuffling all except one of the linear orders used in L_{i-1}, omitting each one in turn, and produce L_i by shuffling these (with perhaps the last one omitted) together with L_{i-1} and enclosing them between endpoints. L_1, for instance, will be the shuffle of L_0 together with the linear orders $\mathbf{1} + \sigma(\mathbf{3}, \mathbf{4}) + \mathbf{1}$, $\mathbf{1} + \sigma(\mathbf{2}, \mathbf{4}) + \mathbf{1}$, and $\mathbf{1} + \sigma(\mathbf{2}, \mathbf{3}) + \mathbf{1}$ (with perhaps the last one omitted), with endpoints added. L_2 will be produced by shuffling L_1 together with the shuffles produced by leaving out from L_1, in turn, each one of the linear orders $\mathbf{1} + \sigma(\mathbf{3}, \mathbf{4}) + \mathbf{1}$, $\mathbf{1} + \sigma(\mathbf{2}, \mathbf{4}) + \mathbf{1}$, and $\mathbf{1} + \sigma(\mathbf{2}, \mathbf{3}) + \mathbf{1}$, and adding endpoints, and so on.

Since we intend to construct a linear order with certain computable properties, we need to describe our construction in some detail. We will build each $L_i = \mathbf{1} + \sigma(L^1_{i-1}, \dots, L^k_{i-1}) + \mathbf{1}$ around L_{i-1} (which will be L^1_{i-1}) in such a way that we will keep careful track of which elements of $\mathbb{N}$ are used in which copy of L^*_{i-1}, where these elements lie within their separate L^*_{i-1}'s (i.e., with respect to the linear orders of which the L^*_{i-1} is a shuffle) and where these separate L^*_{i-1}'s lie with respect to each other. There are several ways of performing such a construction. We describe one: clearly we can construct a copy of L_0 with the required properties. Assuming that we can perform such constructions for all the L^*_{i-1}'s, we build a copy of $\sigma(L^1_{i-1}, \dots, L^k_{i-1})$ by first laying down copies of the L^*_{i-1}'s with the required properties, in order (any order will do, we choose the obvious one), to form the sum $L^1_{i-1} + \cdots + L^k_{i-1}$. At each further stage we lay down several such sums, one between each adjacent pair of existing L^*_{i-1}'s, one to the extreme left and one to the extreme right. It should be clear that we can mesh the construction of the separate copies of the L^*_{i-1}'s in such a way that we know exactly where each element of the universe $\mathbb{N}$ lies with respect to the particular L^*_{i-1} in which it lies, and with respect to the linear order within that L^*_{i-1} (one of those shuffled to produce that L^*_{i-1}) in which it lies (and, in fact, with respect to the linear order within that, and within that, all the way down to the copy of $\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, or $\mathbf{4}$ in which this element lies). We will see that this will allow us to show that $\mathfrak{L}$, the linear order constructed by this infinite process, will be n-decidable for every n.

Note that each of our L_i has a decidable copy. One way to see this is to observe that L_i can be defined, up to isomorphism, by a (first-order) sentence. The sentence for $\sigma(L^1_{i-1}, \ldots, L^k_{i-1})$ would say that each element lies within a closed interval $[x, y]$ isomorphic to one of the (finitely many) L^*_{i-1}'s, and that to the left of this interval and to its right and between it and every interval isomorphic to a different L^*_{i-1}, there lie intervals isomorphic to each one of the L^*_{i-1}'s. We can write this as a sentence in the language of linear order by incorporating sentences that define the L^*_{i-1}'s. That this sentence defines the linear order up to isomorphism can be seen via a Cantor back-and-forth argument. Since the theory of linear orders is decidable, it follows that $\mathrm{Th}(L_i)$ is computable, and hence (by the Henkin construction, which is algorithmic) has a decidable model which must be isomorphic to L_i (since it satisfies the defining sentence).

We will show that $\mathfrak{L}$ is n-decidable by providing a computable procedure which, when applied to any sequence $\bar{a}$ in $\mathfrak{L}$, will produce a sequence $\bar{b}$ in the decidable copy of L_n which satisfies there precisely the same Σ_n formulas that $\bar{a}$ satisfies in $\mathfrak{L}$. Since the copy of L_n is decidable, this will imply that $\mathfrak{L}$ is n-decidable.

To guarantee that $\mathfrak{L}$ has no decidable copy we will make $\mathrm{Th}(\mathfrak{L})$ noncomputable by allowing each L_i to be one of two possible linear orders, L_i^- and L_i^+, and by choosing between them in such a way that we diagonalize across a list of the computably enumerable functions, preventing each of them from being the characteristic function for $\mathrm{Th}(\mathfrak{L})$. The linear orders L_i^- and L_i^+ will be distinguished from each other by a Π_{i+3} sentence ψ_i, true of the former but not of the latter. L_i^- will be a shuffle of L^*_{i-1}'s and L_i^+ will be a shuffle of the same L^*_{i-1}'s *with one additional L^*_{i-1} shuffled in*. This will be the only difference between them. This additional L^*_{i-1}, appearing in L_i^+ but not in L_i^-, will contain an interval of a certain order type, definable by a Σ_{i+3} sentence, not isomorphic to any interval in L_i^-. The negation of this Σ_{i+3} sentence will be the ψ_i true of L_i^- but not of L_i^+. Since ψ_i asserts the nonexistence of an interval of a certain order type, it follows that it will distinguish also between an $\mathfrak{L}$ in which L_i^- was chosen to be L_i and one in which L_i^+ was chosen to be L_i. Our default is that $L_i = L_i^-$; we begin building L_i^- and switch to L_i^+ only if the ith computably enumerable function says that $\mathfrak{L} \models \psi_i$. The fact that L_i^+ is just L_i^- with one extra L^*_{i-1} shuffled in will allow us to make this switch at any time. In this manner we put the ith computably enumerable function out of the running as a possible enumerator of $\mathrm{Th}(\mathfrak{L})$.

This strategy of ours, when employed against an L_i, will cause us to change L_i from L_i^- to L_i^+, and consequently change every L_j with $j > i$, and consequently change their defining sentences ψ_j. Even if we have already acted against these L_j, we will need to reconsider them and perhaps act against them again once the jth computably enumerable function has made up its mind on the truth, in $\mathfrak{L}$, of the new defining sentence ψ_j, whence the (finite) injury in our construction.

Notice that this does not jeopardize our strategy for ensuring that $\mathfrak{L}$ be n-decidable: for a given n we just have to *guess* at a stage when $L_1, \ldots, L_n$ have all settled down (between being L_*^- or L_*^+); we can then use the decidable copy of that L_n (which will never again be tampered with) to provide a decision procedure for the Σ_n formulas in $\mathfrak{L}$. It is the *uniformity*, over n, of these decisions, that we will have impaired.

4 Construction of the L_i We begin with $L_0 = \mathbf{1} + \sigma(\mathbf{2}, \mathbf{3}, \mathbf{4}) + \mathbf{1}$. In general, having defined L_{i-1} to be a linear order of the form $\mathbf{1} + \sigma(L^1_{i-1}, \dots, L^k_{i-1}) + \mathbf{1}$, we define L_i as follows:

$$\begin{aligned}
L^1_i &= \mathbf{1} + \sigma(L^1_{i-1}, \dots, L^k_{i-1}) + \mathbf{1}, \quad \text{that is, } L_{i-1},\\
L^2_i &= \mathbf{1} + \sigma(L^2_{i-1}, \dots, L^k_{i-1}) + \mathbf{1},\\
L^3_i &= \mathbf{1} + \sigma(L^1_{i-1}, L^3_{i-1}, \dots, L^k_{i-1}) + \mathbf{1},\\
L^4_i &= \mathbf{1} + \sigma(L^1_{i-1}, L^2_{i-1}, L^4_{i-1}, \dots, L^k_{i-1}) + \mathbf{1},\\
&\ \ \vdots\\
L^j_i &= \mathbf{1} + \sigma(L^1_{i-1}, \dots, L^{j-2}_{i-1}, L^j_{i-1}, \dots, L^k_{i-1}) + \mathbf{1},\\
&\ \ \vdots\\
L^{k+1}_i &= \mathbf{1} + \sigma(L^1_{i-1}, \dots, L^{k-1}_{i-1}) + \mathbf{1}
\end{aligned}$$

and define

$$\begin{aligned}
L^-_i &= \mathbf{1} + \sigma(L^1_i, L^2_i, \dots, L^k_i) + \mathbf{1}, \text{ and}\\
L^+_i &= \mathbf{1} + \sigma(L^1_i, L^2_i, \dots, L^k_i, L^{k+1}_i) + \mathbf{1}.
\end{aligned}$$

The number k will depend on how many switches have been made from L^-_* to L^+_*; k may be as few as 3 and as many as $i + 3$. For each $j \geq 2$, L^j_i has exactly one L^*_{i-1} missing, namely, L^{j-1}_{i-1}. Notice, however, that a shuffle of any two, or more, of the L^*_i's will contain every one of the L^*_{i-1}'s. It follows that both L^-_i and L^+_i (and hence each L^*_{i+1}) will have all the L^*_{i-1}'s appearing, and doing so in whatever order one may desire. This allows us to establish the following two facts.

Fact 4.1 *For each L^j_i with $j \geq 2$ (we need this for just the last two L^j_i's, but it is true in general), there is a Π_{i+2} sentence φ^j_i that is true of L^j_i but not of any of the other L^*_i's, nor of any shuffle of (shuffles, of shuffles, of . . .) these other L^*_i's.*

Proof: By induction on i.

Base step ($i = 1$): take φ^j_1 to be the Π_3 sentence that says there is no "maximal block" of size j (i.e., j consecutive elements, the first of which has no immediate predecessor, the last of which has no immediate successor).

Inductive step: take φ^j_i to be the sentence that says that, for each x and y, the interval $[x, y]$ is not isomorphic to L^{j-1}_{i-1}. In order to write this as a Π_{i+2} sentence, take the Π_{i+1} sentence φ^{j-1}_{i-1} true of L^{j-1}_{i-1} but not of the other L^*_{i-1}'s, a sentence whose existence is guaranteed by the inductive hypothesis, negate it, and change the first existential quantifier from "there are elements $x_1, \dots, x_n$ such that . . . " to "there are elements $x_1, \dots, x_n$ *between x and y* such that . . . " We can then replace the phrase "the interval $[x, y]$ is not isomorphic to L^{j-1}_{i-1}" with this Σ_{i+1} formula to get the Π_{i+2} sentence we desire.

Since this sentence speaks of the nonexistence, in L^j_i, of an interval of a certain type, the fact that it is false in every one of the other L^*_i's will make it false in every shuffle of these other L^*_i's, and in every shuffle of those shuffles, and so on. $\square$

Consider now the sentence saying that, for each pair x, y, the interval $[x, y]$ is not isomorphic to L_i^{k+1}. Replace, as before, the phrase "the interval $[x, y]$ is not isomorphic to L_i^{k+1}" with the similarly modified version of the negation of the sentence φ_i^{k+1}. This produces a Π_{i+3} sentence true of L_i^- but not of L_i^+, the promised sentence ψ_i, that distinguishes between $\mathfrak{L}$ with L_i^- chosen for L_i, and $\mathfrak{L}$ with L_i^+ for L_i.

Fact 4.2 *The L_i^*'s are all $\equiv_i$ to each other, and to every shuffle of (shuffles, of shuffles, of . . .) L_i^*'s.*

Proof: By induction on i.

Base step ($i = 1$): the L_1^*'s are all infinite linear orders, as is every shuffle produced from them; consequently **P2** can easily win every 1-k **E-F Game**.

Inductive step: consider **P1**'s opening move in an i-k **E-F Game** played on any two L_i^*'s (or shuffles, of shuffles, of . . . L_i^*'s). Consider the L_{i-2}^*'s in which the elements of **P1**'s sequence lie; as we have seen, the L_{i-2}^*'s all appear in every L_i^*, and do so in every possible order (with the possible exception of the very last L_{i-2}^* which, if we are presently working with L_{i-2}^-, appears nowhere). So **P2** can pick a matching sequence whose elements lie in identical L_{i-2}^*'s, and in identical positions in those L_{i-2}^*'s, as do the elements of **P1**'s sequence. To show that this is a winning move we need to show that **P2** can match **P1** for the remaining $i-1$ moves of the game.

Consider now the interval (a, b) between any pair a, b of elements in **P2**'s sequence. If a and b lie in the same copy of some L_{i-2}^*, **P2**'s strategy guarantees that the interval (a, b) will be isomorphic to the corresponding interval in **P1**'s sequence. Otherwise, if a and b lie in separate L_{i-2}^*'s, the interval (a, b) will have order type $\alpha + \beta + \gamma$, where α will be the left-open, right-closed interval which is the end of a's L_{i-2}^*, γ the left-closed, right-open interval which is the beginning of b's L_{i-2}^*, both of which will be isomorphic to the corresponding parts of the corresponding interval for **P1**'s sequence, and β will have order type $\beta_1 + \beta_2 + \beta_3$, where β_1 is the tail end of a's L_{i-1}^* (and hence an L_{i-1}^* itself, with the left endpoint removed), β_3 is the front end of b's L_{i-1}^* (and hence an L_{i-1}^* itself, with the right endpoint removed), and β_2 is a shuffle of L_{i-1}^*'s. This will also be true of the corresponding interval for **P1**'s sequence; that is, that sequence also will be of the form $\alpha + \beta_1 + \beta_2 + \beta_3 + \gamma$, with the five summands as described. Since the two α's and the two γ's will be isomorphic, and for each j, the two β_j's will, by the inductive hypothesis, be $\equiv_{i-1}$, it follows that **P2** can beat **P1** at every $(i-1)$-k **E-F Game** played within these two intervals. □

This, and the fact that each L_i has a decidable copy, allows us to show that $\mathfrak{L}$ is i-decidable for each i. Consider any sequence $\bar{a}$ in $\mathfrak{L}$. Wait for a stage $j \geq i$ by which $\bar{a}$ has been enumerated into the construction and lies wholly within some copy of L_j, and further still, for a stage when $L_1, \ldots, L_j$ have all settled down (between being L_*^- or L_*^+). This latter stage cannot be recognized during the construction (this is where the nonuniformity comes in: our algorithm for i is predicated on guessing such a stage). Our construction allows us to determine exactly how the elements of $\bar{a}$ lie within their copies of L_i^*'s. Find a sequence $\bar{b}$ in the decidable copy of L_i with its elements in the same situation with respect to L_i^*'s (all existing L_i^*'s occur in L_i, in every possible order). The intervals between the copies of L_i^*'s in L_i are shuffles of L_i^*'s, and in L_j, and $\mathfrak{L}$, are shuffles, of shuffles, of . . . L_i^*'s, and therefore, by Fact 4.2, are all $\equiv_i$. Consequently $\bar{a}$ satisfies precisely the same Σ_i formulas in L_j, and in $\mathfrak{L}$, as does $\bar{b}$ in its copy of L_i. Since that copy is decidable, we have provided a computable procedure that determines exactly which Σ_n formulas $\bar{a}$ satisfies in $\mathfrak{L}$.

5 *Construction of* $\mathfrak{L}$ The construction of $\mathfrak{L}$ is a standard, finite-injury, priority construction.

Stage 1: Begin constructing a copy of L_1^- as described before.

Stage s: Look for the least $e < s$ that *requires attention* and *may be addressed* at this stage. (An e requires attention if it has never been addressed or if it has been *injured* since it was last addressed. It may be addressed at this stage if the eth computably enumerable function has shown its hand on ψ_e and says that that sentence is true in $\mathfrak{L}$.) Address e by changing L_e from L_e^- to L_e^+. This will also change all the L_i^*'s with $i > e$ and consequently change all those L_i, and their distinguishing sentences ψ_i. Consider all those i to be injured at this stage. Continue the construction of (the present versions) of $L_1, \ldots, L_{s-1}$ and begin the construction of L_s^- around this, as described before.

By the argument presented in the paragraph following the proof of Fact 4.2, the linear order $\mathfrak{L}$ so produced is n-decidable for every n. It has no decidable copy since the set of sentences true in $\mathfrak{L}$ is not computable: the eth computably enumerable function could not possibly denote exactly which sentences were true in $\mathfrak{L}$ since, if it were the first computably enumerable function on the list to do so, there would come a stage in the construction after which none of the earlier computably enumerable functions are ever addressed (and hence ψ_e would never change), when e would both require attention and may be addressed, and consequently would be, thus causing it to be in error on the truth in $\mathfrak{L}$ of the sentence ψ_e. This completes the proof of our theorem.

6 *Intrinsically n-decidable* It should be noted that the linear order we have constructed does have computable copies that are *not* n-decidable for all n, in fact, computable copies that are not even 1-decidable. It follows from the characterization of *intrinsically* 1-decidable linear orders (i.e., 1-decidable linear orders all of whose computable copies are also 1-decidable) in Moses [5] that every such linear order is decidable. So there is no linear order that is n-decidable for all n and intrinsically n-decidable for all n but has no decidable copy.

Consider, however, the language of linear order expanded by adding a constant symbol for each element of the $\mathfrak{L}$ we constructed and the structure $\mathcal{M}$ in this language produced from $\mathfrak{L}$ by interpreting each constant symbol by the corresponding element. It is clear from our construction of $\mathfrak{L}$ that $\mathrm{Th}(\mathcal{M})$ is noncomputable whereas for each n, the set of Σ_n sentences true in $\mathcal{M}$ *is* computable. It follows that $\mathcal{M}$ has no decidable copy but is n-decidable for all n, *and intrinsically so* (every element in a computable copy of $\mathcal{M}$ will be a constant and hence, for every Σ_n formula φ and sequence $\bar{a}$ in that copy, $\varphi(\bar{a})$ will be a Σ_n *sentence* in the language). We have established the following.

Corollary 6.1 *There is a structure that is n-decidable and intrinsically n-decidable for all n but has no decidable copy.*

We do not know whether there is a more natural structure with this property; Chisholm can show that there is no tree.

Acknowledgments The second author was supported by a George Washington University Facilitating Fund Award.

REFERENCES

[1] Downey, R., "Computability theory and linear orderings," pp. 823–976 in *Handbook of Recursive Mathematics*, vol. 2, edited by Yu. L. Ershov, S. S. Goncharov, A. Nerode, J. B. Remmel, and V. W. Marek, Elsevier, New York, 1998. Zbl 0941.03045 MR 2000j:03059 1

[2] Ehrenfeucht, A., "An application of games to the completeness problem for formalized theories," *Fundamenta Mathematicæ*, vol. 49 (1961), pp. 129–41. Zbl 0096.24303 MR 23:A3666 2

[3] Fraïssé, R., "Sur quelques classifications des systèmes de relations," *Publications in Science and Universal Algebra*, series A, vol. 1 (1954), pp. 35–182. Zbl 0068.24302 MR 16,1005h 2

[4] Fröhlich, A., and J. C. Shepherdson, "Effective procedures in field theory," *Philosophical Transactions of the Royal Society of London*, series A, vol. 248 (1955), pp. 407–32. Zbl 0070.03502 MR 17,570d 1

[5] Moses, M., "Relations intrinsically recursive in linear orders," *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, vol. 32 (1986), pp. 467–72. Zbl 0596.03038 MR 87k:03047 6

[6] Moses, M., "n–Recursive linear orders without $(n+1)$-recursive copies," pp. 572–92 in *Logical Methods: in Honor of Anil Nerode's Sixtieth Birthday*, edited by J. N. Crossley, J. B. Remmel, R. A. Shore, and M. E. Sweedler, Birkhäuser, Boston, 1993. Zbl 0824.03020 MR 95i:03098 1

[7] Rabin, M., "Computable algebra, general theory and theory of computable fields," *Transactions of the American Mathematical Society*, vol. 95 (1960), pp. 341–60. Zbl 0156.01201 MR 22:4639 1

[8] Rosenstein, J. G., *Linear Orderings*, Academic Press, New York, 1982. Zbl 0488.04002 MR 84m:06001 1, 1, 2

Department of Mathematics
Western Illinois University
Macomb, Illinois 61455
email: JA-Chisholm@wiu.edu

Department of Mathematics
The George Washington University
Washington, D.C. 20052
email: moses@gwu.edu