# THE COMPLEXITY OF RECURSION THEORETIC GAMES

MARTIN KUMMER

*Wir meinen, das Märchen und das Spiel gehöre zur Kindheit: wir Kurzsichtigen! Als ob wir in irgendeinem Lebensalter ohne Märchen und Spiel leben möchten!* Friedrich Nietzsche

ABSTRACT. We show that some natural games introduced by Lachlan in 1970 as a model of recursion theoretic constructions are undecidable, contrary to what was previously conjectured. Several consequences are pointed out; for instance, the set of all $\Pi_2$-sentences that are uniformly valid in the lattice of recursively enumerable sets is undecidable. Furthermore we show that these games are equivalent to natural subclasses of effectively presented Borel games.

## 1. INTRODUCTION

Games are ubiquitous. The reader may first think of the ancient games of Chess and Go, but the concept is much more general. It has been argued that games are constitutive for man who is considered as a *homo ludens* by Huizinga in [14]. We may remark that this thesis is more convincing in Latin (or e.g. Dutch and German) than in English because there one word combines the meaning of the two English words *play* and *game*. Games are not confined to humanity, animals also play games, and in [9] it is shown that games already appear in natural history, in particular in the evolution of macromolecules. An 'evolutionary game theory' was developed in biology [29]. Games play a fundamental role in the fine arts and in literature. They are studied in sociology [39], in philosophy, in particular in Wittgenstein's philosophy of language [42], and even in theology, where the concept of a *Deus ludens* was proposed [36].

In mathematics, games are studied in the 'theory of games' founded in [32] (see also [31] for an interesting account of its early history) with important applications in economy. A well-known game in this context is Prisoner's Dilemma. There is also a combinatorial game theory with roots in recreational mathematics [1, 24]. A paradigmatic game in this field is Nim. For many combinatorial games a complete analysis can be given [3]. On the other hand, some combinatorial games are apparently very hard and defy an underlying theory. For them there seems to be no simpler way to determine which player has a winning strategy than to enumerate all possible moves in an exhaustive search.

In theoretical computer science, finite games are studied in complexity theory, as just mentioned. Here also new games have been invented like e.g. 'interactive proofs' [34]. Infinite games are studied in automata theory as e.g. the regular games of [5]; see [18] for a presentation in textbook form. They have found applications in hardware and software design and provide powerful decision procedures of certain logics; see [11] for an up-to-date survey. Another field where infinite games appear is the study of online algorithms [4].

In logic, games also abound. In model theory, we have the finite Ehrenfeucht-Fraissé games and also infinite games used to build certain structures in infinitely many steps [13]. In set theory, one studies the axiom of determinacy (see [2], Chap. B.2, §10) and in descriptive set theory, weaker versions of determinacy axioms are considered (see [2], Chap. C.8, §6). For applications of determinacy in recursion theory cf. [2], Chap. C.4, §4, and [23], Chap. V.5. In recursion theory many arguments can be viewed as providing effective winning strategies in certain infinite games. This point of view was first stressed by Lachlan [22] and was later strongly advocated and made popular in the work of Soare. By now the game paradigm is widely used in recursion theory as a visualization and heuristic.

There are easy examples of effectively presented games where none of the players has a recursive winning strategy, e.g. an example of Rabin using a simple set [35] (also in abbreviated form in [37], Ex. 8-5, p. 121 f.). Later Jones [16] gave more explicit examples making use of the undecidability of Hilbert's tenth problem.

In the present paper we study games proposed by Lachlan [22] where both players are enumerating finitely many sets and in the end, after infinitely many rounds, player 1 wins iff the enumerated sets satisfy a specified formula, the winning condition. As Lachlan pointed out, these games are the basic building blocks of constructions of r.e. sets. They refer to a single requirement of a construction that usually has to satisfy an infinite sequence of requirements. Based on the analysis of many concrete examples, Lachlan conjectured that for a single requirement it is decidable which player has a winning strategy. He proved a strong partial result that pointed in this direction. Later in [19] another attempt was made with the idea of reducing these games to the regular games mentioned above. But this approach fell short of a decidability proof, too.

We show here that, somewhat surprisingly, the problem is undecidable, even with a rather high degree of unsolvability. We get interesting consequences on the (non-)existence of recursive winning strategies, and we get that the set of all $\Pi_2$-sentences uniformly valid in the lattice of r.e. sets is undecidable. Our undecidability proof resembles arguments from complexity theory used to show that certain finite combinatorial games are PSPACE-hard [10, 38]. We also classify the complexity of the problem as $\Delta_2^1$ in the analytical hierarchy. Finally we characterize these games in terms of effectively presented Borel games, and it turns out that they have maximal possible complexity.

For general background on logic we refer the reader to [2], for the more specific recursion theory background see [37, 41].

## 2. The basic results

The following game was introduced by Lachlan [22]. There it was called *basic game of the second kind.* Here we call it simply basic game.

**Definition 2.1.** Let $n \geq 1$ and $F$ be any quantifier-free first-order formula without equality and with variables $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$, the operations union, intersection and complement, and the unary predicate isEmpty$(x)$.

A *basic game* $G = G_F$ is defined as follows: Two players are enumerating alternately, starting with player 1, in rounds, natural numbers into some sets $U_1, \ldots, U_n$ and $V_1, \ldots, V_n$, respectively. Each player may enumerate finitely many elements in each round (in particular, he is also allowed to enumerate nothing, i.e. to pass at his turn). Player 1 enumerates the $U_i$'s and player 2 enumerates the $V_i$'s.

$F$ specifies the winning condition in the following sense: After $\omega$ rounds of $G$, player 1 wins iff the formula $F$ is satisfied by the canonical interpretation with the enumerated sets.

*Remark* 2.2. Note that the order in which the players play is not essential, because, as the referee stressed, the action of one player does not prevent the action of the other player. Thus, if player 2 has a winning strategy in $G_F$ with $F = F(u_1, \ldots, u_n, v_1, \ldots, v_n)$, then player 1 has a winning strategy in $G_{F'}$ with $F' = \neg F(v_1, \ldots, v_n, u_1, \ldots, u_n)$. This is used later on.

Also, though we do not use it, we could restrict one (or both) players to enumerate at most one number in each round. This does not change the outcome, since a player could append his original moves to a list of moves and enumerate in each round the first element of the list (which is then removed from the list).

**Example 2.3.** a) Let us consider as an easy but instructive example the construction of a creative set $K$, cast in the language of basic games. The task is specified as follows: One has to enumerate a set $K$ and sets $U_i$ such that for all $i \in \omega$ the following requirements $R_i$ are satisfied (where $W_i$ is the $i$-th r.e. set):[1]

$$R_i: \ K \cap W_i = \emptyset \ \rightarrow \ U_i \neq \emptyset \ \wedge \ U_i \cap (K \cup W_i) = \emptyset.$$

This can be considered as an infinite game where player 1 enumerates $K$ und the $U_i$'s while player 2 enumerates the $W_i$'s. We need to show that player 1 has a recursive winning strategy.

The usual way to do this is to find a strategy for a single requirement and then to combine the single strategies in a suitable way to play all of them. This is simple in our examples but may in general require more sophisticated devices, like the priority method.

Here a single requirement is specified by the following formula $F$:

$$\text{isEmpty}(u_1 \cap v_1) \rightarrow \neg\text{isEmpty}(u_2) \ \wedge \ \text{isEmpty}(u_2 \cap (u_1 \cup v_1)).$$

In the basic game $G_F$ player 1 has the following winning strategy: In round 0 he enumerates 0 into $U_2$, which momentarily satisfies $F$, and then waits to see what player 2 is doing. If at some round player 2 enumerates 0 into $V_1$, he enumerates 0 into $U_1$ which satisfies $F$ permanently.

Note that this strategy of player 1 is recursive, and the number 0 can be replaced by any number. Therefore one can play infinitely many strategies simultaneously, and this provides the desired winning strategy for the original game.

b) We can push the previous example somewhat further by devising a basic game for a proof of Myhill's Theorem that every creative set is $m$-complete [37], §11.3,

---

[1]Here as well as in the rest of the paper we assume as usual that $\neg$ is more binding than $\wedge$ and $\vee$, which in turn are more binding than $\rightarrow$ and $\leftrightarrow$.

Theorem V. To this end we have the following requirements $R_i$:

$$R_i : [K \cap U_{2i} = \emptyset \;\rightarrow\; V_{2i} \neq \emptyset \;\wedge\; V_{2i} \cap (K \cup U_{2i}) = \emptyset]$$
$$\wedge\; |V_{2i+1}| = 1 \rightarrow$$
$$|U_{2i+1}| = 1 \;\wedge\; (V_{2i+1} \subseteq B \leftrightarrow\; U_{2i+1} \subseteq K).$$

The $U_i$'s belong to player 1; $B, K$ and the $V_i$'s belong to player 2. Using Lemma 3.2 below we can express these requirements in the language of Definition 2.1 (there is a slight twist in picking the right replacement for the subformula $|V_{2i+1}| = 1$).

Player 1 has the following winning strategy for $R_i$: Wait until player 2 enumerates a number $x_i$ into $V_{2i}$ and a number $y_i$ into $V_{2i+1}$. Then enumerate $x_i$ into $U_{2i+1}$ and wait until $y_i$ is enumerated into $B$. If this ever happens enumerate $x_i$ into $U_{2i}$. This satisfies $R_i$: If player 2 does not enumerate $x_i$ into $K$, $R_i$ holds, since the first conjunct of the hypothesis of $R_i$ is false. If player 2 enumerates $x_i$ into $K$, $R_i$ holds, since the conclusion of $R_i$ holds.

Since the different strategies do not interfere, they can be played simultaneously such that all $R_i$ are satisfied. Note that each single strategy as well as the combined one is recursive. Using this and the recursion theorem we get for every creative set $K$ and every r.e. set $B$ a recursive $m$-reduction which reduces $B$ to $K$, i.e., Myhill's Theorem follows.

Note that the game approach lead to stronger results since we obtained recursive strategies that win against *any* strategy of player 2, not just recursive ones (which would be sufficient to prove the result). This feature also holds for all examples in [22], Section 2. It does however not generalize to arbitrary basic games; see Remark 4.1 below.

In the examples above we slightly abused the original intention of Lachlan, who conceived the basic games for proving results about the lattice of r.e. sets. In fact, most[2] of the well-known theorems of the lattice of r.e. sets, as e.g. Friedberg's Splitting Theorem, can be proved by providing winning strategies in suitable basic games using the additional predicate isFinite$(x)$ to formulate the requirements. Many examples can be found in [22].

An important question of the metamathematics of recursion theory is whether there exists an effective procedure to tell us whether player 1 has a winning strategy for any given single requirement. Formally we ask if the following decision problem $P$ of Lachlan [22], pp. 303, 309, is decidable.

**Problem 2.4.** $P$: Decide for any given formula $F$ whether player 1 has a winning strategy in the basic game $G_F$.

Lachlan conjectured in [22] that $P$ is decidable (even for the extended language with the additional predicate isFinite$(x)$, the 'mixed basic games', which we consider in Section 6). The following result refutes this conjecture.

**Theorem 2.5.** *Problem $P$ is undecidable.*

From the proof of the theorem we get a number of interesting consequences.

**Corollary 2.6.**     (1) *The complexity of $P$ is at least as high as the complexity of True Arithmetic; in particular it is non-arithmetical.*

---

[2]Lachlan [22], p. 293, provided a formalization of 'game derivability'. It is apparently still open as to whether every sentence of the theory of the lattice of r.e. sets is game derivable.

(2) $P$ is also undecidable if we are asking for a recursive *winning strategy for player* 1.

(3) $P$ is also undecidable if player 1 and player 2 are restricted to recursive strategies.

(4) For any $n$ there are effectively computable formulas $F$ such that neither player has a winning strategy in $G_F$ that is recursive in $\mathbf{0}^{(n)}$.

(5) The set of all sentences of the $\Pi_2$-theory of the lattice of r.e. sets (in the language with equality, operations $\cap, \cup$ and constants $0, 1$) that are uniformly valid[3] is undecidable. This holds even if quantification is restricted to finite sets and r.e. indices are used for the uniformity (with respect to recursive indices the problem is decidable).

Lachlan proved in [22], Section 3, that the related problem $P^*$, where the predicate isEmpty($x$) is replaced by isFinite($x$) (so-called 'basic games of the first kind'), is in fact decidable with uniformly recursive winning strategies. This is in sharp contrast to Corollary 2.6, (1), (4).

Corollary 2.6, (5) contrasts Lachlan's well-known result that the $\Pi_2$-theory of the lattice of r.e. sets is decidable (see [20, 21] and the overview in [41], Chap. XVI, 2).

## 3. The basic proofs

Let us first sketch the outline of the proof of Theorem 2.5 and later provide the details of the nontrivial steps.

We reduce Hilbert's tenth problem to $P$, i.e., for each diophantine equation $p_1 = p_2$, where $p_1, p_2$ are polynomials (in several variables) with nonnegative integer coefficients, we effectively specify a formula $F$ such that the equation has a nonnegative integer solution iff player 1 has a winning strategy in $G_F$.

In [28] the celebrated result was shown that Hilbert's tenth problem is undecidable (see [2], Chap. C.2, §5) and that for every r.e. relation $R \subseteq \omega^k$ there is a diophantine equation $p_1 = p_2$ with variables $x_1, \ldots, x_k$ and $y_1, \ldots, y_n$ such that for every $(a_1, \ldots, a_k) \in \omega^k$,

$$(a_1, \ldots, a_k) \in R \iff \text{there exist } (b_1, \ldots, b_n) \in \omega^n \text{ such that } p_1 = p_2$$
$$\text{for } x_i = a_i, i = 1, \ldots, k \text{ and } y_j = b_j, j = 1, \ldots, n.$$

Furthermore, the diophantine equation can be found effectively in any r.e. index of $R$.

*Remark* 3.1. We could also directly reduce True Arithmetic (TA) to $P$, but this would be less perspicuous. However, it should be noted that essentially the same proof also works using the undecidability of TA instead of the more sophisticated undecidability of Hilbert's tenth problem, which by the way was not available when [22] was written.

The basic idea of our reduction may be best illustrated by an example. Suppose we are given the equation $y_1 * y_2 + 5 = y_3 + y_1 * y_4$. Then player 1 enumerates finite

---

[3] A $\Pi_2$-sentence is called uniformly valid [19] if for any given indices of r.e. sets to be substituted for the universally quantified variables one can uniformly compute indices of r.e. sets to be substituted for the existentially quantified variables such that the matrix holds with these substitutions. Any uniformly valid sentence is valid, but not conversely. An easy counterexample is the valid sentence $\forall x \exists y [x \neq y]$ whose uniform version is the negation of the recursion theorem.

sets $M_1, \ldots, M_5, N_1, \ldots, N_4$ and claims that:

(1) $M_3, M_4$ are pairwise disjoint and $N_1, N_3$ are pairwise disjoint.
(2) $|M_3| = |M_1| * |M_2|$.
(3) $|M_4| = 5$.
(4) $M_5 = M_3 \cup M_4$.
(5) $|N_3| = |M_1| * |N_2|$.
(6) $N_4 = N_1 \cup N_3$.
(7) $M_5 = N_4$.

Claims (1), (4), (6), and (7) can be directly put into the winning condition. For claim (3) it is not difficult to devise a subgame specified by a formula $\text{Equal}_5(M_4, \ldots)$ that player 1 can win iff this claim is satisfied.

Player 2 may then challenge one of the claims (2), (3), and (5), and he should win iff the challenged claim turns out to be false. This game can be reduced to a basic game (where of course additional sets are used).

How can player 2 be sure that player 1 does not cheat by later enumerating additional elements into his sets? This can be done using the powerful idea of 'double-checking' (or 'committing', as Marcus Schaefer suggested): Suppose player 1 is to enumerate a finite set $M$. Then player 2 enumerates a subset $A$ of $M$. The enumeration phase is ended by player 1 when he makes a signaling set $S$ nonempty. While $S$ is empty he wins if $A$ does not equal $M$. In this way player 1 can force $A = M$. This is done by the following formula $F$:

$$A \not\subseteq M \ \lor \ (A \neq M \ \land \ S = \emptyset).$$

The basic idea (which is modified later on) is that this formula appears as a disjunction in the winning condition so that player 1 immediately wins if player 2 does not play as intended. The crucial point is that later we work not with $M$ but with the set $A$ instead. Note that each player can freeze $A$ if it is in his interest.

From the point of view of player 2, he can force that the enumerated double-checked set $A$ is *finite* unless the winning condition $F \ \lor \ (S \neq \emptyset \ \land \ \ldots)$ is falsified: If he makes $A = M$ as long as $S = \emptyset$, player 1 is forced to eventually make $S \neq \emptyset$ (otherwise the winning condition is false), and then the set $A$ is finite and we even have a canonical index for it.

The hard part of the reduction is to show that there is a formula

$$\text{Mult}(M, A, B, \ldots)$$

such that for any canonically given fixed finite sets $M, A, B$, player 1 has a winning strategy in $G_{\text{Mult}}$ iff $|M| = |A| * |B|$. Note that these are not basic games in the strict sense of Definition 2.1 but rather basic games with parameters. For ease of presentation we do not stress this distinction in the following and also speak of these modified basic games simply as basic games.

To get an idea of how this is done the reader may first devise a corresponding formula $\text{Greater}(A, B, \ldots)$ for the condition $|A| > |B|$ that is used in building Mult. In fact, this was the way in which the author discovered the reduction.

The formula Mult will implement the following game $G'$ where player 1 enumerates $N$ and player 2 enumerates $L$:

(1) Player 2 enumerates a new element of $A$ into a test set $L \subseteq A$.
(2) Player 1 enumerates new elements of $M$ into a test set $N \subseteq M$.

(3) Player 2 wins instantly if player 1 did not enumerate exactly $|B|$ new elements.

(4) Player 2 may start a new stage. Then the game continues with step 1.

Player 2 wins iff he wins in step (3) or at the end $L = A$ and $N \neq M$. The set $L$ serves as a 'tally set' or 'loop counter'. Note that if $|M| = |A| * |B|$, then player 1 has a winning strategy else player 2 has one. To actually realize this game in Mult we are relying heavily on double-checking.

Now we can write down the winning condition $F$ for player 1 in the example above:

$$
\begin{aligned}
F \;=\; & M_3 \cap M_4 = \emptyset \;\wedge\; N_1 \cap N_3 = \emptyset \\
& \wedge\; M_5 = M_3 \cup M_4 \;\wedge\; N_4 = N_1 \cup N_3 \;\wedge\; M_5 = N_4 \\
& \wedge\; F',
\end{aligned}
$$

where $F'$ is the following formula:

$$
\begin{aligned}
[ \quad & \bigvee_{i=1}^{5}(A_i \not\subseteq M_i \;\vee\; (M_i \neq A_i \;\wedge\; S = \emptyset)) \\
\vee\; & \bigvee_{i=1}^{4}(B_i \not\subseteq N_i \;\vee\; (N_i \neq B_i \;\wedge\; S = \emptyset)) \\
\vee\; & (S \neq \emptyset \;\wedge\; T_1 \cup T_2 \cup T_3 = \emptyset) \\
\vee\; & (T_1 \neq \emptyset \;\wedge\; \mathrm{Equal}_5(A_4, \ldots)) \\
\vee\; & (T_2 \neq \emptyset \;\wedge\; \mathrm{Mult}(A_3, A_1, A_2, \ldots)) \\
\vee\; & (T_3 \neq \emptyset \;\wedge\; \mathrm{Mult}(B_3, A_1, B_2, \ldots)) \quad ].
\end{aligned}
$$

For each subformula $\mathrm{Equal}_5$ or Mult the variables indicated by the dots are new variables that occur only in the corresponding subformula.

Now suppose that the diophantine equation has a solution. Then player 1 has the following winning strategy. He enumerates the finite sets $M_i, N_j$ such that all claims are satisfied, where their size is chosen according to the values of the $y_k$'s in the solution. Then he waits, leaving $S$ empty until player 2 has enumerated the $A_i, B_j$ such that $A_i = M_i, B_j = N_j$ for all $i, j$. If this is never the case, player 1 wins by the first part of $F'$. Also by the first part of $F'$ he can force that $A_i, B_j$ are fixed in the following by fixing $M_i, N_j$. Now player 1 enumerates an element into $S$, thereby forcing player 2 to enumerate an element into some $T_k$. Then player 1 applies his winning strategy to the selected formula $\mathrm{Equal}_5$ or Mult and wins the game.

If the diophantine equation has no solution, then player 2 has a winning strategy. He makes $A_i = M_i$ and $B_j = N_j$ for all $i, j$ until player 1 enumerates an element into $S$. If this never happens, he wins. So suppose that player 1 makes $S$ nonempty. Let $X$ be the union of all $A_i$ and $B_j$. Player 2 waits until the first part of $F$ is satisfied if all $M_i, N_j$ are restricted to $X$. If this never happens, he wins. If it happens, he enumerates, if necessary, additional elements into $A_i, B_j$ in order to satisfy $A_i = M_i \cap X$ and $B_j = N_j \cap X$ for all $i, j$. Since the diophantine equation has no solution, one of the claims (1)–(7), with $M_i$ replaced by $A_i$ and $N_j$ replaced by $B_j$, is false. So far we know that claims (1), (4), (6), and (7) are true. Therefore one of the claims (2), (3), or (5) is false. Player 2 determines which one it is, makes

the corresponding $T_k$ nonempty, and plays such that the corresponding formula $\text{Equal}_5$ or Mult is not satisfied. This makes $F$ false, so player 2 wins.

Note that one could dispense with the $T$-variables, but this would make the game more complicated, since then player 1 would have to play all subgames on $\text{Equal}_k$ and Mult simultaneously.

This finishes the outline of the proof. We can leave it to the reader to generalize the above construction to arbitrary diophantine equations.

It remains to show that there are formulas $\text{Equal}_k$, Greater and Mult as required. This is done by the following lemmas. For ease of presentation we call a basic game *effectively determined* [19] if one of the players has a recursive winning strategy.

**Lemma 3.2.** *For every $k \geq 0$ there is a formula $\text{Equal}_k(M, U_1, \ldots, U_k, V_1, V_2)$ such that for every canonically given finite set $M$ the corresponding basic game where player 1 enumerates the $U_i$'s and player 2 enumerates the $V_j$'s is effectively determined, and player 1 has a winning strategy iff $|M| = k$.*

*Proof.* We define $\text{Equal}_k(M, U_1, \ldots, U_k, V_1, V_2)$ as follows:

$$\bigwedge_{1 \leq i < j \leq k} U_i \cap U_j = \emptyset \ \wedge \ \bigwedge_{i=1}^{k} U_i \neq \emptyset \ \wedge \ U_1 \cup \ldots \cup U_k = M$$

$$\wedge \neg [V_1 \cap V_2 = \emptyset \ \wedge V_1 \neq \emptyset \ \wedge \ V_2 \neq \emptyset \ \wedge \bigvee_{i=1}^{k} V_1 \cup V_2 \subseteq U_i].$$

The verification is left to the reader. For $k = 0$ the formula simplifies to $M = \emptyset$. $\square$

**Lemma 3.3.** *There is a formula $\text{Greater}(A, B, U_1, U_2, V_1, V_2)$ such that for any canonically given finite sets $A, B$ the corresponding basic game where player 1 enumerates the $U_i$'s and player 2 enumerates the $V_j$'s is effectively determined, and player 1 has a winning strategy iff $|A| > |B|$.*

*Proof.* We define $\text{Greater}(A, B, U_1, U_2, V_1, V_2)$ as follows ($\oplus$ denotes the exclusive or):

$$[U_1 \subseteq U_2 \subseteq A \ \wedge \ ((U_1 = U_2) \oplus (V_1 = V_2))] \ \vee \ \neg(V_1 \subseteq V_2 \subseteq B).$$

Suppose that $|A| > |B|$; then player 1 has the following winning strategy. It proceeds in at most $|B|+1$ stages. At the beginning of each stage we have $U_2 - U_1 = \emptyset = V_2 - V_1$. Player 1 enumerates a new element $a$ from $A$ into $U_2$. This makes $U_2 - U_1$ nonempty and forces player 2 to enumerate at least one new element from $B$ into $V_2 - V_1$. If he does not do this he loses. If he does it, player 1 enumerates $a$ into $U_1$ which forces player 2 to enumerate $V_2 - V_1$ into $V_1$. This finishes the stage. After at most $|B|$ stages the elements of $B$ are exhausted, and player 1 wins by making $U_2 - U_1$ nonempty while $V_2 - V_1$ is empty.

If $|A| \leq |B|$, then player 2 can counter each move that makes $U_2 - U_1$ nonempty by enumerating a new element of $B$ into $V_2$. If player 1 clears $U_2 - U_1$ by making it empty, then player 2 does the same with $V_2 - V_1$. Since $|A| \leq |B|$ player 2 will not run out of moves and will win the game. $\square$

*Remark 3.4.* The referee observed that the conditions $U_1 \subseteq U_2, V_1 \subseteq V_2$ could be replaced by the weaker conditions $U_1 \subseteq A, V_1 \subseteq B$, but this added freedom makes the proof slightly less perspicuous.

**Lemma 3.5.** *There is a formula* $\mathrm{Equal}(A, B, U_1, \ldots, U_4, V_1, \ldots, V_4)$ *such that for any canonically given finite sets* $A, B$, *the corresponding basic game in which player 1 enumerates the* $U_i$'s *and player 2 enumerates the* $V_j$'s *is effectively determined, and player 1 has a winning strategy iff* $|A| = |B|$.

*Proof.* The formula Equal is defined as follows using the formula Greater from the previous lemma:

$$\neg[\mathrm{Greater}(A, B, V_1, V_2, U_1, U_2) \ \vee \ \mathrm{Greater}(B, A, V_3, V_4, U_3, U_4)].$$

Note that the order of the $U_i$'s and $V_i$'s is reversed. Here we are using Remark 2.2. By Lemma 3.3, player 1 has a winning strategy iff not $|A| > |B|$ and not $|B| > |A|$, i.e. iff $|A| = |B|$. □

**Lemma 3.6.** *There is a formula* $\mathrm{Mult}(M, A, B, K_1, K_2, N_1, N, L_1, L, M_1, M_2, T)$ *such that for any canonically given finite sets* $M, A, B$ *the corresponding basic game where player 1 enumerates* $K_1, K_2, N_1, N$ *and player 2 enumerates the remaining sets is effectively determined, and player 1 has a winning strategy iff* $|M| = |A| * |B|$.

*Proof.* We are actually implementing the game $G'$ mentioned above using the additional sets for double-checking. $T$ is a signaling set of player 2 which indicates, if nonempty, that player 2 tries to win by step (3) in game $G'$.

The formula $\mathrm{Mult}(M, A, B, \ldots)$ is defined as follows:[4]

$$
\begin{aligned}
&(M = \emptyset \ \wedge \ (A = \emptyset \ \vee \ B = \emptyset)) \\
&\vee \ \neg(L_1 \subseteq L \subseteq A \ \wedge \ M_1 \subseteq N_1 \ \wedge \ M_2 \subseteq N) \\
(3.0) \quad &\vee \ |L - K_1| > 1 \\
&\vee \ [ \ M \neq \emptyset \ \wedge \ A \neq \emptyset \ \wedge \ B \neq \emptyset \ \wedge \\
&\quad N_1 \subseteq N \subseteq M \ \wedge \ K_1 \subseteq K_2 \subseteq L \ \wedge \\
(3.1) \quad &\quad (T = \emptyset \ \wedge \ K_2 \neq L \ \rightarrow \ M_2 \neq N) \ \wedge \\
(3.2) \quad &\quad (T \neq \emptyset \ \wedge \ K_2 = L \ \wedge \ L_1 \neq L \ \rightarrow |B| = |M_2 - M_1|) \ \wedge \\
(3.3) \quad &\quad (T = \emptyset \ \wedge \ K_2 = L = L_1 \ \rightarrow \ N_1 = N) \ \wedge \\
(3.4) \quad &\quad (T = \emptyset \ \wedge \ K_2 = L = L_1 \ \wedge \ M_1 = M_2 \ \rightarrow \ K_1 = L) \ \wedge \\
(3.5) \quad &\quad (K_1 = A \ \rightarrow \ M_2 = M) \ ].
\end{aligned}
$$

The intuition behind these 'rules' is the following:

Player 1 has to satisfy the invariant $K_1 \subseteq K_2 \subseteq L$ and $N_1 \subseteq N \subseteq M$. Likewise player 2 has to satisfy $L_1 \subseteq L \subseteq A$ and $M_1 \subseteq N_1$ and $M_2 \subseteq N$.

(0) Player 2 can enumerate at most one element of $A$ at a time into the loop counter $L$.

(1) Player 1 forces player 2 to enumerate all of $N$ into $M_2$.

(2) Player 2 may decide to verify that player 1 has enumerated exactly $|B|$ new elements into $N$. This is verified with the double-checked set $M_2 - M_1$ of player 2.

(3) Player 2 forces player 1 to enter all of $N$ into $N_1$.

(4) Player 2 can prepare to start a new round only if he enters all of $M_2$ into $M_1$.

---

[4]Writing the subformulas (3.1)–(3.5) as conditionals was suggested by Marcus Schaefer. The intuition is that the conditionals are 'rules' that player 1 must follow while player 2 tries to spoil this, i.e., to satisfy the hypothesis of a rule and falsify the conclusion.

(5) If player 2 has enumerated all of $A$, then there are no new elements in $M$ left.

The intuition behind $K_1, K_2$ and $L_1$ is that they are 'step counters'. They play the role of signaling sets which can now be recycled.

Note that in order to keep things simpler we have used some abbreviations which require additional sets when spelled out. For $|L - K_1| > 1$ we should write more explicitly

$$C_1 \neq \emptyset \ \wedge \ C_2 \neq \emptyset \ \wedge \ C_1 \cap C_2 = \emptyset \ \wedge \ C_1 \cup C_1 \subseteq L - K_1,$$

where $C_1, C_2$ are additional sets of player 1. For $|B| = |M_2 - M_1|$ we should write more explicitly the formula

$$\text{Equal}(B, M_2 - M_1, U_1, \ldots, U_4, V_1, \ldots, V_4),$$

where the $U_i$'s belong to player 1 and the $V_i$'s belong to player 2.

a) Suppose $|M| \neq |A| * |B|$; then player 2 has a recursive winning strategy. We may assume that $M, A, B$ are nonempty, for otherwise player 2 wins by just doing nothing. Furthermore, we may assume that $N \subseteq M$ and $K_2 \subseteq L$ always; for otherwise, player 2 wins by doing nothing more (since the strategy we describe always falsifies the first three disjunctions of the formula). We can also assume that $N_1 \subseteq N$ and $K_1 \subseteq K_2$ always, since player 2 can simply wait until these conditions are satisfied.

The strategy of player 2 will always satisfy the conditions $L_1 \subseteq L \subseteq A$, $M_2 \subseteq N$, and $M_1 \subseteq N_1$.

Player 2 performs his strategy in stages as follows. When he starts stage $s \geq 0$ he ensures the invariant

$$R_s : s = |L|, |M_2| = |L| * |B|, K_1 = K_2 = L, M_1 = M_2 \subseteq N_1 \subseteq N.$$

Then player 2 enumerates a new element $a$ of $A$ into $L$. As long as $K_2 \neq L$, he makes $M_2 = N$ and wins by (3.1) if player 1 does not eventually enumerate $a$ into $K_2$. So, if this happens he checks whether $|B| \neq |M_2 - M_1|$. If this is the case, he enumerates a number into $T$ and wins by (3.2) (by performing the winning strategy that makes the formula $\text{Equal}(B, M_2 - M_1, \ldots)$ false). If $|B| = |M_2 - M_1|$, then exactly $|B|$ new elements of $M$ were enumerated into $M_2$, i.e., we have $|M_2| = (s + 1) * |B|$. Then player 2 enumerates $a$ into $L_1$ and waits until $N = N_1$. If this never happens, then player 2 wins by (3.3). So assume player 1 enumerates all of $N$ into $N_1$. Then $M_2 \subseteq N_1$ and player 2 enumerates all of $M_2$ into $M_1$ and waits until $a$ is enumerated into $K_1$. If this never happens, he wins by (3.4). If it happens and $L \neq A$, he starts a new stage $s + 1$ where the invariant $R_{s+1}$ holds.

If $|M| < |A| * |B|$, then there cannot be $|A|$ stages completed. Therefore there must be a stage $s < |A|$ in which player 2 wins.

If $|M| > |A| * |B|$ and there are $|A|$ stages completed (otherwise player 2 wins), then at the end of stage $|A| - 1$ the invariant $R_{|A|}$ holds. But this implies that condition (3.5) fails. Therefore player 2 wins.

b) Suppose that $|M| = |A| * |B|$; then player 1 has a recursive winning strategy. This is clear if one of the sets is empty. For the following we assume that they are all nonempty. Player 1 enumerates his sets such that $N_1 \subseteq N \subseteq M$, $K_1 \subseteq K_2 \subseteq L$ always. For player 2 we can assume that $L_1 \subseteq L \subseteq A$, $M_1 \subseteq N_1$ $M_2 \subseteq N$ always, since if one of these conditions does not hold, player 1 does nothing and either wins immediately or waits until it is satisfied.

The strategy proceeds in stages where each stage is started by player 2 when he enumerates a new element $a$ of $A$ into $L$. When player 2 does this he must wait until player 1 enumerates it into $K_1$ before he can enumerate another new element $b \neq a$ into $L$. Otherwise player 1 wins by (3.0).

So let us get started. Player 1 ensures that at the end of stage $s \geq 0$ the following invariant holds:

$$Q_s : |L| = s + 1, |N| = |L| * |B|, M_1 = M_2 = N_1 = N, K_1 = K_2 = L_1 = L,$$

and at the start of the game obviously $Q_{-1}$ holds.

At stage $s$, player 1 selects $|B|$ new elements from $M$ and enumerates them into $N$. He waits until all of them are enumerated into $M_2$. If this never happens, (3.1) (as well as (3.2)–(3.5)) is satisfied and therefore the formula is satisfied and player 1 wins. Note that, since player 2 must obey $M_2 \subseteq N$, player 1 can freeze $M_2 = N$. Also, since $N_1 \subseteq M_1$ player 1 freezes $N_1 = M_1$.

Now player 1 enumerates $a$ into $K_2$. If player 2 makes $T$ nonempty, he loses because $|B| = |M_2 - M_1| = |N - N_1|$. So, what else can player 2 do? As long as he does not enumerate $a$ into $L_1$, player 1 simply waits and does nothing which would win in end. Thus we may assume that player 2 eventually does this, in which case player 1 enumerates $N$ into $N_1$. Then he waits until player 2 enumerates $M_2$ (which is still frozen to $N$) into $M_1$, after which he enumerates $a$ into $K_1$. Then $Q_s$ holds.

It follows that if player 2 does not perform $|A|$ stages, he loses. If he performs $|A|$ stages, then at the end $Q_{|A|-1}$ holds. But this implies that (3.5) (as well as (3.1)–(3.4)) holds, i.e., player 1 wins. $\square$

This finishes the proof of Theorem 2.5.

*Remark* 3.7. The reduction can be modified such that it becomes computable in polynomial time (when all constants are encoded in binary). For this we build up the sets $M$ with $|M| = k$ by successive multiplication and addition according to Horner's rule applied to $k$.

Let us state the following corollary of the proof which is useful for later applications.

**Corollary 3.8.** *For every recursive relation $R \subseteq \omega^n$ there is a formula*

$$FR(A_1, \ldots, A_n, U_1 \ldots, U_k, V_1, \ldots, V_k)$$

*such that for any canonically given finite sets $A_1, \ldots, A_n$, player 1 has a winning strategy in $G_{FR}$ iff $R(|A_1|, \ldots, |A_n|)$ holds; otherwise player 2 has one. Furthermore, the winning strategies are uniformly recursive in the canonical indices of $A_1, \ldots, A_n$ and any recursive index of $R$. The formula can be found uniformly in any r.e. index of $R$.*

*Proof of Corollary* 2.6. (1) It suffices to show that arbitrary $\Sigma^0_{2n+1}$-sets are uniformly reducible to $P$ for all $n \geq 0$. The case $n = 0$ was shown above. To see how this is done for larger $n$ let us consider as an example $n = 1$. For any $\Sigma^0_3$-set $H$ and $e \in \omega$ we can uniformly compute a diophantine equation $p_1 = p_2$ with variables $x, y, z_1, \ldots, z_k$ such that

$$e \in H \iff (\exists x)(\forall y)(\exists z_1, \ldots, z_k) p_1 = p_2.$$

The corresponding game is roughly as follows. Player 1 enumerates a set $X$ that is double-checked by player 2, then player 2 enumerates a set $Y$ that is double-checked by player 1, and then player 1 enumerates sets $Z_1, \ldots, Z_k$ that are double-checked by player 2. After this player 1 claims that $(|X|, |Y|, |Z_1|, \ldots, |Z_k|)$ is a solution of the equation (actually, this is claimed with the double-checked sets). This part of the game is the same as above. Player 1 has a winning strategy iff $e \in H$.

In the following we outline the proof and leave all further details to the reader. The formula $F$ looks as follows:

$$F = B \subseteq Y \ \wedge \ F',$$

where $F'$ is the following formula:

$$
\begin{aligned}
[ \quad & (A \not\subseteq X \ \vee \ (A \neq X \ \wedge \ S_1 = \emptyset)) \\
\vee \quad & (S_1 \neq \emptyset \ \wedge \ S_2 = \emptyset \ \wedge \ B = Y) \\
(3.6) \qquad \vee \quad & (S_1 \neq \emptyset \ \wedge \ S_2 \neq \emptyset \ \wedge \ \bigvee_{i=1}^{k'} (C_i \not\subseteq Z_i \ \vee \ (C_i \neq Z_i \ \wedge \ S_3 = \emptyset))) \\
\vee \quad & (S_1 \neq \emptyset \ \wedge \ S_2 \neq \emptyset \ \wedge \ S_3 \neq \emptyset \ \wedge \ \bigcup_\ell T_\ell = \emptyset) \\
\vee \quad & (T_1 \neq \emptyset \ \wedge \ldots) \\
\vee \quad & (T_2 \neq \emptyset \ \wedge \ldots) \\
\vee \quad & \ldots ].
\end{aligned}
$$

Here the sets $B, S_1, S_3$ belong to player 1 and the sets $A, C_i, S_2, T_\ell$ belong to player 2. $Z_{k+1}, \ldots, Z_{k'}$ are additional sets to verify the equation as in the proof of Theorem 2.5.

As long as $S_1$ is empty, player 2 has to copy $X$ into $A$. If player 2 does this and $S_1$ stays empty, he wins. When $S_1$ becomes nonempty and $S_2$ is still empty, player 1 has to copy $Y$ into $B$, and so on. After the double-checks, $A, B, C_i$ have been enumerated and all $S_j$ are nonempty, player 2 chooses a signaling set $T_\ell$, and player 1 has to win the associated subgame. This is pretty much the same as in the proof of Theorem 2.5 with the following modifications: It is convenient to formulate the unions and intersections with the double-checked sets $C_i$ and put this formula $F''$ into $F'$, e.g. as $T_1 \neq \emptyset \ \wedge \ F''$. If $A$ or $B$ appears in this condition we replace it by $C_{k'+1}$, $C_{k'+2}$ and add the conditions $|C_{k'+1}| = |A|$ and $|C_{k'+2}| = |B|$. In this case the disjunction in (3.6) runs up to $k' + 2$.

(2) Just note that in the formulas of the proof of Theorem 2.5 that result from encoding a diophantine equation, player 1 has a recursive winning strategy if he has a winning strategy at all, i.e., if the diophantine equation has a solution. Since the latter problem is undecidable, it is also undeciable whether player 1 has a recursive winning strategy.

(3) This follows from (2) and the observation that player 2 has a recursive winning strategy if the diophantine equation has no solution.

(4) Let $E$ be any $\Sigma_{n+1}^0$-complete set. For the predicate $R(x, y) = [\chi_E(x) = y]$ consider the game where player 1 picks $a$ and then player 2 picks $b$, and player 2 wins iff $R(a, b)$ holds. In this game player 2 has a winning strategy, but he does not have a winning strategy recursive in $\mathbf{0}^{(n)}$ since $R$ is not recursive in $\mathbf{0}^{(n)}$. It is straightforward, by the method of (1), to translate this game into a basic game: Player 1 enumerates a set $A$ that is double-checked as $A'$ by player 2, then

player 2 enumerates a set $B$ that is double-checked as $B'$ by player 1. Now the game continues by evaluating the $\Sigma_{n+1}^0$-condition for $E$ as in (1). If player 2 had a winning strategy recursive in $\mathbf{0}^{(n)}$, then we could use it to compute $\chi_E(x)$ recursive in $\mathbf{0}^{(n)}$: Simulate the game by enumerating a set $A = \{1, \ldots, x\}$. Then using the winning strategy of player 2, let $A$ be double-checked by player 2 and find out which set $B$ player 2 enumerates, by double-checking it as $B' = B$. When player 2 enumerates a number in the corresponding signaling set we must have $\chi_E(x) = |B'|$. Note that this process terminates since otherwise player 2 would lose.

It can also be seen that in this game neither of the players has a winning strategy if both players are restricted to $\mathbf{0}^{(n)}$-strategies, i.e. the game is not determined with respect to such strategies.

(5) We show that the formulas resulting from the reduction of the proof of Theorem 2.5 can be rewritten in the language of the lattice of r.e. sets.

Let us first consider the language that contains union, intersection, and constants $0$ and $1$ denoting $\emptyset$ and $\omega$, respectively. Also we have equality in this language. Any formula of our original language can be rewritten as a Boolean combination of atomic formulas of the form

$$A_1 \cap \ldots \cap A_k \cap \overline{B_1} \cap \ldots \cap \overline{B_\ell} = \emptyset.$$

This is equivalent to

(3.7) $$A_1 \cap \ldots \cap A_k \subseteq B_1 \cup \ldots \cup B_\ell,$$

where the left-hand side is $\omega$ if $k = 0$ and the right-hand side is $\emptyset$ if $\ell = 0$. Since $X \subseteq Y$ can be rewritten as $X \cap Y = X$ we see that the formulas of our reduction can be expressed in the language of the lattice of r.e. sets. Now consider the corresponding $\Pi_2$-sentences where all variables of player 2 are universally quantified and all variables of player 1 are existentially quantified. Since the corresponding basic games are effectively determined we find that player 1 has a winning strategy iff the corresponding $\Pi_2$-sentence is uniformly valid. The straightforward details of the last step, using the recursion theorem, can be found in Proposition 4.2 in [19].

The basic games of the reduction are still effectively determined with the same winners if the players are enumerating only finite sets. Therefore we find that the $\Pi_2$-sentence corresponding to a game $G$ is uniformly valid where quantification is restricted to finite sets iff player 1 has a winning strategy in $G$. Thus, the second claim of Corollary 2.6(5) follows.

The latter problem is decidable if recursive indices are used for the uniformity. This follows from [19], Theorem 6.1 (like Corollary 6.2 there) when applied to the formula

$$(\forall M_1, \ldots, M_i)(\exists N_1, \ldots, N_j)[\bigwedge_{k=1}^{i} \mathrm{isFinite}(M_k) \ \rightarrow \ F \wedge \bigwedge_{\ell=1}^{j} \mathrm{isFinite}(N_\ell)],$$

where $(\forall M_1, \ldots, M_i)(\exists N_1, \ldots, N_j)F$ is the $\Pi_2$-sentence in question.

The undecidability results can also be shown for the language of the lattice containing only the predicate $\subseteq$. This is because we can define $\emptyset$, $\omega$, and $A \cap B$ by the subgames for $U_1 \subseteq V_1$, $V_2 \subseteq U_2$, and $U_3 \subseteq A \ \wedge \ U_3 \subseteq B \ \wedge \ (V_3 \subseteq A \ \wedge \ V_3 \subseteq B \ \rightarrow \ V_3 \subseteq U_3)$, respectively. The formula for $A \cup B$ is similar. Here the $U_i$'s belong to player 1 and the $V_i$'s belong to player 2. We modify the winning condition for player 1 by prefixing it conjunctively with the conjunction of these formulas and replacing each occurence of $\emptyset, \omega, A \cap B$ by $U_1, U_2, U_3$, respectively.

To get the undecidabilty result when quantification is restricted to finite sets one can check that in our reduction we do not need formulas of the form (3.7) with $k = 0$. Therefore we can dispense with $\omega$.                                  □

*Remark* 3.9. From the proof of (5) it follows that the original game is undecidable for the language that contains the operations union and intersection and only the equality predicate. Likewise it is undecidable for the language that contains only the $\subseteq$-predicate and no operations.

## 4. On the degree of unsolvability of $P$

We have seen that $P$ is at least as hard as TA. What about upper bounds? Right from the definition it can be shown that $P \in \Sigma^1_2$. Furthermore, the basic games are determined as can be easily seen by translating them into Borel games[5] on Baire space $\mathcal{N} = \omega^\omega$ and appealing to Borel determinacy[6] [25]. Therefore it follows that $P \in \Delta^1_2$. We come back to the issue of Borel games in the next section.

By using the methods of the last section—easier proofs can be given by using the characterization of the next section—it can be shown that this upper bound is the optimal placement of $P$ in the analytical hierarchy. To this end one shows first that for every recursive tree $T \subseteq \omega^{<\omega}$ there is, in a uniform way, a basic game $G$ such that player 1 has a winning strategy in $G$ iff $T$ has an infinite branch. We may assume that the finite sequences are effectively encoded in $\omega$ such that $\mathrm{code}(s) < \mathrm{code}(t)$ if $s$ is a proper initial segment of $t$. $G$ implements the following game, in which player 1 enumerates sets $E_1, E_2$ and player 2 sets $A_1, A_2, B_1, B_2$:

(1) Player 2 enumerates an element into either $A_2 - A_1$ or $B_2 - B_1$.
(2) If $A_2 - A_1 \neq \emptyset \ \wedge \ B_2 - B_1 = \emptyset$, then player 1 may enlarge $E_2$ by a finite amount. After that player 2 selects one of the following alternatives:
  (a) He wins instantly if $|E_1|$ is not the father of $|E_2|$ in (the encoding of) $T$.
  (b) To perform a new stage. Then he enumerates $A_2$ into $A_1$ making $A_2 - A_1$ empty and continues with step 1.
(3) This is analogous to step 2 with $A_i$ interchanged with $B_i$, $i = 1, 2$, and $E_1$ interchanged with $E_2$.

Player 2 wins iff he wins in step 2, (a) or step 3, (a). If $T$ has an infinite branch then player 1 has a winning strategy by enumerating into the $E_i$'s encodings of successive nodes of that branch. If $T$ does not have an infinite branch then player 2 has a winning strategy by alternately enumerating elements into $A_2 - A_1$ and $B_2 - B_1$. This forces player 1 to enumerate encodings of successive nodes of $T$ into the $E_i$'s. Since this enumerated branch is finite, player 2 eventually wins by step 2(a) or step 3(a).

Using the fact that the set of all indices of recursive trees is in the arithmetical hierarchy and also by using Corollary 2.6(1), it follows that for each $i$ there is in a uniform way a basic game $G_i$ such that player 1 has a winning strategy in $G_i$ iff $i$ is an index of a recursive tree with an infinite branch. Thus, a well-known $\Sigma^1_1$-complete set $A$ reduces to $P$, i.e. $P$ is $\Sigma^1_1$-hard. By [37], §16.7, Corollary XLI(b)

---

[5]The players alternately choose numbers $\langle i_1, \ldots, i_n \rangle$ with the intended meaning that player 1 (player 2) enumerates the finite set with canonical index $i_k$ into $U_k$ ($V_k$) for $1 \leq k \leq n$.
  [6]A closer look reveals that we only need the determinacy for Boolean combinations of lightface $\Sigma^0_2$-sets which was already established in [8]. See [12] for a self-contained presentation.

it follows that there is a basic game such that player 1 has a winning strategy but no hyperarithmetic one.

But $P$ is still much harder than $\Sigma_1^1$: One can in a similar way as above (basically by iterating the game twice; see Example 5.2 below) show that the hyperjump of $A$ (i.e. the set of all indices $i$ such that $\varphi_i^A$ is the characteristic function of a finite-path tree; cf. [37], p. 412) reduces to $P$. Therefore $P$ properly belongs to $\Delta_2^1$. The game can be iterated further, and so the lower bound can be sharpened to the corresponding iteration of the hyperjump. We leave it open up to which ordinal the iterated hyperjump of $A$ reduces to $P$.

By reversing the roles of player 1 and player 2 and using the fact that the index set of recursive trees actually belongs to $\Pi_2^0$, there are unformly computable formulas $F_i'$ such that player 1 has a *recursive* winning strategy in $G_{F_i'}$ iff $i$ is an index of a recursive finite-path tree. It follows that

$$P^{\mathrm{eff}} = \{F : \text{Player 1 has a recursive winning strategy in } G_F\}$$

is $\Pi_1^1$-hard. An easy computation shows that $P^{\mathrm{eff}}$ also belongs to $\Pi_1^1$, therefore it is $\Pi_1^1$-complete.

Finally let us consider

$$
\begin{aligned}
P^{\mathrm{strongly\ eff}} \quad = \quad &\{F : \text{Player 1 has a recursive strategy in } G_F \text{ that wins}\\
&\text{against any recursive strategy of player 2}\}.
\end{aligned}
$$

It is easy to see that $P^{\mathrm{strongly\ eff}}$ belongs to $\Sigma_4^0$. We now show that $P^{\mathrm{strongly\ eff}}$ is also hard for $\Sigma_4^0$.

Let $M$ be a $\Sigma_4^0$-complete set. Then there is a recursive function $g$ such that $M = \{x : (\exists i)(\forall j)[W_{g(x,i,j)} \text{ is finite}]\}$. As usual (see [41], Chap. I, Definition 4.1) let $W_{k,s}$ denote the standard approximation of $W_k$ after $k$ steps of computation, where we assume that if $x \in W_{k,s}$, then $x, k < s$ (in particular $W_{k,s} = \emptyset$ if $k \geq s$).

Now consider the following game $G_{M,x}$:

(1) Player 1 enumerates finitely many numbers into $I$.
(2) Player 2 enumerates finitely many numbers into $J$.
(3) Player 1 enlarges $S$ by some finite amount.
(4) Player 2 enlarges $T$ by some finite amount.
(5) Player 1 wins instantly if $W_{g(x,|I|,|J|),|T|} \subseteq W_{g(x,|I|,|J|),|S|}$.
(6) Player 1 may start a new iteration; in that case the game is continued with step (3).

Player 1 wins $G_{M,x}$ iff he eventually wins in step (5). Note that player 1 has a winning strategy iff $x \in M$. Furthermore this strategy may be chosen to be recursive. If $x \notin M$, then for any strategy of player 1 there is a recursive strategy of player 2 that wins for player 2.

It is straightforward by the methods of the previous section to effectively translate this game into a basic game. Therefore $P^{\mathrm{strongly\ eff}}$ is $\Sigma_4^0$-complete.

*Remark* 4.1. It follows that $P^{\mathrm{eff}}$ is a proper subset of $P^{\mathrm{strongly\ eff}}$. I.e., there are formulas $F$ such that player 1 does not have a recursive winning strategy in $G_F$, but he has one that wins against any recursive strategy of player 2. Concrete examples are provided using recursive trees that have an infinite branch but no recursive infinite branch.

## 5. A characterization of basic games

We characterize the problem $P$ in terms of effectively presented Borel games on $\mathcal{N}$. It turns out that basic games and $\mathcal{B}(\Sigma_2^0)$-games are equivalent, where $\mathcal{B}(\Sigma_2^0)$ denotes the Boolean combinations of lightface $\Sigma_2^0$-sets.

Let use first recall some basic notions from effective descriptive set theory (for background see [30], Chap. 3 and [37], §15.2).[7]

Let node $: \omega \to \omega^{<\omega}$ denote the inverse of code. A set $A \subseteq \omega$ is called the *support* of the open set $O \subseteq \mathcal{N}$ if

$$O = \{f \in \mathcal{N} : (\exists a \in A)[f \text{ extends node}(a)]\}.$$

An open set is called *effectively open* (or $\Sigma_1^0$) iff it has an r.e. (or equivalently, recursive) support. Let $O_i$ denote the open set with support $W_i$.

A set $S \subseteq \mathcal{N}$ is a $\Sigma_2^0$-set if there is an r.e. set $M \subseteq \omega$ such that

$$S = \bigcup \{\overline{O_n} : n \in M\}.$$

Let $S_i$ denote the set that is $\Sigma_2^0$ via $W_i$.

A set $B$ is a $\mathcal{B}(\Sigma_2^0)$-set if it is a Boolean combination of $\Sigma_2^0$-sets. We code these sets as numbers in some standard fashion.[8] Let $B_k$ denote the $\mathcal{B}(\Sigma_2^0)$-set with code $k \in \omega$.

For any set $A \subseteq \mathcal{N}$ we denote by $G(A)$ the Gale-Stewart game on $\mathcal{N}$, where $A$ is the winning set of player 1. If $A$ is a $\mathcal{B}(\Sigma_2^0)$-set, $G(A)$ is called a $\mathcal{B}(\Sigma_2^0)$-game.

In [7] Gale-Stewart games on Cantor space $2^\omega$ were considered, where the winning set of player 1 is an effectively closed set. It is shown there that the winning strategies of such games correspond to recursively bounded $\Pi_1^0$-classes.

Finally let $B_2^0 = \{k \in \omega : \text{Player 1 has a winning strategy in } G(B_k)\}$. Then we can state our characterization as follows:

**Theorem 5.1.** $B_2^0 \equiv P$.

*Proof.* It is an easy exercise to define in a uniform way for every basic game $G$ a $\mathcal{B}(\Sigma_2^0)$-set $B$ such that player 1 has a winning strategy in $G$ iff player 1 has a winning strategy in $G(B)$.

The hard part is to prove the converse. Let us first consider $\Sigma_2^0$-sets and then generalize to Boolean combinations of such sets.

So suppose that $S$ is a $\Sigma_2^0$-set. Then we can uniformly compute a strictly increasing recursive function $g$ such that for all $f \in \mathcal{N}$,

$$f \in S \iff (\exists j)[f \notin O_{g(j)}].$$

Let $M(k, t) = \{\text{node}(a) : a \in W_{k,t}\}$.

Now we are ready to define a game $G_S$ in which player 1 enumerates sets $D, E_1$ and player 2 sets $E_2, T$ as follows:

(1) Player 2 enumerates finitely many numbers into $E_2$.
(2) Player 1 wins instantly if $|E_2|$ does not code the empty sequence.
(3) Player 1 enlarges $E_1$ by some finite amount.

---

[7] In descriptive set theory the effective analogs of the 'boldface' classes $\mathbf{\Sigma_n^0}$ are called 'lightface' $\Sigma_n^0$. The reader should not confuse this notation with the levels of the arithmetical hierarchy of sets of numbers. In the rest of the paper $\Sigma_n^0$ and $\Pi_n^0$ will always denote Borel classes.

[8] E.g. as follows: the number $\langle b, i_1, \ldots, i_n \rangle$ codes the Boolean combination $t_b[S_{i_1}, \ldots, S_{i_n}]$ if $t_b(x_1, \ldots, x_n)$ is the $b$-th Boolean term with operations union, intersection and complement, and variables $x_1, \ldots, x_n$.

(4) Player 2 wins instantly if node($|E_2|$) is not the father of node($|E_1|$).
(5) Player 2 enlarges $E_2$ by some finite amount.
(6) Player 1 wins instantly if node($|E_1|$) is not the father of node($|E_2|$).
(7) Player 2 enlarges $T$ by some finite amount.
(8) Player 1 enlarges $D$ by some finite amount.
(9) Player 2 wins instantly if node($|E_2|$) extends a node in $M(g(|D|), |T|)$.
(10) Player 2 may start a new iteration; in that case the game is continued with step (3).

Player 1 wins the game either if he wins instantly in steps (2) or (6), or if at the end $D$ is finite.

We claim that player 1 has a winning strategy in $G(S)$ iff he has a winning strategy in $G_S$.

First, if player 1 has a winning strategy in $G(S)$ he uses this same strategy in $G_S$ on codes instead of nodes, and in addition he enlarges $D$ in step (8) such that $|D|$ is the least $j$ such that node($|E_2|$) does not extend any sequence in $M(g(j), |T|)$. Note that $j$ always exists by the choice of the approximation $W_{k,t}$ and the hypothesis that $g$ is stricly increasing. Let $f$ denote the function that is produced in a play of $G(S)$ when player 1 is following his winning strategy. Then there exists a minimal $j$ such that $f \notin O_{g(j)}$, i.e., $f$ does not extend any sequence in $M(g(j), t)$ for all $t$. Therefore, in the corresponding play on $G_S$, $D$ settles to a set with cardinality $j$, i.e., player 1 wins.

Second, if player 1 has a winning strategy in $G_S$ he wins against any player 2 that makes $T$ infinite. If in such a play a function $f$ is produced ($f$ is the extension of all node($|E_{1,s}|$) where $E_{1,s}$ is the value of $E_1$ after iteration $s$), then since $D$ is finite we must have $f \notin O_{g(|D|)}$, i.e., $f \in S$. Thus we directly get a winning strategy in $G(S)$.

Using the methods of Section 3 one can effectively translate the game $G_S$ into a basic game. We defer the details to the Appendix and just mention that the condition that $D$ is finite is translated into the subformula Greater$(U, D, U_1, U_2, V_1, V_2)$, where the $U$'s belong to player 1 and the $V$'s belong to player 2. It is easy to see that player 1 has a recursive strategy that wins if $D$ is finite. On the other hand, player 2 has the following copying strategy that wins if $D$ is infinite: For each $i \geq 0$ and $b \in \{1, 2\}$, enumerate the $i$-th element of $D$ into $V_b$ iff player 1 enumerated the $i$-th element of $U$ into $U_b \cap U_2$ (and $|D| > i$). Here the $i$-th element is understood according to the order of enumeration of $D$ and $U$, respectively.

The construction of $G_S$ is readily generalized: Suppose that $B$ is a $\mathcal{B}(\Sigma_2^0)$-set. Since $\Sigma_2^0$-sets are uniformly closed under union and intersection, it follows that for $B$ we can uniformly compute $n \geq 1$ and strictly increasing recursive functions $g_i, h_i$, $1 \leq i \leq n$, such that for all $f \in \mathcal{N}$,

$$f \in B \iff \bigvee_{i=1}^{n} [(\exists j)[f \notin O_{g_i(j)}] \wedge (\forall j)[f \in O_{h_i(j)}]].$$

The game $G_B$ is defined similar to the game $G_S$ above, only steps (7)–(10) are modified as follows:

(7) (a) Player 1 may enlarge $T_1$ by a finite amount.
    (b) Player 2 may enlarge $T_2$ by a finite amount.
(8) (a) Player 1 may enlarge $D_1^1, \ldots, D_n^1$ by a finite amount.
    (b) Player 2 may enlarge $D_1^2, \ldots, D_n^2$ by a finite amount.

(9)   (a) Player 2 wins instantly if node($|E_2|$) extends a sequence in

$$\bigcup_{i=1}^{n} M(g_i(|D_i^1|), |T_2|).$$

(b) Player 1 wins instantly if node($|E_1|$) extends a sequence in

$$\bigcup_{i=1}^{n} M(h_i(|D_i^2|), |T_1|).$$

(10)         Player 1 may start a new iteration; in that case the game is continued
             with step (3).

Player 1 wins either if he wins instantly in steps (2) or (6), or if there is an $i$,
$1 \leq i \leq n$, such that at the end $D_i^1$ is finite and $D_i^2$ is infinite.

Similar as above it follows that player 1 has a winning strategy in $G(B)$ iff
player 1 has a winning strategy in $G_B$. Furthermore in more or less the same way
as above it is possible to effectively translate $G_B$ into a basic game. Let us just
mention that now the loop counter is enumerated by player 1, and player 1 wins if
he wins finitely or if the loop counter is infinite and the conditions on the $D$'s hold.
This completes the proof sketch.                                          □

Note that we have obtained the stronger result that for the corresponding basic
games and $\mathcal{B}(\Sigma_2^0)$-games the winning strategies can also be effectively translated
back and forth. One may call a recursive class of $\mathcal{B}(\Sigma_2^0)$-games with this property
'strongly complete'. There should be many other natural strongly complete classes
except basic games. Our result may simplify the proofs for strong completeness
since it suffices to show that in such a class basic games can be emulated.

Theorem 5.1 summarizes our previous results: It can be used to give direct proofs
of the lower bounds mentioned in Section 4. Using the strong completeness of $P$,
all results of Corollary 2.6, except (5), follow easily as well.

**Example 5.2.** Let us just provide the proof of a result claimed in Section 4. We
define in a uniform way $\Sigma_2^0$-games $G_i$ such that player 1 has a winning strategy
in $G_i$ iff $\varphi_i^A$ has an infinite branch, where $A$ is the $\Sigma_1^1$-complete set $\{x \in \omega :
\varphi_x$ codes a tree $\subseteq \omega^{<\omega}$ with an infinite branch$\}$ and $\varphi_i^A$ is a total tree.

The game is roughly as follows. We omit the subgames for challenging and
checking the totality and the tree property.

Player 1 enumerates numbers $a_0, a_1, \ldots$ which code successive finite sequences.
He wins if player 2 does not interrupt. If player 2 interrupts at stage $s_0$, then
player 1 presents a computation of the oracle Turing machine $i$ with input $a_{s_0}$. Let
$(x_1, b_1), \ldots, (x_n, b_n)$ with $x_j \in \omega$, $b_j \in \{0, 1\}$ be the oracle queries and answers in
this computation. Player 2 wins if the result of the computation is 0. Otherwise
he may challange an $(x_j, b_j)$, by claiming that $A(x_j) = 1 - b_j$.

If $b_j = 0$, he enumerates numbers $c_0, c_1, \ldots$ which code successive finite sequences
until player 1 interrupts. He wins if player 1 does not interrupt. If player 1 inter-
rupts at stage $s_1$, player 2 presents a computation of Turing machine $x_j$ with input
$c_{s_1}$ and wins iff the result of this computation is 1.

If $b_j = 1$, player 1 must enumerate a sequence $d_0, d_1, \ldots$ until player 2 interrupts
him. Player 1 wins if there is no interruption. If player 2 interrupts at stage $s_2$,
then player 1 has to provide a computation of Turing machine $x_j$ with input $d_{s_2}$.
Player 1 wins iff the result of this computation is 1.

It is straightforward to check that the winning set of player 1 in this game is a $\Sigma_2^0$-set.

In effective descriptive set theory the class $\mathcal{G}(\Sigma_2^0)$ is considered where $\mathcal{G}$ is the 'game quantifier'; cf. [30], p. 322. The connection is the following: For a set $A \subseteq \omega \times \mathcal{N}$ let $A_k = \{f \in \mathcal{N} : (k, f) \in A\}$. $A$ belongs to lightface $\Sigma_2^0$ iff there is a recursive function $h$ such that for all $k \in \omega$, $A_k = S_{h(k)}$. By definition, a set $N \subseteq \omega$ belongs to $\mathcal{G}(\Sigma_2^0)$ iff there is a set $A \subseteq \omega \times \mathcal{N}$ in lightface $\Sigma_2^0$ such that $N = \{k \in \omega : \text{Player 1 has a winning strategy in } G(A_k)\}$. Therefore our set $\{k \in \omega : \text{Player 1 has a winning strategy in } G(S_k)\}$ is $m$-complete for $\mathcal{G}(\Sigma_2^0)$. In particular, $B_2^0$ is hard for $\mathcal{G}(\Sigma_2^0)$.

Solovay has shown that $\mathcal{G}(\Sigma_2^0)$ equals the class of all $\Sigma_1^1$-inductive sets; see [17], Theorem 2.5.2 and [30], 7.C.10, p. 414. Therefore, $B_2^0$ is hard for the $\Sigma_1^1$-inductive sets, and lower bounds can be transferred from this class.

## 6. A characterization of mixed games

Recall that mixed games are modified basic games with the additional predicate isFinite$(x)$. These games are charaterized one level higher up in the effective Borel hierarchy. This may come as a surprise since Lachlan [22] proved that games with the single predicate isFinite$(x)$ are recursive.

Let $P^{\mathrm{mix}}$ denote the set of all formulas $F$ in the language of mixed games such that player 1 has a winning strategy in $G_F$.

Let us recall the following notations. A set $A \subseteq \mathcal{N}$ is called a $\Sigma_3^0$-set iff there is an r.e. set $W$ such that $A = \bigcup\{\overline{S_i} : i \in W\}$. A set $C$ is called a $\mathcal{B}(\Sigma_3^0)$-set if it is a Boolean combination of $\Sigma_3^0$-sets. We code these sets as numbers in some standard fashion. Let $C_k$ denote the $\mathcal{B}(\Sigma_3^0)$-set with code $k \in \omega$. Finally let $B_3^0 = \{k \in \omega : \text{Player 1 has a winning strategy in } G(C_k)\}$. Then we can state our characterization as follows:

**Theorem 6.1.** $B_3^0 \equiv P^{\mathrm{mix}}$.

*Proof.* Again it is straightforward to define in a uniform way for every mixed game an equivalent $\mathcal{B}(\Sigma_3^0)$-game.

For the converse let use first consider $\Sigma_3^0$-sets and then generalize to Boolean combinations thereof. So suppose that $A$ is a $\Sigma_3^0$-set. Then we can uniformly compute a recursive function $g$ such that for all $f \in \mathcal{N}$,

$$f \in A \iff (\exists i)(\forall j)(\exists a \in W_{g(i,j)})[f \text{ extends node}(a)].$$

Now we are ready to define a game $G_A$ where player 1 enumerates sets $C_1, C_2, D_1, D_2, E_1, T$, and player 2 enumerates sets $D_3, E_2$ as follows:

(1) Player 2 enumerates finitely many numbers into $E_2$.
(2) Player 1 wins instantly if $|E_2|$ does not code the empty sequence.
(3) Player 1 enlarges $E_1$ by some finite amount.
(4) Player 2 wins instantly if node$(|E_2|)$ is not the father of node$(|E_1|)$.
(5) Player 2 enlarges $E_2$ by some finite amount.
(6) Player 1 wins instantly if node$(|E_1|)$ is not the father of node$(|E_2|)$.
(7) Player 1 enlarges $D_1, D_2$ by some finite amount such that $D_1 \subseteq D_2$.
(8) Player 2 enlarges $D_3$ by some finite amount such that $D_3 \subseteq D_2$.
(9) Player 1 enlarges $C_1, C_2, T$ by some finite amount such that $C_1 \subseteq C_2$.

(10) Player 2 wins instantly if there does not exist $i \leq |D_2| - |D_3|$ such that for all $j < |C_2| - |C_1|$,

$$\text{node}(|E_2|) \text{ extends a sequence in } M(g(i,j), |T|).$$

(11) Player 1 may start a new iteration; in that case the game is continued with step (3).

Player 1 wins $G_A$ iff he wins in steps (3) or (6), or at the end

(6.1)          $[D_2 \text{ is infinite } \wedge \ D_2 - D_1 \text{ is finite } \wedge \ C_2 - C_1 \text{ is infinite}] \ \vee$

(6.2)          $[D_2 - D_1 \text{ is infinite } \wedge \ D_2 - D_3 \text{ is finite}].$

Using our methods from the previous sections, $G_A$ can be uniformly implemented as a mixed game. It remains to prove the correctness.

For $s \geq 0$, let $X_s$ denote the value of set $X$ at the end of the $s$-th iteration.

a) Suppose that player 1 has a winning strategy in $G_A$; then we have to show that he also has a winning strategy in $G(A)$. We translate the enumerations of $E_1$ and $E_2$ into respective moves in $G(A)$ and vice versa. In addition we let player 2 in $G_A$ play the following additional strategy on $D_3$:

If in iteration $s + 1$ player 1 enumerates the $i$-th element of $D_{2,s+1} - D_{1,s}$ into $D_1$, then player 2 enumerates for all $j \geq i$ with $j < |D_{2,s+1} - D_{3,s}|$ the $j$-th element of $D_{2,s+1} - D_{3,s}$ into $D_3$. (This is a well-known strategy from recursion theory, called *dumping*; see [41], p. 197.)

It follows that if $D_2 - D_1$ is infinite, then $D_2 - D_3$ is infinite, too. Therefore, since player 1 is winning the game, at the end (6.1) holds. Let $i = |D_2 - D_1|$. Then, since $D_2$ is infinite, at infinitely many iterations $s + 1$ the $i$-th element of $D_{2,s+1} - D_{1,s}$ is enumerated into $D_{1,s+1}$. At all of these stages we have $|D_{2,s+1} - D_{3,s+1}| \leq i$. Since $|C_{2,s+1}| - |C_{1,s+1}|$ goes to infinity and player 2 cannot win in step (10) (otherwise we would not have a winning strategy for player 1), it follows that there must be a $k \leq i$ such that

$$(\forall j)(\exists a \in W_{g(k,j)})[f \text{ extends node}(a)],$$

where $f$ is the function played in $G(A)$. Thus $f \in A$ and player 1 wins $G(A)$ with the translated strategy.

b) Suppose that player 1 has a winning strategy in $G(A)$; then we have to show that he also has a winning strategy in $G_A$. Again we translate the moves in $G(A)$ into enumerations of $E_1$ and $E_2$ and vice versa. For the following we assume that player 2 always enumerates proper successors of the nodes of player 1 (otherwise player 1 already wins in step (6)). We show how player 1 has to enumerate the sets $C_1, C_2, D_1, D_2, T$ in order to win in $G_A$.

First, he lets $D_{2,s+1} = C_{2,s+1} = T_{s+1} = \{0, \ldots, s\}$.

Second, he defines an auxiliary function $f(i, s)$ as follows:

$$\begin{aligned}
f(i, 0) &= i, \\
f(i, s + 1) &= f(i, s) + 1, \text{ if } |D_{2,s} - D_{3,s}| < i, \\
f(i, s + 1) &= f(i, s), \text{ else.}
\end{aligned}$$

$f$ is used to implement the well-known *kicking* strategy from recursion theory; see [41], p. 67.

Third, let

$$m(i, s+1)$$
$$= \max\{k \le s : (\forall j < k)(\exists a \in M(g(i,j), s+1))[\text{node}(|E_{2,s+1}|) \text{ extends } a]\}.$$

Note that $m$ is nondecreasing in its second argument.

In iteration $s+1$, at step (7), let $i_{s+1}$ be the minimal $i$ such that $m(i, s+1) > m(i, s)$ if such an $i$ exists and $i_{s+1} = s$ otherwise. Enumerate, if it exists, the $f(i_{s+1}, s+1)$-th element of $D_{2,s+1} - D_{1,s}$ into $D_1$. (The use of $m$ is also well known from recursion theory; cf. [37], §12.6, Theorem XIX.)

In step (9) compute $m(s+1) = \max\{m(k, s+1) : k \le |D_{2,s+1} - D_{3,s+1}|\}$ and enumerate the $j$-th element of $C_{2,s+1} - C_{1,s}$ into $C_1$, for all $j \ge m(s+1)$ with $j < |C_{2,s+1} - C_{1,s}|$ (another instance of dumping). Thus we have $|C_{2,s+1}| - |C_{1,s+1}| \le m(s+1)$ and player 1 will not lose in step (10).

Let us now verify that with these definitions player 1 wins in $G_A$. We have already seen that he does not lose in a finite number of iterations. Therefore it suffices to show that condition (6.1) or (6.2) holds.

Since player 1 is using a winning strategy in $G(A)$ the resulting $f$ belongs to $A$. Therefore there must be an $i$ such that $m(i, s)$ goes to infinity. Let $i'$ be the least such $i$.

If infinitely often $|D_{2,s+1} - D_{3,s+1}| < i'$, then $f(i', s+1)$ goes to infinity and therefore $D_2 - D_1$ is infinite, but $D_2 - D_3$ is finite. Then player 1 wins by condition (6.2).

Otherwise there is $s_0$ such that $|D_{2,s} - D_{3,s}| \ge i'$ for all $s \ge s_0$. Thus $m(s+1)$ goes to infinity and $C_2 - C_1$ is infinite. Also $f(i', s+1) = f(i', s_0)$ for all $s \ge s_0$ which implies that $D_2 - D_1$ is finite. So condition (6.1) holds and player 1 again wins.

This completes the correctness proof.

In an analogous way as in the proof of the previous theorem the construction can be generalized to Boolean combinations of $\Sigma_3^0$-sets. $\qquad\square$

Again, the proof shows a stronger result in that winning strategies can be effectively translated back and forth.

$B_3^0$ also belongs to $\Delta_2^1$, and it is not difficult to prove that $B_3^0$ is not Turing reducible to $B_2^0$. Thus it follows that there is no effective way to compute for a requirement of a mixed game an equivalent (in the sense that the winner is preserved) requirement of a basic game.

$B_3^0$ is hard for the class $\mathcal{G}(\Sigma_3^0)$ that was studied in effective descriptive set theory. This class seems to be rather complex; cf. [26], Theorem E. In [15] a characterization of the class and of the complexity of winning strategies in terms of recursion in higher types is obtained. Related results appear in unpublished work of Martin and Solovay [27].

## 7. Games with infinitely many requirements

Usually in recursion theory one has to satisfy an infinite r.e. sequence of requirements, not just a single one. As in [22] we can extend our games to model this case.

To this end we have an infinite r.e. sequence of formulas $F_i$, $i \in \omega$, as in Definition 2.1, but now with an infinite supply of variables $u_0, u_1, \ldots$ and $v_0, v_1, \ldots$. In each round a player may enumerate finitely many numbers into finitely many of his

sets. Player 1 has the infinite sequence of sets $U_i$, $i \in \omega$, and player 2 the infinite sequence of sets $V_i$, $i \in \omega$. Player 1 wins iff after $\omega$ rounds all $F_i$ are satisfied by the enumerated sets.

Let $P_{\text{inf}}$ and $P_{\text{inf}}^{\text{mix}}$ be the corresponding decision problems for infinite basic and mixed games.

As in the previous sections we may define $\Pi_3^0$-sets and $\Pi_4^0$-sets. Let $Q_i$ be the $i$-th $\Pi_3^0$-set and let $R_i$ be the $i$-th $\Pi_4^0$-set.

Let $P_3^0 = \{k \in \omega : \text{Player 1 has a winning strategy in } G(Q_k)\}$ and $P_4^0 = \{k \in \omega : \text{Player 1 has a winning strategy in } G(R_k)\}$.

It is easy to see that for each infinite basic game there is in a uniform way an equivalent $\Pi_3^0$-game.[9] Also for each infinite mixed game there is in a uniform way an equivalent $\Pi_4^0$-game.

The converse can be shown in a similar way as in the previous sections. For details we refer the reader to the Appendix. Thus we get the following result:

**Theorem 7.1.** $P_3^0 \equiv P_{\text{inf}}$ and $P_4^0 \equiv P_{\text{inf}}^{\text{mix}}$.

One can also consider infinitary games with the only predicate isFinite$(x)$. Lachlan [22], p. 305, gave an example where player 1 has a winning strategy for each finite subsequence of the requirements, but not for the whole sequence. Therefore, a quick reduction to the case of a single requirement (which is decidable) fails. We leave it open to classify the complexity of this problem. Cf. also question (ii) of Lachlan [22], p. 309.


## 8. Conclusion

Our results seem to be devastating for Lachlan's orginal program of effectivizing the proof search by analyzing single requirements in terms of games. However, in practice this approach turns out to be very useful.[10] Is there an explanation for this discrepancy?

We conclude with some directions for further research for the recursion theorist, the complexity theorist and the philosopher.

First, is there a nice characterization of the Turing degrees of $B_2^0$, $B_3^0$, $P_3^0$, $P_4^0$ and the other levels of the effective Borel hierarchy? Are there other natural problems, not necessarily about games, that are equivalent to these sets or higher levels? There was much interest in these issues in the Seventies, but it may be worthwhile to re-examine them. One can also consider variants of the basic games with other predicates.

Second, it may be of interest to study resource bounded enumeration games. For instance, one could add a parameter $s$ to the input and play the games on the finite domain $\{0, \ldots, s\}$. What is the complexity of determining which player has a winning strategy? If $s$ is given in unary, the problem of whether player 1 has a winning strategy for a given formula $F$ (with the predicate isEmpty$(x)$) and $s$ is PSPACE-complete. Many other questions can be asked.

---

[9]Therefore it follows from Theorem 6.1 that for each r.e. index of an infinite basic game $G$ we can uniformly compute an equivalent single mixed formula $F$ (i.e., player 1 has a winning strategy in $G$ iff he has a one in $G_F$).

[10]This was already observed by Lachlan [22], p. 307, in connection with an analogous situation for his 'enumeration games'.

Third, what is the impact on philosophy of the infinite games studied in recursion theory (or more generally in logic)? For a starting point of this question we refer the reader to [6, 40].

## 9. Appendix

**Details of the proof of Theorem 5.1.** We present the details of how the game $G_S$ from Section 5 is implemented as a basic game. First, using Corollary 3.8, it follows that there are formulas $F_1(X, Y, \ldots), F_2(X, Y, \ldots), F_3(X, Y, Z, \ldots)$ with corresponding effectively determined basic games $G_1, G_2, G_3$ such that for any canonically given finite sets $X, Y, Z$: Player 1 has a winning strategy in $G_1$ iff node($|X|$) is not the father of node($|Y|$). Player 1 has a winning strategy in $G_2$ iff node($|Y|$) is the father of node($|X|$). Player 1 has a winning strategy in $G_3$ iff node($|X|$) does not extend any sequence in $M(g(|Y|), |Z|)$. Let $r$ denote the code of the empty sequence.

Now we can write down the formula $F_S$ which implements the game $G_S$. Here the variables $C, D, E_1, E_2', M_1, M_2, M_3, T', U, U_1, U_2, X$ belong to player 1 and the variables $A, D', E_1', E_2, N_1, \ldots, N_4, T, V_1, \ldots, V_{r+2}, Y_1, Y_2$ belong to player 2. The primed variables are used for double-checking. $A$ is the loop counter, $M_1, M_2, M_3$ are the step counters of player 1 and $N_1, \ldots, N_4$ are the step counters of player 2. $F_S$ is the following formula:

(9.0) $\quad \neg(N_1, N_2, N_3, N_4 \subseteq A \ \wedge \ E_1' \subseteq E_1 \ \wedge \ D' \subseteq D) \ \vee$

(9.1) $\quad |A - M_3| > 1 \ \vee$

(9.2) $\quad (A = \emptyset \ \wedge \ E_2' = E_2) \ \vee$

$\qquad (E_1 = \emptyset \ \wedge \ A \neq \emptyset \ \wedge \ E_2' \subseteq E_2 \ \wedge$

(9.3) $\quad \neg \mathrm{Equal}_r(E_2', V_1, \ldots, V_r, U_1, U_2)) \ \vee$

(9.4) $\quad (X \neq \emptyset \ \wedge \ E_2' \subseteq E_2 \ \wedge \ A = M_1 = N_1 = N_2 \ \wedge \ F_1(E_1', E_2', \ldots)) \ \vee$

(9.5) $\quad [\ X = \emptyset \ \wedge \ M_1, M_2, M_3 \subseteq A \ \wedge \ E_2' \subseteq E_2 \ \wedge \ T' \subseteq T \ \wedge$

(9.6) $\quad (Y_1 = Y_2 = \emptyset \ \wedge \ A \neq M_1 \ \rightarrow \ E_1' \neq E_1) \ \wedge$

(9.7) $\quad (Y_1 \neq \emptyset \ \wedge \ A = M_1 \ \wedge \ A \neq N_1 \ \rightarrow \ F_2(E_1', E_2', \ldots)) \ \wedge$

(9.8) $\quad (Y_1 = Y_2 = \emptyset \ \wedge \ A = M_1 = N_1 \ \wedge \ A \neq N_2 \ \rightarrow \ E_2' = E_2) \ \wedge$

(9.9) $\quad (Y_1 = Y_2 = \emptyset \ \wedge \ A = M_1 = N_1 = N_2 \ \wedge \ A \neq N_3 \ \rightarrow \ T' = T) \ \wedge$

(9.10) $\quad (Y_1 = Y_2 = \emptyset \ \wedge \ A = M_1 = N_1 = N_2 = N_3 \ \wedge \ A \neq M_2 \ \rightarrow \ D' \neq D) \ \wedge$

$\qquad (Y_2 \neq \emptyset \ \wedge \ A = M_1 = N_1 = N_2 = N_3 = M_2 \ \wedge \ A \neq N_4 \ \rightarrow$

(9.11) $\qquad F_3(E_2', D', T', \ldots)) \ \wedge$

(9.12) $\quad (Y_1 = Y_2 = \emptyset \ \wedge \ A = M_1 = N_1 = N_2 = N_3 = M_2 = N_4 \ \rightarrow \ M_3 = A) \ \wedge$

(9.13) $\quad (Y_1 = Y_2 = \emptyset \ \rightarrow \ \mathrm{Greater}(U, D', U_1, U_2, V_{r+1}, V_{r+2}))\ ].$

Subformula (9.4) corrsponds to step (6), (9.5) is a consistency condition, and the other subformulas (9.2)–(9.11) correspond to steps (1)–(9), respectively. The formal correctness proof is done in the same way as the correctness proof for Mult.

**Details of the proof of Theorem 7.1.** We present the details of how to uniformly compute for any given $\Pi_3^0$-set $A$ an r.e. sequence of formulas $H_i$, $i \in \omega$, such that player 1 has a winning strategy in the corresponding infinite basic game $G$ iff player 1 has a winning strategy in $G(A)$. So, suppose $A$ is such a set. Then there

uniformly exists a recursive function $g$, strictly increasing in the second argument, such that for all $f \in \mathcal{N}$,

$$f \in A \iff (\forall i)(\exists j)(\forall a \in W_{g(i,j)})[f \text{ does not extend node}(a)].$$

We define formulas $H_i$ with the following intuition:

$H_0$ implements the following game $G_0$, where player 1 enumerates $E_1$ and player 2 enumerates $E_2$:

(1) Player 2 enumerates finitely many numbers into $E_2$.
(2) Player 1 wins instantly if $|E_2|$ does not code the empty sequence.
(3) Player 1 enlarges $E_1$ by some finite amount.
(4) Player 2 wins instantly if node($|E_2|$) is not the father of of node($|E_1|$).
(5) Player 2 enlarges $E_2$ by some finite amount.
(6) Player 1 wins instantly if node($|E_1|$) is not the father of node($|E_2|$).
(7) Player 1 may perform a new iteration; in that case the game is continued with step (3).

Player 1 wins iff he wins finitely in steps (2) or (6), or infinitely many iterations are performed. $G_0$ is just the first part of the game from the proof of Theorem 5.1 (cf. steps (1)–(6) there).

$H_{i+1}$ implements a game $G_{i+1}$ which player 1 should be able to win iff for the $f$ built in $G_0$ by successive approximations there exists a $j$ such that $f$ does not extend node($a$) for all $a$ in $W_{g(i,j)}$. $G_{i+1}$ corresponds to the second part of the game from the proof of Theorem 5.1 (cf. steps (7)–(10) there). Player 1 enumerates $D_i$ and player 2 enumerates $T_i$:

(1) Player 2 enlarges $T_i$ by some finite amount.
(2) Player 1 enlarges $D_i$ by some finite amount.
(3) Player 2 wins instantly if node($|E_2|$) extends a sequence in

$$M(g(i, |D_i|), |T_i|).$$

(4) Player 2 may start a new iteration; in that case the game is continued with step (1).

Player 1 wins iff he does not lose in step (3) and $D_i$ is finite.

This description is incomplete, since we did not specify the winner in case some of the games are blocked. Therefore the actual implementation below is more complicated.

As in the proof of Theorem 5.1, it should follow that player 1 has a winning strategy for all $G_i$ iff player 1 has a winning strategy in $G(A)$.

There is a subtlety in playing the $D_i$'s which is resolved below: Player 1 has to make sure that the condition in step (3) is satisfied in a save way, i.e., node($|E_2|$) may not extend at some later time any node in $M(g(i, |D_i|), |T_i|)$ when player 2 blocks $G_{i+1}$, and therefore freezes $D_i$, but enlarges $E_2$ in $G_0$.

We now present the details for the formulas.

$H_0$ is defined as follows, where the sets $E_1, E_2', L, L_1, L_2, U_1, U_2, X$ belong to player 1 and $E_1', E_2, K_1, \ldots, K_4, V, V_1, V_2, Y, Z$ belong to player 2. Let $F_1, F_2$ be formulas as in the previous subsection and let $r$ denote the code of the empty sequence. The set $L$ is the loop counter, now under control of player 1. $L_1, L_2, K_2, K_3, K_4$ are step counters, and $K_1, X, Y, Z$ are signaling sets, indicating, if nonempty, that player 2 has prepared for step (2), or player 1 tries to win in step (6), in $G_0$, or player 2 tries to win in step (4), in $G_0$, or step (3), in some $G_{i+1}$, respectively. The

$U$'s and $V$'s are used to play a formula Greater.

(9.0)    $|L - K_4| \leq 1 \ \wedge \ L_1, L_2 \subseteq L \ \wedge \ E_2' \subseteq E_2 \ \wedge$

(9.1)    $[ \ Z \neq \emptyset$

(9.2)    $\vee \ \neg(K_1, K_2, K_3, K_4 \subseteq L \ \wedge \ E_1' \subseteq E_1)$

(9.3)    $\vee \ (L \neq \emptyset \ \wedge \ K_1 = \emptyset \ \wedge \ E_2' = E_2)$

(9.4)    $\vee \ (E_1 = \emptyset \ \wedge \ K_1 \neq \emptyset \ \wedge \ |E_2'| \neq r)$

(9.5)    $\vee \ (X \neq \emptyset \ \wedge \ L = L_1 = K_2 = K_3 \ \wedge \ L_2 \neq L \ \wedge \ F_1(E_1', E_2', \ldots))$

(9.6)    $\vee \ ( \ X = \emptyset \ \wedge$

(9.7)        $(Y = \emptyset \ \wedge \ L \neq L_1 \ \rightarrow \ E_1' \neq E_1) \ \wedge$

(9.8)        $(Y \neq \emptyset \ \wedge \ L = L_1 \ \wedge \ K_2 \neq L \ \rightarrow \ F_2(E_1', E_2', \ldots)) \ \wedge$

(9.9)        $(Y = \emptyset \ \wedge \ L = L_1 = K_2 \ \wedge \ K_3 \neq L \ \rightarrow \ E_2' = E_2) \ \wedge$

(9.10)        $(Y = \emptyset \ \wedge \ L = L_1 = K_2 = K_3 \ \rightarrow \ L = L_2) \ \wedge$

        $(Y = \emptyset \ \wedge \ L = L_1 = K_2 = K_3 = L_2 = K_4$

(9.11)            $\rightarrow \ \neg \text{Greater}(V, L, V_1, V_2, U_1, U_2)) \ ) \ ].$

By Corollary 3.8, there uniformly exists, for each $i$, a formula $F_3^i(X, Y, Z, \ldots)$ such that for any canonically given finite sets $X, Y, Z$ player 1 has a winning strategy in the corresponding basic game iff node$(|X|)$ does not extend any sequence in $M(g(i, |Y|), |Z|)$. We assume that the additional set variables are not used elsewhere.

Now we are ready to define the formula $H_{i+1}$. The variables $D_i, K_{i,0}, K_{i,1}, T_i'$, $U_{i,0}, \ldots, U_{i,5}$ belong to player 1 and the variables $D_i', L_{i,0}, L_{i,1}, L_{i,2}, T_i, V_{i,0}, \ldots, V_{i,3}$ belong to player 2. In addition there are the variables $E_2', L, L_1, L_2, Z$ of player 1 and $K_2, K_3, K_4$ of player 2 that were already used in $H_0$. $L_{i,0}$ is the loop counter and $K_{i,0}, K_{i,1}, L_{i,1}, L_{i,2}$ are step counters.

(9.12)    $\neg(L_{i,1}, L_{i,2} \subseteq L_{i,0} \ \wedge \ D_i' \subseteq D_i)$

(9.13)    $\vee \ |L_{i,0} - K_{i,1}| > 1$

(9.14)    $\vee \ (Z = \emptyset \ \wedge \ \text{Greater}(U_{i,0}, L, U_{i,1}, U_{i,2}, V_{i,0}, V_{i,1}))$

(9.15)    $\vee \ [ \ (K_{i,1} \subseteq K_{i,0} \subseteq L_{i,0} \ \wedge \ T_i' \subseteq T_i) \ \wedge$

(9.16)        $(Z = \emptyset \ \wedge \ L_{i,0} \neq L_{i,1} \ \rightarrow \ T_i' = T_i) \ \wedge$

(9.17)        $(Z = \emptyset \ \wedge \ L_{i,0} = L_{i,1} \ \wedge \ L_{i,0} \neq K_{i,0} \ \rightarrow \ D_i' \neq D_i) \ \wedge$

        $(Z \neq \emptyset \ \wedge \ L = L_1 = K_2 = K_3 = L_2 \ \wedge \ K_4 \subseteq L \ \wedge \ K_4 \neq L \ \wedge$

(9.18)        $L_{i,0} = L_{i,1} = K_{i,0} \ \wedge \ L_{i,0} \neq L_{i,2} \ \rightarrow \ F_3^i(E_2', D_i', T_i', \ldots)) \ \wedge$

(9.19)        $(Z = \emptyset \ \wedge \ L_{i,0} = L_{i,1} = K_{i,0} = L_{i,2} \ \rightarrow \ K_{i,1} = L_{i,0}) \ \wedge$

(9.20)        $(Z = \emptyset \ \rightarrow \ \text{Greater}(U_{i,3}, D_i', U_{i,4}, U_{i,5}, V_{i,2}, V_{i,3})) \ ].$

a) Suppose player 1 has a winning strategy in $G(A)$. He translates the moves of the $f$ played in $G(A)$ into the sets $E_1, E_2$, back and forth, as in the proof of Theorem 5.1.

So he enumerates as specified in the formula $H_0$ successive nodes into $E_1$ and double checks the moves of player 2. If player 2 does not enumerate successors of $E_1$ in (9.9), player 1 wins $G_{H_0}$ by making $X \neq \emptyset$ and playing formula (9.5).

On each stage $s$ player 1 plays the sets in all formulas $H_i$ with $i < s$. He always plays the formulas (9.14) and (9.20), and he wins Greater$(U_{i,0}, L, \ldots)$ and

$\text{Greater}(U_{i,3}, D_i', \ldots)$ if $L$ is finite or $D_i'$ is finite, respectively. Therefore if player 2 loses $G_{H_0}$ in (9.5), then $L$ is finite and player 1 wins each $G_{H_{i+1}}$ in (9.14) (if player 2 wants to escape by making $Z \neq \emptyset$, he loses by (9.18)).

By the same reasoning player 2 loses if he makes $Y \neq \emptyset$ and wants to win $G_{H_0}$ by (9.8). He loses this subformula because player 1 made $\text{node}(|E_1|)$ a direct successor of $\text{node}(|E_2|)$. So, in this case player 1 wins $G_{H_0}$ and he wins each $G_{H_{i+1}}$ by (9.14) and (9.18).

Therefore we may assume that player 2 enumerates, as intended, successive nodes into $E_2$. Player 1 always starts a new iteration of $G_{H_0}$, if possible, and enumerates into $E_1$ the node specified by the winning strategy. In addition, player 1 plays the formula $\neg\text{Greater}(V, L, \ldots)$ such that he wins it if $L$ is infinite. If player 2 blocks in some round, player 1 wins $G_{H_0}$ and, by (9.14), each $G_{H_{i+1}}$. We still have to argue that he is able to avoid losing in (9.18) and (9.20).

To this end, if player 2 enumerates $T_i$ in step (9.16), which is double-checked by player 1 as $T_i'$, then player 1 waits in $G_{H_{i+1}}$ and plays $G_{H_0}$ until the length of $\text{node}(|E_2'|)$ in step (9.9) is greater than the length of each node in $M(g(i,d), |T_i'|)$ for all $d \in \omega$ (this is the subtlety alluded to above; note that we need to check only finitely many sets, since $g$ is strictly increasing in $d$ and $M(x,t) = \emptyset$ for $x \geq t$). If this never happens, because player 2 blocks, player 1 wins, as discussed above. Otherwise, player 1 chooses the least $d \geq |D_i|$ such that $\text{node}(|E_2'|)$ does not extend any sequence in $M(g(i,d), |T_i'|)$ and makes $|D_i| = d$. In this way he wins (9.20) since the $f$ played belongs to $A$ and therefore in the end each $D_i$ is finite.

If player 2 tries to win by (9.18) and makes $Z \neq \emptyset$, then player 1 freezes $E_2'$. He has already won $G_{H_0}$ by (9.1). Then he checks if the hypothesis of (9.18) holds. If not, he has won $G_{H_{i+1}}$. If it holds, he applies the winning strategy to the subgame $F_3^i(E_2', D_i', T_i', \ldots)$ and wins it by the choice of $D_i = D_i'$.

Therefore, player 1 wins each $G_{H_{i+1}}$ and the whole game.

b) Suppose player 1 does not have a winning strategy in $G(A)$. Then, using determinacy,[11] player 2 has one. We show that player 2 has a winning strategy in the infinite basic game.

He translates the moves on $f$ in $G(A)$ into the sets $E_1, E_2$, back and forth, as above. In this way player 1 loses $G_{H_0}$ in (9.8) if he does not also enumerate into $E_1$ successive nodes. If player 1 avoids that loss, he will win $G_{H_0}$, but, as we will see in a moment, he will lose one of the $G_{H_{i+1}}$.

On each stage $s$ player 2 plays the sets in all formulas $H_i$ with $i < s$. In particular he always plays the formulas $\text{Greater}(U_{i,0}, L, \ldots)$ in (9.14) and $\text{Greater}(U_{i,3}, D_i', \ldots)$ in (9.20) which he wins if $L$ is infinite or $D_i'$ is infinite, respectively. He also always plays the formula $\neg\text{Greater}(V, L, \ldots)$ in (9.11). He wins it if $L$ is finite. Therefore he forces player 1 to perform infinitely many iterations in $G_{H_0}$, unless player 1 loses prematurely.

On $H_{i+1}$, in step (9.16), player 2 will increase $T_i$ each time he comes along. Each time player 1 has committed the double-checked set $D_i' = D_i$ in (9.17) by making $L_{i,0} = K_{i,0}$, player 2 waits in $G_{H_{i+1}}$ and plays $G_{H_0}$ until the condition $L = L_1 = K_2 = K_3 = L_2$ and $K_4 \neq L$ holds (this happens after player 1 was forced to make $E_2'$ equal to the new value of $E_2$ in step (9.9) and refrained from playing (9.5)). Then he checks whether $\text{node}(|E_2|)$ extends a node in $M(g(i, |D_i'|), |T_i'|)$. If

---

[11]We could, with a little bit more work, avoid using this hypothesis and show that player 1 has a winning strategy in $G(A)$ if he has one in the basic game.

this is the case he makes $Z \neq \emptyset$ (if not already done for some other $H_{j+1}$) and wins the game on $F_3^i(E_2', D_i', T_i', \ldots)$ in (9.18) and therefore the game on $H_{i+1}$ and the whole game.

Now suppose player 1 avoids losing prematurely in (9.8) and (9.18), and that he does not block; he would also lose. Then in each $G_{H_j}$ infinitely many iterations are performed and, since the $f$ played does not belong to $A$, player 1 is forced to make one of the $D_i$ infinite, say for $i = i_0$. But then player 2 wins in the end formula (9.20) which falsifies $H_{i_0+1}$ and wins the whole game for player 2.

The detailed way of how to play the sets does not need any new ideas and is left to the reader.

## Acknowledgement

## References

1. Wilhelm Ahrens. *Mathematische Unterhaltungen und Spiele.* B. G. Teubner, Leipzig, 1901.
2. Jon Barwise (Editor). *Handbook of mathematical logic.* North-Holland, Amsterdam, 1977. MR0457132 (56:15351)
3. Elwyn R. Berlekamp, John H. Conway, Richard K. Guy. *Winning ways.* 2 Vols., Academic Press, London, 1982. MR0654502 (84h:90091b)
4. Allan Borodin, Ran El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, Cambridge, 1998. MR1617778 (2000g:68053)
5. J. Richard Büchi, Lawrence H. Landweber. *Solving sequential conditions by finite-state strategies.* Trans. Amer. Math. Soc. **138** (1969), 295–311. MR0280205 (43:5926)
6. James P. Carse. *Finite and infinite games.* Free Press, New York, 1986.
7. Douglas A. Cenzer, Jeffrey B. Remmel. *Recursively presented games and strategies.* Math. Social Sci. **24** (1992), 117–139. MR1189721 (93m:03078)
8. Morton D. Davis. *Infinite games of perfect information.* Ann. of Math. Stud., vol. 52, 1964, pp. 85–101. MR0170727 (30:965)
9. Manfred Eigen, Ruthild Winkler. *Das Spiel.* Piper Verlag, Munich, 1975. English Translation: *Laws of the game.* Knopf, New York, 1981.
10. Shimon Even, Robert E. Tarjan. *A combinatorial problem which is complete for polynomial space.* J. ACM **23** (1976), 710–719. MR0436659 (55:9600)
11. Erich Grädel, Wolfgang Thomas, Thomas Wilke (Editors). *Automata, logics, and infinite games.* Lecture Notes in Computer Science, vol. 2500, Springer-Verlag, Berlin, 2002. MR2070731 (2005b:68009)
12. Yuri Gurevich. *Infinite games.* Bull. Eur. Assoc. Theor. Comput. Sci. EATCS **38** (1989), 93–100.
13. Wilfred Hodges. *Building models by games.* Cambridge University Press, Cambridge, 1985. MR0812274 (87h:03045)
14. Johan Huizinga. *Homo ludens: proeve eener bepaling van het spel-element der cultuur.* H. D. Tjeenk Willink, Haarlem, 1938. English Translation: *Homo ludens: a study of the play-element in our culture.* Routledge & K. Paul, London, 1949.
15. Thomas John. *Recursion in Kolmogorov's R-operator and the ordinal $\sigma_3$.* J. Symbolic Logic **51** (1986), 1–11. MR0830066 (87i:03094)
16. James P. Jones. *Some undecidable determined games.* Internat. J. Game Theory **11**(2) (1982), 63–70. MR0686391 (84b:90107)
17. Alexander S. Kechris. *On Spector classes.* Cabal Seminar 76–77. Lecture Notes in Mathematics vol. 689, Springer-Verlag, Berlin, 1978, pp. 245–277. MR0526922 (81b:03053)

18. Bakhadyr Khoussainov, Anil Nerode. *Automata theory and its applications.* Birkhäuser, Boston, 2001. MR1839464 (2004e:03073)

19. Martin Kummer, Matthias Ott. *Effective strategies for enumeration games.* Interner Bericht 1995, 41. Fakultät für Informatik, Universität Karlsruhe, 1995. Available from: URL: `http://www.ubka.uni-karlsruhe.de/cgi-bin/psview?document=ira/1995/41`. Also published in: Computer Science Logic, CSL'95, Lecture Notes in Computer Science, vol. 1092, Springer-Verlag, Berlin, 1996, pp. 368–387. MR1461887 (98c:03026)

20. Alistair H. Lachlan. *On the lattice of recursively enumerable sets.* Trans. Amer. Math. Soc. **130** (1968), 1–37. MR0227009 (37:2594)

21. Alistair H. Lachlan. *The elementary theory of recursively enumerable sets.* Duke Math. J. **35** (1968), 123–146. MR0227008 (37:2593)

22. Alistair H. Lachlan. *On some games which are relevant to the theory of recursively enumerable sets.* Ann. of Math. **91**(2) (1970), 291–310. MR0284333 (44:1562)

23. Manuel Lerman. *Degrees of unsolvability.* Springer-Verlag, Berlin, 1983. MR0708718 (85h:03044)

24. Edouard Lucas. *Récréations mathématiques.* 4 Vols., Gauthier-Villars, Paris, 1881–1894.

25. Donald A. Martin. *Borel determinacy.* Ann. of Math. **102**(2) (1975), 363–371. MR0403976 (53:7785)

26. Donald A. Martin. $\Pi^1_2$ *monotone inductive definitions.* Cabal Seminar 77–79. Lecture Notes in Mathematics, vol. 839, Springer-Verlag, Berlin, 1981, pp. 215–233. MR0611175 (83g:03047)

27. Donald A. Martin, Robert M. Solovay. *Unpublished notes.* 1975. Communicated by Alexander S. Kechris.

28. Yuri V. Matijacevič. *Enumerable sets are diophantine.* Dokl. Akad. Nauk. SSSR **191**(2) (1970), 279–282. (in Russian)

29. John Maynard Smith. *Evolution and the theory of games.* Cambridge University Press, Cambridge, 1982.

30. Yiannis N. Moschovakis. *Descriptive set theory.* North-Holland, Amsterdam, 1980. MR0561709 (82e:03002)

31. Sylvia Nasar. *A beautiful mind.* Simon & Schuster, New York, 1998. MR1631630 (99j:01021)

32. John von Neumann, Oskar Morgenstern. *Theory of games and economic behavior.* Princeton University Press, Princeton, 1944. MR0011937 (6:235k)

33. Friedrich Nietzsche. *Menschliches, Allzumenschliches; Zweiter Band.* E. W. Fritzsch, Leipzig, 1886. English Translation by R. J. Hollingdale: *Human, all too human.* Cambridge University Press, Cambridge, 1986.[12]

34. Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley, Reading, 1994. MR1251285 (95f:68082)

35. Michael O. Rabin. *Effective computability of winning strategies.* Ann. of Math. Stud., vol. 39, 1957, pp. 147–157. MR0093740 (20:263)

36. Hugo Rahner. *Der spielende Mensch.* Johannes Verlag Einsiedeln, Freiburg, 1952. English Translation: *Man at play.* Burns & Oates, London, 1965.

37. Hartley Rogers, Jr. *Theory of recursive functions and effective computability.* McGraw-Hill, New York, 1967. MR0224462 (37:61)

38. Thomas J. Schaefer. *On the complexity of some two-person perfect-information games.* J. Comput. and System Sci. **16** (1978), 185–225. MR0490917 (80e:90131)

39. Bernhard Schäfers (Editor). *Grundbegriffe der Soziologie.* 5th Edition, Leske + Budrich, Opladen, 1998.

40. Josef Seifert. *Schachphilosophie.* Wissenschaftliche Buchgesellschaft, Darmstadt, 1989.

41. Robert I. Soare. *Recursively enumerable sets and degrees.* Springer-Verlag, Berlin, 1987. MR0882921 (88m:03003)

42. Ludwig Wittgenstein. *Philosophische Untersuchungen.* In: Schriften 1, Suhrkamp, Frankfurt am Main 1960. English Translation: *Philosophical investigations.* Blackwell, Oxford, 1953.

INIT GmbH, Kaeppelestrasse 6, D-76131 Karlsruhe, Germany
*E-mail address*: `mkummer@init-ka.de`

---

[12]Our motto is translated there (p. 277) as follows: *We think that play and fairy tales belong to childhood: how shortsighted that is! As though we would want at any time of life to live without play and fairy tales!*