JOURNAL OF COMBINATORIAL THEORY, Series B 54, 257–290 (1992)

On the Strength of König's Duality Theorem for Infinite Bipartite Graphs

RON AHARONI*

The Technion, Haifa, 32000, Israel

MENACHEM MAGIDOR†

Hebrew University of Jerusalem, Jerusalem, 91904, Israel

AND

RICHARD A. SHORE‡

Cornel University, Ithaca, New York 14853

Communicated by the Managing Editors

Received March 15, 1989

We prove that König's duality theorem for infinite graphs (every graph G has a matching F such that there is a selection of one vertex from each edge in F which forms a cover of G) is inherently of very high complexity in terms of both the methods of proof it requires and the computational complexity of the covers it produces. In particular, we show that there is a recursive bipartite graph such that any cover as required by the theorem is highly non-computable; indeed it must be above (in Turing degree) all the recursive iterations of the Turing jump. This implies that the theorem is proof theoretically at least as strong as the system ATR_0 which is known to be strictly stronger than compactness or König's lemma. Thus the theorem cannot be proven by elementary means plus compactness. Transfinite methods are actually necessary. The actual cover given by the proof considered is seen to have an additional maximality property which makes the assertion of its existence imply a stronger system, $\Pi^1_1\text{-CA}_0$. We refine this known proof of König's theorem to show that in fact its consequences are equivalent to $\Pi^1_1\text{-CA}_0$.

* Research partially supported by Japan TS Research Fund.

† Research partially supportyed by Grant 87-00040/1 from the U.S.–Israel Binational Science Foundation.

‡ Research partially supported by NSF Grant DMS-8601048 and Grant 84-000067 from the U.S.–Israel Binational Fund. The hospitality of the Mathematics Departments of Ben-Gurion and Hebrew Universities is also gratefully acknowledged.

0095-8956/92 $3.00

1. Introduction

Lovasz and Plummer [13] cite what they call the König Max–Min theorem (the matching and covering number are equal in (finite) bipartite graphs) as the most important and basic theorem in matching theory. (A graph G is *bipartite* if its set of vertices can be divided into two disjoint sets M and W such that every edge in the graph joins (that is, consists of) a vertex in M and a vertex in W. A *matching* F in G is a collection of disjoint edges of G. The *matching number* of a graph G is the maximum cardinality of a matching in G. A *cover* of a graph G is a set C of vertices of G such that every edge contains a vertex in C. The *covering number* of a graph G is the minimal cardinality of a cover of G.) This theorem was proved for all finite bipartite graphs in König [12]. Erdős conjectured that a strong form of this theorem holds for all bipartite graphs regardless of cardinality. Note that to give the theorem more substantial content for infinite graphs, we assert not just the equality of the cardinalities of the matching and covering numbers but rather the existence of a (necessarily maximal) matching from which we may choose a cover. We call a cover of G a *König cover* if it consists of a selection of one vertex from each edge of a matching in G. As, for example, in Brualdi [3] and Aharoni [2] we call this version the König duality theorem:

Theorem 1.1 (König's duality theorem). *In every bipartite graph G there is a matching F and a selection of one vertex from each edge in F which produces a cover C of G, i.e., every bipartite graph has a König cover.*

The proof of the original theorem for finite graphs is quite ingenious but still fairly short and certainly completely elementary (in the technical sense which we shall make precise later that refers to the methods needed to establish the result). Now most theorems of this sort in finite graph theory which generalize to infinite graphs have fairly simple proofs in the general case. Simple here can be understood in a number of different ways which we shall also make precise later. First consider the case of graphs on the natural numbers and the degree of difficulty of constructing the desired matching and cover in the case of König's theorem or the required decomposition in Dilworth's theorem, the homogeneous set in Ramsey's theorem or the coloring in various chormatic number problems. (We shall use the word "solution" to stand ambiguously for such a desired matching, decomposition, coloring, etc.)

Generally speaking, a combinatorial or graph theoretic proof requiring only a direct analysis or construction can be extended to produce a recursive solution, i.e., one effectively computable from the inputs. An example of such a theorem which is nonetheless certainly nontrivial is the Dushnik–Miller theorem that an ordered set can be embedded in a product of n chains if there are n linear orderings on the set whose intersection is the given order. In this example the n chains required can be chosen to be recursive if the n linear orderings on the set in the hypothesis are recursive. Similarly, Tverberg's proof of the infinite version of Brook's theorem on coloring graphs with certain properties with at most k colors supplies a greedy algorithm that establishes Schmerl's recursive version of the theorem for strongly recursive graphs (ones in which the neighbors of a vertex can be computed effectively). (See Schmerl [18] and Tverberg [26].)

The other common, relatively simple proof technique used to lift results from the finite to the infinite is compactness of König's lemma. This method is more powerful than a simple direct elementary construction and typically signals the need for more complicated algorithms. Almost invariably, a proof that requires some version of compactness of König's lemma corresponds to a construction procedure that produces the desired solution recursively in $\mathbf{0}'$, the degree of unsolvability of the halting problem. At times a more careful construction will produce a recursive solution to the recursive problem but not always. Thus, for example, the decomposition required in Dilworth's theorem (every ordered set with width at most n is the union of at most n chains) is proved to exist by an application of König's lemma and can be easily constructed (for recursive graphs) recursively in $\mathbf{0}'$. As is shown in Schmerl [17], however, there is in general no recursive solution. (For more on these matters see Kierstead [10] and Kierstead *et al.* [11] and the references therein.)

The homogeneous sets required in Ramsey's theorem also require a compactness or König's lemma type argument. In fact, the situation here is a bit more complicated than that for Dilworth's theorem. In the latter theorem we can manage with binary trees in the application of König's lemma, while the former requires the lemma for all finitely branching trees. This division (which will be explained in more detail as Systems 2.2 and 2.3) corresponds to a recursion of complexity theoretic one: One can always find a decomposition of a recursive graph as required in Dilworth's theorem which is strictly recursive in $\varnothing'$. On the other hand, there are recursive colorings of the triples of natural numbers which have homogeneous sets only of degree $\varnothing'$ [8].

The situation for König's duality theorem seems quite different from that for the other combinatorial results mentioned so far. After a series of partial results by various authors including the case of countable graphs by Podewski and Steffens [15], the full theorem was finally proven in Aharoni [2]. The proofs, however, even in the countable case, are very difficult and highly non-elementary. They involve uses of Zorn's lemma, even for graphs on well-ordered sets such as the natural numbers, and arbitrarily long transfinite inductions (of all countable lengths again just for the case of graphs on the natural numbers). The constructions, moreover, give no somple or natural bounds on the complexity of the desired solution.

We shall analyze the complexity of this result both in terms of how hard it is to construct the desired covers and what methods are needed to prove the theorem. Since we are hoping to reach readers with a variety of backgrounds and interests in both combinatorics and logic, we shall try to supply the background material in both areas. In addition, we shall separate the combinatorial and complexity (or recursion theoretic) material from the more logically (or proof theoretically) oriented material so as to make the former independent of the latter.

Our first goal in this paper is to characterize in a formally precise way the complexity of constructing, for recursive graphs, the covers that the theorem asserts exist. The complexity of these covers will be shown to be very high. As a first approximation to their complexity we shall see that it exceeds not only that of the halting problem but also of all finite iterations of the halting problem. Equivalently, finding such covers is harder than answering all first-order questions about arithmetic. Actually the complexity of the problem far exceeds that of first-order arithmetic. We can iterate the halting problem (or equivalently, starting with the recursive sets, the operations of projection, complementation, and recursive union) into the transfinite along any recursive well ordering and still be below the complexity of constructing such covers.

The recursion theoretic hierarchies used to measure such complexity will be described and explained in the first four pages of Section 2. We shall first review the definition of the standard halting problem familiar from undecidability results. It corresponds to a single application of the (Turing) jump operator. We shall then explain the transfinite iterations of this jump operator which correspond to the hyperarithmetic hierarchy that we use to measure degree theoretic complexity. (This hierarchy is essentially an effective analog of the Borel hierarchy for sets of natural numbers.) Section 4 (up to Theorem 4.13) will describe the constructions of various recursive graphs on $\mathbb{N}$ and show that their covers must be of very high complexity in a way that will quite precisely characterize the difficulty of constructing the covers guaranteed to exist by König's duality theorem for countable graphs. To indicate the inherent complexity of the theorem in the uncountable case, we shall describe a recursive graph on the reals which has no projective solution. These parts of the paper can be read independently of the rest of the paper (the rest of Section 4, Section 3, and the bulk of Section 2). These other sections will be devoted to a second measure of complexity: the strength of the axioms needed to prove the existence of the desired solution to a given combinatorial problem in general (Section 2) and for König's duality theorem and related results in particular (Section 3 and the rest of Section 4).

The general program of calibrating the strength of mathematical theorems of the axiomatic systems needed to establish them (and, more interestingly, determining the systems to which they are actually equivalent) is now called *reverse mathematics.* Although many people have made important contributions to this area, it owes its existence and fruitfulness primarily to Harvey Friedman and Stephen Simpson. In Section 2 we give a brief description of some of the main axiomatic systems of induction and comprehension used to measure this type of complexity. We also briefly survey the relations between these systems and some of the basic theorems of algebra and analysis. For a guide to the details as well as a more comprehensive view, we recommend the relevant articles in Harrington *et al.* [4] and Simpson [21] and the references therein.

Section 3 will be devoted to proving König's duality theorem for countable graphs. It begins with a standard proof which illustrates the use of higher order axioms in the various known proofs of the theorem. Indeed this standard proof falls entirely outside the scope of all the systems discussed in Section 2. We then combine some recursion theoretic techniques (basis theorems) with a finer analysis of this proof to produce a proof of the theorem for graphs on $\mathbb{N}$ in one of the standard systems described in Section 2 and so a general bound on the proof theoretic complexity inherent in solutions to this problem.

Finally, in the concluding parts of Section 4 we apply our complexity theoretic results for solutions to König's theorem to characterize precisely the proof theoretic strength of various results implicit in the known proofs of the theorem and to give a nearly optimal lower bound for the strength of the theorem itself in terms of these axiom systems. In particular, we shall prove that elementary means supplemented by the compactness theorem or König's lemma do not suffice to prove the duality theorem or any of a number of related results in matching theory. The reader interested primarily in the question of how hard it is to construct the covers can ignore these parts of the paper and the proof theoretic applications. We do feel, however, that they have some very interesting things to say about the theorem and associated results.

We now preview our results for those readers who are familiar with the standard recursion hierarchies and the common proof theoretic systems for second-order arithmetic. The constructions of Section 4 will show that there is a recursive bipartite graph such that any König cover for it must be above all the hyperarithmetic sets in Turing degree. This result is essentially the best possible. The desired cover is defined by an arithmetic condition and so the standard basis theorems tell us that there is always one recursive in Kleene's $\mathcal{O}$ (of strictly lower hyperdegree, etc.). (An analogous construction for the reals produces a recursive graph on $\mathcal{R}$ for which any solution must be more complicated then all the projective sets.) In proof theoretic terms, the recursion theoretic result immediately shows that König's theorem implies ATR_0 and so is strictly stronger than compactness.

On the other hand, the proof presented in Section 3 shows (with some additional analysis) that the theorem is provable in Π^1_1-CA_0. Now, as stated, the theorem is a Π^1_2 sentence and so, on general grounds, cannot imply Π^1_1-CA_0. We shall see, however, that the proof supplied in Section 3 actually proves a somewhat stronger prescribing a type of maximality property of the solution. The stronger version of the theorem implicit in the proof will be seen to require solutions of degree at least that of $\mathcal{O}$ for some recursive graphs. It will therefore be precisely equivalent to the theory Π^1_1-CA_0. Thus the only remaining question is whether the original version of the theorem is provable in ATR_0. This question seems non-trivial and of combinatorial interest. In particular, Lemma 3.2 asserts the existence of a maximal matching of a certain sort. This lemma alone guarantees the existence of $\mathcal{O}$ and so is equivalent to Π^1_1-CA_0. As some such maximal matching is used in all the known proofs of König's duality theorem any proof in ATR_0 would seem to require a new approach. The various related maximal matching results are also connected with attempts to generalize to infinite families Hall's theorem characterizing when a finite family of non-empty sets has an injective choice function. In Section 4 we describe one from Podewski and Steffens [15] (the existence of maximal representable subfamilies) which is sufficient for their proof of König's duality theorem for countable graphs and show that it too implies Π^1_1-CA_0 (Theorem 4.25). It would be interesting to see which of the known maximal matching results are, in fact , equivalent to Π^1_1-CA_0.

The situation in the uncountable case is harder to analyze as there are no systems of third-order arithmetic corresponding to the hierarchy of second-order systems used in the countable case. The first point to notice is that the full theorem for all graphs easily implies the axiom of choice (Proposition 3.6). The known proofs do not use simply a well ordering of the graph being analyzed but also of higher level objects as they apply Zorn's lemma to families of subsets of the graph. That a well ordering of the vertices and edges is not sufficient is seen both from the analysis of recursive graphs on $\mathbb{N}$ (which are automatically well ordered) and the one constructed on $\mathcal{R}$. The point made by the construction for $\mathcal{R}$ is that even assuming a definable well ordering of $\mathcal{R}$ such as exists in L brings us no closer to a definable solution. This need for a choice principle at one level higher than the graph being analyzed is circumvented in the special case of $\mathbb{N}$ by the use of a basis theorem. Perhaps the point here is that Π^1_1-CA_0

(indeed even ATR_0) implies such as a choice principle (Σ_1^1-AC) which we, in fact, use in the proof of the theorem in Π_1^1-CA_0.

2. Recursion Theoretic Hierarchies and Proof Theoretic Systems

In this section we try to give brief descriptions and explanations of both the recursion and proof theoretic measures of complexity that we use to calibrate problems and something of a guide to the literature. Both views are intimately connected with syntactic and definability measures in first- and second-order arithmetic and so we have to deal with these concepts as well. We begin with the computational approach.

Our starting point is the notion of a recursive function or set (we often identify sets with their characteristic functions). We say that a partial function $\varphi\colon \mathbb{N} \to \mathbb{N}$ (that is one whose domain may be any subset of $\mathbb{N}$) is a *partial recursive function* if it is theoretically computable by a Turing machine or, equivalently, by your favorite general purpose computer. (By "theoretically" we just mean that we permit calculations that may require arbitrarily large, although always finite, time and memory.) If φ is total, we say that it is a *recursive function.* We adopt the view expressed by Church's thesis which identifies these functions with the intuitively effectively calculable ones. (For more details and further information and references on the recursion theory we need we refer the reader to Rogers [16].)

It is easy to believe that we can recursively enumerate the possible programs for such a machine and so have an effective list φ_e of the partial recursive functions. (Not all programs will compute total functions.) We say that e is an index for the partial recursive function φ_e. Such an index should be thought of simply as a code for the finite set of instructions that make up the program for calculating the corresponding function. That we have listed the partial recursive functions rather than the total ones is crucial. There is no obvious way of deciding if a given program halts on any or all inputs. This is the famous halting problem which supplies the basic example of a non-recursive set: $K = \{\langle x, y\rangle \mid \varphi_x(y) \text{ is convergent}\}$. The proof that K is non-recursive is the typical diagonal argument: If K were calculable, the function φ would also be calculable where φ is given by $\varphi(n) = \varphi_n(n) + 1$ if $n \in K$ and 0 otherwise. If this function φ were calculable, it would be one of the ones on our list, say φ_e. Whether we assume that $e \in K$ or not, we quickly obtain a contradiction. Thus φ, and so K, is not recursive.

We can now lay out a sequence beginning with K of progressively more and more complicated sets of "relativizing" the halting problem and then iterating this procedure. By *relativization* (to A) we mean a process which adds to our basic machine a black box or *oracle* (A). So equipped, our machines can generate questions of the form "is z in A" and in a single step receive (from the oracle for A) the correct answer. Thus we convert our list $\langle \varphi_e \mid e \in \mathbb{N} \rangle$ of all partial recursive functions φ_e to one $\langle \varphi_e^A \mid e \in \mathbb{N} \rangle$ of all *functions partial recursive in A*. The ones *recursive in A* (and so intuitively the ones calculable from A) are simply the total functions on this list. If $B = \varphi_e^A$ for some e, we say that B is *Turing reducible to* (or simply *recursive in*) A and write $B \leqslant_T A$. The *Turing degrees*, or simply the *degrees*, are just the equivalence classes $\mathbf{a}$, $\mathbf{b}$ (with the induced ordering which is also written $\mathbf{b} \leqslant_T \mathbf{a}$) of sets under this ordering. We can now relativize the halting problem to an arbitrary set A by considering the set of convergent computations from machines with oracle A. This defines an operation on sets, called the *Turing jump*, which takes a set A to $A' = \{\langle x, y \rangle \mid \varphi_x^A(y)$ is convergent$\}$. The unsolvability of the halting problem (in relativized form) says then that, for every A, A' is not recursive in A. (The proof is exactly as for K, except that we replace everything by its relativized version: "recursive in A" for "recursive," φ_e^A for φ_e, etc. This procedure of relativizing proofs is almost always routine and we shall use it frequently.) As A is clearly recursive in A', we have an operation that, by iteration, gives us an increasing sequence of more and more complicated sets. Beginning with the empty set $\varnothing$, we have the sequence of jumps $\varnothing$, $\varnothing'$, $\varnothing''$, $\varnothing^{(3)}$, $\varnothing^{(4)}$, ... (We write $A^{(n)}$ for the result of applying the jump operation n times to A.)

Our next step is to continue the iteration of the jump operator into the transfinite along recursively given well orderings. It is clear that if we have a set A at level α of such an ordering we can form one at level $\alpha + 1$ by simply applying the jump to get A'. The question is what to do at limit levels. Our answer is to take recursive unions. To be precise, if we have a sequence of sets $\langle A_i \mid i \in I \rangle$ for a recursive index set I, we let the *recursive union*, $\bigoplus \{A_i \mid i \in I\}$, of the sequence be the set $\{\langle i, x \rangle \mid x \in A_i \wedge i \in I\}$. ($\langle i, x \rangle$ is simply the ordered pair with first element i and second element x. It can also be viewed as a recursive function giving a code for the pair from the two inputs i and x. Similarly, we use $\langle x_1, ..., x_n \rangle$ to denote the appropriate ordered n-tuple or its recursive code.) When the index set I is all of $\mathbb{N}$ or otherwise clear from the context, we write $\langle A_i \rangle$ and $\bigoplus A_i$ for $\langle A_i \mid i \in I \rangle$ and $\bigoplus \{A_i \mid i \in I\}$, respectively.

Recursive unions are closely connected to a notion that is basic to this area: *uniformity*. Roughly speaking, a procedure is uniform if there is a recursive operation that takes indices for the input of the procedure to indices for the output. (Remember that this means simply that there is an effective way of converting programs that calculate the input objects such as sets, graphs, or orderings into programs that compute the corresponding output objects.) Thus, for example, a sequence $\langle A_n \rangle$ of recursive sets is *uniformly recursive* if there is a recursive function f such that $f(n)$ is an index for A_n, i.e., $\varphi_{f(n)}$ is the characteristic function of A_n. (Note that this is equivalent to the predicate $x \in A_n$ being a recursive predicate of x and n.) Similarly, we say that $\bigoplus A_i$ is a *uniform upper bound* (in Turing degree) for the A_i, that is, all the A_i are uniformly computable from it: to see if $x \in A_i$ just ask if $\langle i, x\rangle \in \bigoplus A_1$. The point of the uniformity here is that the index for A_i as a set recursive in $\bigoplus A_i$ is clearly given by a recursive function of i. In other words, the sequence $\langle A_i\rangle$ is uniformly recursive in $\bigoplus A_i$.

If we apply this procedure of taking recursive unions to the sequence of jumps $\langle \varnothing^{(n)}\rangle$ we obtain $\varnothing^{(\omega)} = \{\langle n, x\rangle \mid x \in \varnothing^{(n)}\}$. We can, by using recursive unions at limit levels in this way, iterate the jump along any recursive well ordering. Again, to be precise, a well ordering (of $\mathbb{N}$) is just a certain type of binary relation on $\mathbb{N}$ and so we view it as a set of pairs of natural numbers. The ordering is recursive if the corresponding set of pairs is recursive. For technical convenience we also require that in a *recursive well ordering* the least element of the ordering is 0, the successor function, s, is recursive as are the domain D of the ordering and the set L of numbers at limit levels of the ordering. We let $\mathcal{O}$ be the set of indices of all such recursive well orderings. Given such a well ordering $<_e$ we can iterate the jump along it in the obvious way: $J(e, 0) = \varnothing$, $J(e, s(x)) = J(e, x)'$ for every x in the domain D of s and $J(e, j) = \bigoplus \{J(e, x) \mid x <_e y\}$ for $y \in L$. The sets constructed in this way are usually referred to as *H-sets* with the intention being that, if $e \in \mathcal{O}$, then $H_e = \bigoplus \{J(e, x) \mid x$ is in the domain of $<_e\}$. This hierarchy is called the hyperarithmetic hierarchy. The sets in it are our standards of comparison for complexity beyond the sets definable in (first-order) arithmetic. $\varnothing^{(\omega)}$ is simply H_e for e an index of the standard ordering of $\mathbb{N}$. It is of the same degree as the truth set of arithmetic, that is, the set of all true sentences of arithmetic.

As with the jump itself, relativization gives us a whole hierarchy of operators. Here the hierarchy itself depends on the relativization. We can define the H-sets relative to X for any well ordering recursive in X. We let $\mathcal{O}^X = \{e \mid e$ is a recursive index relative to X of a well ordering$\}$. H_e^X is then defined as were the unrelativized H_e, except that we begin, as one would expect, with $H_0^X = X$. (These operators also define a reducibility ordering A *is hyperarithmetic in* B, $A \leqslant_h B$, iff $\exists e \in \mathcal{O}^B (A \leqslant_T H_e^B)$.) The set $\mathcal{O}^X$ itself is actually (Turing) above all the H^X-sets. Combinatorially, its complexity is precisely that of deciding if any given (recursive in X) partial ordering is well founded and, if not, of constructing an infinite descending chain in the ordering. It is with this set that we actually come to the end of our recursion theoretic hierarchies and complexity measures. Before going on to the proof theoretic systems, however, we offer two other views of this hierarchy. The first is phrased in terms of iterations of complementation, projection, and effective union in analogy with the Borel hierarchy. The second is defined in terms of quantification in arithmetic.

Combinatorially, this whole hierarchy can be viewed as an effective analog of the Borel sets for sets of (finite sequences of) natural numbers. The recursive sets are given at the bottom in analogy with the open sets. The jump operator corresponds to projection, i.e., if some set P of n-tuples has gotten in at level α of the hierarchy then its projection $\{\langle x_2, ..., x_n\rangle : \exists x_1(\langle x_1, x_2, ..., x_n\rangle \in P)\}$ is put in at level $\alpha + 1$. We also, as for the Borel sets, put in its complement, all n-tuples not in P. To help see that this corresponds to taking the jump note that we can characterize the jump operator as follows: $\langle n, m\rangle \in A'$ iff there are s, x, and y such that $\langle x, y\rangle \in R_{n,m,s}$, $D_x \subset A$ and $D_y \subset \bar{A}$. Here D_z is the finite set with index z in some canonical recursive coding of the finite sets and $\langle x, y\rangle \in R_{n,m,s}$ is the following recursive relation: the calculation of $\varphi_n^A(m)$ converges in at most s many steps during which the membership questions asked of the oracle which get positive and negative answers constitute the sets D_x and D_y, respectively. Thus if we have the recursive predicates and ones for A and $\bar{A}$ at level α we should obtain A' (and $\bar{A}'$) at level $\alpha + 1$ by projection (and then complementation). (The technical details involve some additional arguments that we omit.) At limit levels we, of course, take effective unions.

Thus to establish some property for all the H-sets it intuitively suffices to prove that the collection of sets with this property contains all recursive sets and is closed under complementation, projection (or the Turing jump), and recursive infinitary union or disjunction. Actually, these recursive unions are tied to the notions of uniformity in the constructions of our hierarchy. Formally, everything is labeled with indices and we take unions over sequences given by recursive lists of indices. In fact, however, there is a standard procedure called *effective transfinite induction* which guarantees that, if these closure properties can be proven uniformly in the indices, then one can conclude that every H-set has the desired property. This type of argument is an application of the recursion theorem. We need to know only that there is such general procedure for the applications in this paper. The reader who wishes to see the details of how it is defined and applied is referred to Rogers [16, Chap. XI].

We turn now to arithmetic and a more syntactic view of the hyperartihmetic hierarchy. It is not necessary to follow this presentation to understand the complexity theoretic results in Section 4. The reader so wishing can therefore go directly to Section 4. We shall, however, use the notations introduced here to describe the proof-theoretic hierarchies that we need later.

In trying to understand the definitional complexity of the H-sets one has natural problems corresponding to each of the first few levels of this hierarchy. Thus ϕ'' is the degree of $\{e \mid \varphi_e \text{ is total}\}$ and ϕ''' is the degree of $\{e \mid \varphi_e \text{ has cofinite domain}\}$. The unaided imagination soon runs out in this game, however. Perhaps the best uniform way to describe the complexity of the iterations of the jump is in terms of the syntactical complexity of their definitions (in terms of the number of quantifiers used) and of the questions about arithmetic that they can enswer.

We begin with a (first-order) language for arithmetic that has constant symbols for 0 and 1, binary function symbols for $+$ and $\cdot$, a binary predicate for $<$, and quantification over $\mathbb{N}$. For technical convenience we also introduce abbreviations for bounded quantifiers. We use $\exists x < y$ and $\forall x < y$ to mean "there is an x less than y such that" and "for all x less than y," respectively. We define a hierarchy of formulas beginning with the ones with only bounded quantifiers which are called both Σ_0^0 and Π_0^0. We then continue inductively: if φ is Σ_n^0 (Π_n^0) then $\forall x\varphi$ ($\exists x\varphi$) is Π_{n+1}^0 (Σ_{n+1}^0). A set A is said to be Σ_n^0 (Π_n^0) if it is definable by a Σ_n^0 (Π_n^0) formula. It is Δ_n^0 if it is both Σ_n^0 and Π_n^0. Simple manipulations show that every *arithmetical formula* (i.e., one of this language for arithmetic) is equivalent to one which is Σ_n^0 or Π_n^0 for some n. There is an interesting and important fact (due to Kleene and Post) connecting these definability notions with complexity as measured by the Turing jump: A set A is Δ_{n+1}^0 if and only if it is recursive in $\varnothing^{(n)}$ (which is itself Σ_n^0 but not Π_n^0). Thus there is a precise correspondence between the iterations of the Turing jump and the levels of definability of sets in arithmetic.

To understand the higher levels of the hyperarithmetic hierarchy and its ultimate scope, we must move beyond first-order arithmetic to second-order arithmetic, i.e., to quantification over sets and functions rather than just over numbers. We add to our language variables X, Y, Z ranging over subsets of $\mathbb{N}$ and a predicate $\in$ for membership. Once again we define a hierarchy of formulas beginning this time with $\Sigma_0^1 = \Pi_0^1 =$ the collection of all Σ_n^0 and Π_n^0 formulas for $n \in \mathbb{N}$. A formula is Σ_{n+1}^1 (Π_{n+1}^1) if it is of the form $\exists X\varphi$ ($\forall X\varphi$), where the quantification ranges over subsets X of $\mathbb{N}$ and φ is Π_n^1 (Σ_n^1). The conventions for sets of numbers being Σ_n^1, Π_n^1, Δ_n^1 are as in the first-order case. The important and basic result of Kleene connecting this hierarchy with the H-sets is that the Δ_1^1 sets are precisely the *hyperarithmetic* ones, i.e., those sets recursive in some H-set H_e. Kleene's $\mathcal{O}$, the set of e which are indices for recursive well orderings, is a Π_1^1 (but not Σ_1^1) set. Indeed, it is complete at this level, i.e., every Π_1^1 set (and so every hyperarithmetic set) is recursive in it. (To say that an ordering has no infinite descending chains is clearly Π_1^1. That $\mathcal{O}$ is not Σ_1^1 follows its completeness and a typical diagonal argument.)

We turn now to proof theory, in particular, to axiom systems in the language introduced above for second-order arithmetic which are suited to the development of much of ordinary mathematics (finite and countable combinatorics, algebra, and analysis). For philosophical considerations, further technical details, specific references for the various results cited below and other systems we refer the reader to the expository papers Simpson [19–22] from which most of what follows has been extracted. (For a comprehensive treatment of all these matters we recommend Simpson [23].)

All our systems contain the basic axioms for $+$, $\cdot$, and $<$, saying that $\mathbb{N}$ is an ordered semi-ring. In addition, they all include the *induction axiom* which says that one can do inductions over any set (that one knows to exist):

$$\text{(Induction) } 0 \in X \wedge \forall n(n \in X \rightarrow n+1 \in X) \rightarrow \forall n(n \in X).$$

We call the system consisting of these axioms P_0. The crucial information missing from P_0 is anything that tells us that there are any sets X of natural numbers. (The induction axiom with its free variable X can be usefully applied only after we know that some set at least exists.) All the systems we consider will be defined by adding various types of set existence axioms to P_0. They will, in addition, all be subsystems of full second-order arithmetic (usually denoted by Z_2 or Π^1_∞-CA_0) which contains, in addition to P_0, the scheme of full comprehension: every formula defines a set or, more formally,

(Π^1_∞-CA_0) $\exists X \forall n\ (n \in X \leftrightarrow \varphi(n))$ for every formula φ of second-order arithmetic in which X is not free.

System 2.1 (RCA_0). RCA_0, for recursive comprehension axiom, is a system just strong enough to prove the existence of the recursive sets but not of $\varnothing'$ nor indeed of any nonrecursive set. In addition to P_0 its axioms include the schemes of Δ^0_1 comprehension and Σ^0_1 induction:

(Δ^0_1-CA_0) $\forall n(\varphi(n) \leftrightarrow \psi(n)) \rightarrow \exists X\ \forall n(n \in X \leftrightarrow \varphi(n))$ for all Σ^0_1 formulas φ and Π^0_1 formulas ψ in which X is not free.

($\Sigma^0_1 - I$) $(\varphi(0) \wedge \forall n(\varphi(n) \rightarrow \varphi(n+1)) \rightarrow \forall n \varphi(n))$ for all Σ^0_1 formulas φ.

RCA_0 serves as our base theory. It captures basic arithmetic and simple combinatorial arguments including, for example, the development of primitive recursive arithmetic and the least number operator. It is also strong enough to establish the results of elementary combinatorics and algebra (finite and countable) including, for example, the existence of algebraic closures for countable fields. As one can prove the fundamental theorem of arithmetic in this system, one can code sequences. With such a coding scheme one can develop the basic theory of the reals from both an algebraic and analytic point of view. Thus, for example, one can prove that polynomials and the common transcendental functions like sin, cos, e^x, etc. are continuous and that the intermediate value theorem holds for continuous functions.

Generally speaking, the consequences of RCA_0 correspond to the positive results of recursive algebra and analysis. Similarly, recursive counterexamples usually indicate theorems independent of RCA_0, such as the existence of a non-recursive set, the uniqueness of algebraic closures, and the maximum value theorem.

System 2.2 (WKL_0). WKL_0, for weak König's lemma, consists of RCA_0 plus a version of König's lemma for binary branching trees:

(WKL_0) Every infinite binary branching tree has an infinite path.

WKL_0 is equivalent to the compactness of the Cantor set 2^ω or of the unit interval. It allows one to carry out most applications of compactness or König's lemma in its classical role of deducing countable combinatorial facts from their finite counterparts. One can also do much more in WKL_0 with continuous functions and ideal theory in countable rings than one can in RCA_0. We list a few examples of mathematical theorems which (over RCA_0) are equivalent to WKL_0:

(1) The Heine–Borel theorem.

(2) Every continuous function on $[0, 1]$ is Riemann integrable, uniformly continuous, and has a maximum value.

(3) Every countable commutative ring has a prime ideal.

(4) Every countable field has a unique algebraic closure.

In recursion theoretic terms, WKL_0 corresponds to problems to which the low basis theorem of Jockusch and Soare [9] provides solutions that are strictly recursive in $\varnothing'$, in fact, ones with jump $\varnothing'$. In terms of the types of codings typically used to prove undecidability results, the theorems equivalent to WKL_0 correspond to coding a set separating a pair of recursively inseparable sets. None of these techniques suffice to prove the existence of $\varnothing'$ itself. That is a task beyond the scope of WKL_0 but precisely suited to our next system.

System 2.3 (ACA_0). ACA_0, for arithmetic comprehension axioms, consists of P_0 plus the comprehension scheme for arithmetic formulas:

(ACA_0) $\exists X\, \forall n(n \in X \leftrightarrow \varphi(n))$ for every arithmetic formula φ in which X is not free.

This system clearly includes RCA_0 and is easily seen to imply WKL_0. It is, however, considerably stronger than WKL_0. Indeed, the standard proof of full König's lemma (every infinite, finitely branching subtree of $\mathbb{N}^{\mathbb{N}}$ has an infinite path) works in ACA_0. In fact, ACA_0 is equivalent (over RCA_0) to this strong version of König's lemma. It also allows one to prove various stronger combinatorial results than WKL_0 does. In addition one can develop in ACA_0 reasonable theories of sequential convergence and some maximality results in algebra. We list a few sample theorems equivalent to ACA_0:

(1) Ramsey's theorem for colorings of $\mathbb{N}^{[k]}$ for any $k > 2$.
(2) The Bolzano–Wierstrass theorem or Ascoli's lemma.
(3) Every countable commutative ring has a maximal ideal.
(4) Every countable vector space over a countable field has a basis.
(5) Every countable field has a transcendence basis.

In recursion theoretic terms ACA_0 proves the existence of $\emptyset'$ and by relativization it proves and, in fact, is equivalent to closure under the jump operator. Thus it corresponds to proofs of undecidability which proceed by coding the halting problem. It is not, however, strong enough to continue the iteration of the jump into the transfinite nor even to prove the existence of $\emptyset^{(\omega)}$. For the existence of the H-sets we must turn to the next system.

System 2.4 (ATR_0). ATR_0, for arithmetical transfinite recursion, consists of P_0 plus the assertion that arithmetic comprehension can be iterated along any countable well ordering:

(ATR_0) If X is a set coding a well ordering $<_X$ with domain D and if Y is a code for a set indexed by $x \in D$ of arithmetic formulas $\varphi_x(z, Z)$ with one free set variable and one free number variable, then there is a sequence $\langle K_x \mid x \in D \rangle$ of sets such that if y is the immediate successor of x in $<_X$, then $\forall n(n \in K_y \leftrightarrow \varphi_x(n, K_x))$ and if x is a limit point in $<_X$ then K_x is $\oplus \{K_y \mid y <_X x\}$.

ATR_0 allows one to develop a good theory of countable ordinals, Borel and analytic sets, and to do many transfinite inductions. The recursion theoretic version of the theory has not been disguised at all here. The appropriate equivalent clearly is closure under (relativized) H-set construction:

(ATR_0) $\forall X \forall e(e \in \mathcal{O}^X \rightarrow H_e^X$ exists).

It is worth pointing out that, in both versions of ATR_0, when we refer to a set coding a well ordering or a member of $\mathcal{O}$, we mean it formally (in the sense of the theory or any particular model). Thus, for example, the class HYP of hyperarithmetic sets do not form a model of ATR_0. The problem is that there are orderings that are not well founded but have no hyperarithmetic descending chain. Such orderings would have jump hierarchies attached to them if HYP were truly a model of ATR_0. As such pseudo-hierarchies do not exist in HYP, it is not a model of ATR_0 (Simpson [23, V. Proposition 2.6]).

An axiom looking perhaps more like the ones used to define the other systems that is equivalent to ATR_0 is Σ_1^1 separation:

(Σ_1^1-SEP) $\neg\exists n(\varphi_0(n) \wedge \varphi_1(n)) \rightarrow \exists X\ \forall n(\varphi_0(n) \rightarrow n \in X \wedge \varphi_1(n) \rightarrow n \notin X)$ for all Σ_1^1 formulas φ_0 and φ_1 in which X is not free.

Σ_1^1 separation is the analog for the Σ_1^1 and hyperarithmetic sets of Lusin's theorem that disjoint analytic sets can be separated by a Borel set. Another important set existence principle implied by ATR_0 is a choice principle for Σ_1^1 collections called Σ_1^1-AC.

(Σ_1^1-AC) $\forall n\ \exists X\varphi(n, X) \rightarrow \exists Y\, \forall n\varphi(n, (Y)_n)$ for every Σ_1^1 formula φ, where $(Y)_n = \{x \mid \langle n, x\rangle \in Y\}$.

As for the earlier systems, we list a few mathematical theorems which are known to be equivalent to ATR_0:

(1) Ramsey's theorem for open subsets of $[\omega]^\omega$.

(2) Every closed set of reals either contains a perfect set of is countable.

(3) Any two countable well orderings are comparable.

(4) Ulm's theorem characterizing countable reduced abelian p-groups in terms of invariants.

While ACA_0 gave us the arithmetic sets, ATR_0 allows us to develop the hyperarithmetic sets. On this basis it is easy to see that ATR_0 is strictly stronger than ACA_0 and so it is not provable from compactness or the full König's lemma: $\mathbb{N}$ with all the arithmetic sets is clearly a model of ACA_0 but not of ATR_0. Even ATR_0 does not take us all the way to $\mathcal{O}$. As it does not prove the existence of $\mathcal{O}$, we should note that in the recursion theoretic version of the axioms for ATR_0 that the phrase $e \in \mathcal{O}^X$ is to be understood as an abbreviation for "e is a recursive index relative to X for a well ordering." To prove the existence of $\mathcal{O}$, we must move on to our fifth and last system.

System 2.5 (Π_1^1-CA_0). Π_1^1-CA_0, for Π_1^1 comprehension axiom, is the system P_0 plus comprehension for Π_1^1 formulas:

(Π_1^1-CA_0) $\exists X\ \forall n(n \in X \leftrightarrow \varphi(n))$ for Π_1^1 formula φ in which X is not free.

Again the recursion theoretic equivalent is readily apparent. It is simply the assertion, suitably relativized, that $\mathcal{O}$ exists:

(Π_1^1-CA_0) $\forall X(\mathcal{O}^X$ exists).

Thus Π_1^1-CA_0 is equivalent to closure under hyperjump. Two other important set existence principles implied by Π_1^1-CA_0 are embodied in the

Kleene and Gandy basis theorems which bound the complexity of a witness to a Σ_1^1 formula, assuming only that one exists at all:

(Kleene basis) $\exists X\varphi(X) \rightarrow \exists X(\varphi(X) \wedge X \leqslant_T \mathcal{O})$ for every Σ_1^1 formula φ.

(Gandy basis) $\exists X\varphi(X) \rightarrow \exists X(\varphi(X) \wedge X <_h \mathcal{O})$ for every Σ_1^1 formula φ.

In Π_1^1-CA_0 one can do just a bit more in most respects than in ATR_0. We list a few theorems known to be equivalent to Π_1^1-CA_0:

(1) The Cantor–Bendixson theorem that every closed subset of $\mathcal{R}$ is the union of a countable set and a perfect set.

(2) Every countable abelian group has a maximal divisible subgroup and is the direct sum of a divisible group and a reduced group.

(3) Ramsey's theorem for $F_\sigma \cap G_\delta$ subsets of $[\omega]^\omega$.

In this paper (Section 4) we shall also establish a couple of new equivalents for Π_1^1-CA_0 drawn from combinatorial matching theory. There are very few results of "ordinary mathematics" which cannot be done in Π_1^1-CA_0. One exception is discussed in Smith [24]. Some recent results connecting marriage theorems with weaker systems can be found in Hirst [5, 6].

3. König's Duality Theorem for Countable Graphs

In this section we shall present a proof (based on that of Aharoni [1, 2]) of König's theorem for countable graphs. An extension and analysis of the proof for graphs on $\mathbb{N}$ shows that for such graphs the theorem is provable in Π_1^1-CA_0. We begin by recalling some basic definitions. They also serve to specify some of our notation.

Definition 3.1. (i) A *graph* G consists of a set V of elements called *vertices* and a set E of *edges* where each edge is an (unordered) pair of vertices.

(ii) The graph G is *bipartite* if its set of vertices can be divided into two disjoint sets M and W such that every edge in the graph joins (that is, consists of) a vertex in M and a vertex in W. The sets M and W are called the sides of G.

(iii) A *matching* F in the graph G is a collection of disjoint edges of G. A matching F in G is said to be a *matching from* X *into* Y if, for every vertex $x \in X$, there is a vertex $y \in Y$ such that $(x, y) \in F$. A matching in a bipartite graph G clearly defines a one-one function between a subset of M and a subset of W.

(iv) A *cover* of the graph G is a set C of vertices of G such that every edge in G contains (or is *covered by*) a vertex in C.

(v) A *König cover* C of G is a cover which consists of a selection of one vertex from each edge of a matching in G.

(vi) The *neighbors* of a vertex x in a graph G are those vertices y such that (x, y) is an edge of G. We denote the *set of neighbors* of x by $N(x)$. If G is bipartite and $x \in M$, say, then clearly $N(x) \subseteq W$.

(vii) In a bipartite graph G we define the *demand* $D(X)$ of a set of vertices $X \subseteq W$ by $D(X) = \{m \in M \mid N(m) \subseteq X\}$, the set of vertices in M all of whose neighbors are in X. If it is necessary to indicate the graph G in which we are forming the demand of X, we shall write it as $D_G(X)$. We subscript the other notations similarly when necessary.

We now consider the proof of Theorem 1.1 for countable graphs. Let G be a fixed countable bipartite graphs. We must show that it has a König cover. Note that any vertices in G that are not in any edges of G are irrelevant to this result. Any König cover of the graph G' gotten by omitting all such edges from G is also one for G as any matching F in G' is a matching in G and any cover of G' selected from F is also a cover of G. We can therefore assume that any graph for which we want to find a König cover has no such isolated points. (This assumption is not strictly necessary, but it simplifies the picture at certain points.) We begin our analysis with a consideration of the family $\mathscr{F} = \{X \subseteq W \mid \text{there is a matching in } G \text{ of } X \text{ into } D(X)\}$.

LEMMA 3.2. *$\mathscr{F} = \{X \subseteq W \mid$ there is a matching in G of X into $D(X)\}$ has a maximum element, i.e., one containing all the others.*

Proof. (Axiom of choice). Let $\langle X_\alpha \mid \alpha < \beta \rangle$ be a sequence listing all the elements of $\mathscr{F}$. For each $\alpha < \beta$ there is a matching H_α of X_α into $D(X_\alpha)$. Let F_α be the restriction of this matching to $X_\alpha - \bigcup \{X_\gamma \mid \gamma < \alpha\}$. It is clear that $F = \bigcup \{F_\alpha \mid \alpha < \beta\}$ is a matching of $X = \bigcup \{X_\alpha \mid \alpha < \beta\}$ into $\bigcup \{D(X_\alpha) \mid \alpha < \beta\} \subseteq D(X)$. This matching F is clearly the desired maximum element of $\mathscr{F}$. ▮

From now on X will be the maximum element of $\mathscr{F}$ and F will be a fixed matching of X into $D(X)$. We are now half way to our desired solution. Our desired matching will extend F by adding on a matching H of $M - D(X)$ into $W - X$. The desired cover will then consist of X and the set $Y = M - D(X)$.

LEMMA 3.3. *With the notation as above, $X \cup Y$ is a cover of G.*

Proof. Consider any edge (m, w) in G (with $m \in M$ and $w \in W$). If

$m \in Y$, then the edge is covered at m. If, on the other hand, $m \in D(X)$ then, by definition of D, $w \in X$ and so (m, w) is covered at w by X as required. ▮

It thus suffices to prove that there is a matching of $M - D(X)$ into $W - X$ to complete the proof of König's theorem. We consider the bipartite graph G' with sides $M' = M - D(X)$ and $W' = W - X$ obtained by deleting the vertices in X and $D(X)$ (and so any edges containing them) from G. Note first that for $Z \subseteq W'$ and $m \in D_{G'}(Z)$, $N_G(m) \subseteq Z \cup X$. Thus $D_G(X \cup Z) \supseteq D_{G'}(Z) \cup D(X)$. If there were now a matching H of some non-empty $Z \subseteq W'$ into $D_{G'}(Z)$ in G', we could combine the given matching F (in G of X into $D_G(X)$) with H to get a matching in G of $X \cup Z$ into $D_G(X \cup Z)$. As the existence of such a matching extending F contradicts the maximality of X in $\mathscr{F}$, there is no non-empty $Z \subseteq W'$ such that there is a matching of Z into $D_{G'}(Z)$ in G'. Thus to obtain the matching of $M - D(X)$ into $W - X$ needed to finish the proof of the theorem, it suffices to prove the following lemma.

LEMMA 3.4. *If G' is a bipartite graph in which*

there is no non-empty $X' \subseteq W'$ for which there is a matching of X' strictly into $D(X')$ in G', (*)

then there is a matching in G' of M' into W'.

We build the matching required in Lemma 3.4 inductively, using the following lemma:

LEMMA 3.5. *If a bipartite graph G' has the property* (*) *specified above and $m \in M'$, then there is a $w \in N(m)$ such that $G'' = G' - \{m, w\}$, i.e., G' with vertices m and w and all edges containing either of them removed, also has property* (*).

Proof. We first claim that if G'' does not satisfy (*) for any particular $w \in N(M)$, then there is an $X'_w \subseteq W'$ containing w such that there is a matching in $G' - \{m\}$ of X'_w into $D(X'_w)$. This claim suffices to prove our current lemma 3.5 as we can simply take $X' = \bigcup \{X'_w \mid (m, w) \in G'\}$ and piece together (as in the proof of Lemma 3.2) a matching H of X' into $D(X')$ in $G' - \{m\}$. As $m \in D_{G'}(X')$ by the definition of X' but it is not in the range of H, H is a matching in G' of X' strictly into $D(X')$, contrary to our assumption that G' satisfies (*). All that remains then is to verify the claim.

Proof of the Claim and so of Lemma 3.5. Suppose G'' does not satisfy (*) and so there is a non-empty $X' \subseteq W' - \{w\}$ and a matching in G'' of X' strictly into $D_{G''}(X')$. Let H be such a matching and $c \in D_{G''}(X')$ be outside the range of H. We say that a finite sequence $(m_0, w_0), ..., (m_k, w_k)$ of edges in G'' is an *H-alternating path* if $m_0 = c$, $w_{2i} = w_{2i+1}$, $m_{2i} + 1 = m_{2i+2}$, (m_{2i+1}, w_{2i+1}) is in H for every i and $m_{2i+1} \neq m_{2j+1}$ if $i \neq j$. Let I be the set of all vertices in $M_{G''}$ which are in an edge of an H-alternating path and let Z be the set of all vertices in $W_{G''}$ matched with vertices in I by H. Now by the definition of an H-alternating path, $I \subseteq D_{G''}(Z)$. Thus we have a matching F' in G'' of Z into $I \subseteq D_{G''}(Z)$ which omits c from its range. As G' satisfies (*), F' cannot be such a matching in G' and so there must be an $x \in I$ which has w as a neighbor. We can now define the required matching of X'_w into $D(X'_w)$ in $G' - \{m\}$ by sending w to x and then reversing the alternating path connecting x to c to reassign each member of W' in the path to the element of M' that is next to it in the alternating path. ▮

Proof of Lemma 3.4 *from Lemma* 3.5. It is now clear that we can inductively define the matching in G' of M' into W' required by Lemma 3.4 by simply successively applying Lemma 3.5 to each $m \in M'$ to get a corresponding $w \in W'$ in the graph consisting of the as yet unremoved vertices and edges. This then concludes the proof of König's theorem for countable graphs. ▮

Before continuing on to analyze the complexity of the above proof for countable graphs and the covers it asserts to exist, we want to note that the axiom of choice is necessary for any proof of the full König duality theorem (i.e., for all graphs) as it easily implies this axiom:

PROPOSITION 3.6. *König's duality theorem* 1.1 *implies the axiom of choice.*

Proof. Let A_i be any family of non-empty sets. Let G be the bipartite graph one of whose sides consists of the sets A_i and whose other side consists of all ordered pairs $\langle i, a \rangle$ with $a \in A_i$. The edges of G are simply all pairs $(A_i, \langle i, a \rangle)$ with $a \in A_i$. It is clear that from the matching F guaranteed by Theorem 1.1 one can easily construct a choice function for the family A_i. ▮

Returning now to the case of a countable graph G, we note that the mere existence of a solution for G, i.e., the required matching F and König cover C, gives us a bound on the complexity of at least one such solution. We assume for convenience that the vertices of G are contained in $\mathbb{N}$. The property of being a solution is clearly arithmetic in G. Thus the standard basis theorems described in Section 2 apply to tell us that there is a solution recursive in the hyperjump of G and indeed ones of strictly smaller hyperdegree than that of G. Of course, if G is recursive, we simply obtain the existence of solutions recursive in, or of strictly smaller hyperdegree than, Kleene's $\mathcal{O}$. We shall see in the next section that this bound is essentially the best possible on König covers as there are recursive graphs for which any König cover must be (Turing) above all the hyperarithmetic sets.

We now wish to analyze the proof theoretic complexity of the König duality theorem. Our first step is to refine the above proof by applying the basis theorems, again under the assumption that the vertices of G are contained in $\mathbb{N}$, to see that the theorem is provable in Π^1_1-CA_0. It is clear that the proofs of Lemmas 3.3 and of the claim within Lemma 3.5 are quite elementary. Problems first arise with the use of some form of the axiom of choice in both the proof of the existence of the maximum member X of $\mathscr{F}$ in Lemma 3.2 and in the argument for Lemma 3.5 from the claim within it that also pieces together a matching. The second troublesome spot is in the induction needed to deduce Lemma 3.4 from Lemma 3.5. We first consider Lemma 3.2. Rather than explicitly relativizing everything to G, we simply assume that G is a recursive graph on $\mathbb{N}$ and prove the existence of solution in the lightface, i.e., unrelativized, case. (As usual, the proofs for an arbitrary G are simply gotten by replacing "recursive" by "recursive in G" and in a similar vein adding G on as an additional primitive predicate in our language of arithmetic.)

Proof of Lemma 3.2 *in* Π^1_1-CA_0. We begin with the observation that $\mathscr{F}$ is obviously a Σ^1_1 class of sets as is, for each $w \in W$, the class $\mathscr{F}_w = \{X \in \mathscr{F} \mid w \in X\}$. The set $T = \{w \mid \mathscr{F}_w \neq \varnothing\}$ is then Σ^1_1 and so exists by Π^1_1-CA_0. The collection $\mathscr{P}_w$ of sets and matchings given by $\mathscr{P}_w = \{\langle X, H\rangle \mid X \in \mathscr{F}_w$ and H is a matching of X into $D(X)\}$ is an arithmetic class which is non-empty for each $w \in T$. Applying now either the Kleene basis theorem (which is a consequence of Π^1_1-CA_0) or Σ^1_1-AC (which follows from the basis theorem) relative to T, we obtain the existence of a sequence $\langle\langle X_w, H_w\rangle \mid w \in T\rangle$ such that, for each $w \in T$, $w \in X_w$ and H_w is a matching of X into $D(X)$. Given this sequence we can now directly define a matching F from $X = \bigcup \{X_w \mid w \in T\}$ into $\bigcup \{D(X_w) \mid w \in T\} \subseteq D(X)$ to consist of the edges (w, m) such that $(w, m) \in H_w$. ∎

The argument deducing Lemma 3.5 from the claim within it is a similar exercise in using Σ^1_1-AC to piece together the required matching of X into $D(X)$ in $G - \{m\}$.

Finally, we consider the inductive proof of Lemma 3.4 from Lemma 3.5. The crucial point here is that the property (*) that we need to propagate is itself Π^1_1 and that the instances of (*) that we need can be listed in advance. More precisely, for each finite set D of vertices in G we let G_D be the graph obtained by deleting from G all vertices in D and all edges with vertices in D. The set $\mathscr{D} = \{D\colon G_D$ satisfies $(*)\}$ is Π^1_1 and so exists by Π^1_1-CA_0. We now begin with the bipartite subgraph G_0 of G given by restricting G_0 to the sides $W_0 = W - X$ and $M_0 = M - D(X)$. Let $\langle m_i \rangle$ be a sequence listing all the elements of M_0. Using $\mathscr{D}$ we may now, by induction, successively choose $w_i \in W_0$ and subgraphs G_i such that (m_i, w_i) is an edge of G_i and such that $G_{i+1} = G_i - \{m_i, w_i\}$ still has property (*). ∎

It is worth noting at this point that, if we are concerned only with finite graphs, the entire proof is elementary. All the orderings, piecing together of matchings, and inductions become purely finitary operations and so doable in weak systems. For a more direct finitistic proof of the theorem for finite graphs see for example Lovasz and Plummer [13]. In the other direction, the proof for uncountable graphs is considerably more complicated than the one given here. For those results see Aharoni [2] or Holz, Podewski, and Steffens [7].

Before proceeding to the proof that strong theories are necessary to prove the duality theorem, we should remark that we have actually proven (in Π^1_1-CA_0) a somewhat stronger version of König's theorem than that given in Theorem 1.1. The extra information is supplied in Lemma 3.3, where we precisely specify the elements of W that are in the cover we construct. While the version given in Theorem 1.1 proves ATR_0, this extended version turns out to prove the existence of $\mathcal{O}$ and therefore to be equivalent to Π^1_1-CA_0 (Theorem 4.18).

THEOREM 3.7 (Extended König duality theorem). *In any countable bipartite graph G with sides M and W, there is a matching F and a selection C of one vertex from each edge in F which is cover of G such that for any $w \in W$, $w \in C$ if and only if there is an $X \subseteq W$ containing w and a matching in G of X into $D(X)$.*

In fact, we shall see (Theorem 4.20) that Lemma 3.2 itself guarantees the existence of $\mathcal{O}$ and so it is equivelent to Π^1_1-CA_0. The situation is similar for various other results asserting the existence of maximal of different sorts (Theorem 4.22).

4. SIMPLE GRAPHS WITH ONLY COMPLEX KÖNIG COVERS

Our first task in this section is to determine the complexity of König covers for recursive graphs. We show that for every H-set H_e there is (and indeed we can effectively find) a recursive graph G_e such that H_e is recursive in any König cover of G_e. The uniformity of the construction will then allow us to combine these graphs into a single recursive graph G such that for every $e \in \mathcal{O}$, H_e is recursive in every König cover of G. On the other hand, general considerations show that there is no recursive graph such that all of its König covers compute $\mathcal{O}$. There are, however, recursive graphs such that any cover of G that satisfies the extra properties specified in what we have called the extended König duality theorem (3.7) does, in fact, compute $\mathcal{O}$ We also make some brief comments on the complexity of covers for graphs on the reals: there are recursive (and so clopen) graphs on $\mathcal{R}$ with no projective covers. Finally, we apply (the relativizations of) these results to show (in RCA_0) that König's duality theorem implies ATR_0 and that the extended version is equivalent to Π^1_1-CA_0. We also show that some related results in matching theory used in various proofs of the basic duality theorem also imply Π^1_1-CA_0.

Essentially all the graphs we actually consider will be recursive rooted ω-branching trees of height at most ω.

DEFINITION 4.1. A *rooted tree* is a set T (whose elements are called *nodes*) partially ordered by $<_T$ with a unique least element called its *root* in which the predecessors of any element are well ordered by $<_T$. A tree T is *recusive* if it nodes from a recursive set of natural numbers and the ordering $<_T$ is a recursive relation.

DEFINITION 4.2. Let T be rooted tree.

(i) The *levels* of T are defined by induction. The 0th level of T consists precisely of the root of T. The $(k+1)$th level of T consist of the immediate successors (in $<_T$) of the nodes on the kth level of T.

(ii) We say the *height of T is at most ω* if every node is at level n for some $n \in \omega$.

(iii) If each node of T has at most ω immediate successors, T is *ω-branching.*

From now on a tree will be a rooted ω-branching recursive tree of height at most ω all of whose nodes are natural numbers. Such a tree T can be viewed as a bipartite graph by letting W consist of the nodes at the even levels of T, M the ones at odd levels, and E the set of pairs (x, y) such that x is an immediate successor of y in T. The resulting graph G is also *recursive* (i.e., M, W, and E are all recursive sets) if we adopt (as we now do) the convention that each node on one of our trees is a larger number than any of the nodes below it in the tree ordering. (In this case we can recursively determine if x is an immediate successor of y in T and the level of any node of T. These are the only facts needed to determine M, W, and E recursively.) We abuse notation slightly by identifying trees with the corresponding bipartite graphs. We also adopt the convention that all sequences $\langle T_n \rangle$ of trees that we consider are uniformly recursive. (Recall that the uniformity of the sequence just means that there is a recursive function f such that, for each n, $f(n)$ is a recursive index for T_n, i.e., for the characteristic functions of the corresponding set and ordering. This requirement is equivalent to the predicates "$i \in T_n$" and "$i_n <_{T_n} j$" being recursive predicates of i, j, and n.)

Our notion of coding by trees is given by the following definitions:

DEFINITION 4.3. A tree T with root node r *codes a fact* φ if, for every König cover C of T, $r \in C \Leftrightarrow \varphi$. A sequence $\langle T_n \rangle$ of trees with root nodes r_n *codes a predicate* $P(x_1, \ldots, x_m)$ if, for every sequence of natural numbers $a_1, \ldots, a_m$, $T_{\langle a_1, \ldots, a_m \rangle}$ codes $P(a_1, \ldots, a_m)$. We say the sequence $\langle T_n \rangle$ *codes a set* A if it codes the predicate "$a \in A$."

The next lemma shows that it is sufficient to code the H-sets by sequences of trees.

LEMMA 4.4. *Given any sequence $\langle T_n \rangle$ of trees coding a set A, we can uniformly construct a single tree T such that A is uniformly recursive in any König cover of T. (The uniformity claimed here means that, given an index for the sequence of indices of the T_n, we can recursively find an index for T and for a reduction procedure, that is, a Turing machine, M such that, when equipped with an oracle for any König cover of T, M computes A.)*

In this lemma and in all our other constructions the uniformities asserted to exist in the statements of the results will be obvious from the constructions. All the manipulations of graphs that we employ will clearly be uniform in the sense that given an index for the inputs we can effectively compute indices for the outputs. (For those unfamiliar with such arguments, just think of the inputs as programs to compute the given graphs and the output as programs to compute the graphs asserted to exist. The manipulations needed to convert the former into the latter will all be simple programming procedures.) Thus we shall never explicitly mention these uniformities in the proofs. We begin by isolating from the proof of Lemma 4.4, a basic fact that will be used several more times.

LEMMA 4.5. *Suppose a tree T appears as a subtree of a tree S in such away that there is only one edge e in S which contains a vertex of T but is not in T. Moreover, assume that this edge e connects the root r of T to the root s of S. (In terms of the tree picture this means that r is on level one of S and T is the subtree of S consisting of r and all nodes above r.) If F is a matching in S and C a corresponding König cover of S selected from F such that either $e \notin F$ or $r \notin C$, then $F \upharpoonright T$, F restricted to T, is a matching in T and, C restricted to T is a König cover of T selected from $F \upharpoonright T$.*

Proof. The restriction of F to T is obviously a matching in T. To see that $C \upharpoonright T$ is a cover of G consider any edge (x, y) in T. It is also an edge of S and so one of its vertices, say x, is in C. If $r \notin C$, then of course $r \neq x$. In this case, x must be selected by C from an edge in F with both vertices in T. Thus it is selected from $F \upharpoonright T$ as well. If $r = x$ but $e \notin F$, the crucial point to notice is that, since e is the only edge of S containing a vertex in T which is not itself an edge of T, the vertex $r(=x)$ chosen by C from F must again have been chosen from an edge with both vertices in T (and so in $F \upharpoonright T$). ∎

Proof of Lemma 4.4. For each n, form three isomorphic copies $T_{n,j}$ ($j = 1, 2, 3$) of T_n. Without loss of generality, these trees can be constructed such that they are pairwise disjoint and there is some number r which is not a node in any of them. We form a new tree T with root r by attaching to r all the $T_{n,j}$, i.e., their root nodes, $r_{n,j}$, are the immediate successors of r in T after which T looks just like the $T_{n,j}$. Let C be any König cover of T with associated matching F. At most one of the edges connecting r to one of the $r_{n,j}$ can be in F. Thus, with at most one exception, the restrictions of C to the $T_{n,j}$ give König covers by Lemma 4.5. As each $T_{n,j}$ codes $n \in A$, we know that $n \in A$ iff at least two of the $r_{n,j}$ ($j = 1, 2, 3$) are in C. ∎

Our goal is thus to show that, for every $e \in \mathcal{O}$, there is a recusive sequence $\langle T_n \rangle$ of trees which codes H_e. In view of the way that the H sets are defined and the remarks on effective transfinite induction in Section 2, it suffices to prove the following three lemmas:

LEMMA 4.6. *There is a uniform procedure for coding any recursive set R by a recursive sequence of trees.*

LEMMA 4.7. *Given a recursive sequence $\langle T_n \rangle$ of trees coding A, we can uniformly find a sequence $\langle S_n \rangle$ of trees coding A', the Turing jump of A.*

LEMMA 4.8. *Given simultaneously recursive sequences $\langle T_{n,i} \rangle$ such that, for each n, the sequence $\langle T_{n,i} \mid i \in \omega \rangle$ codes the set A_n, we can uniformly produce a sequence $\langle T_k \rangle$ of trees that codes $A = \bigoplus A_n = \{\langle n, i \rangle \mid i \in A_n\}$, the effective union of the A_n.*

The last of these three lemmas, 4.8, is obvious: simply take the sequence $\langle T_{\langle n, i \rangle} \rangle$. The first of the three, 4.6, is almost as easy. The second, 4.7, requires two somewhat more complicated constructions.

Proof of Lemma 4.6. If $n \in R$, let T_n consist of root node $\langle n \rangle$ with immediate successors $\langle n, 0 \rangle$ and $\langle n, 1 \rangle$. If $n \notin R$, then in addition let $\langle n, i, j \rangle$ for $j = 0, 1$ be the immediate successors in T_n of $\langle n, i \rangle$ for $i = 0, 1$. It is clear that if $n \in R$, the only König cover of T_n is $\{\langle n \rangle\}$. (There are only two edges in the graph. As both of them contain $\langle n \rangle$, only one can be in the required matching. The only way to choose a cover then is to pick $\langle n \rangle$.) If, on the other hand, $n \notin R$, the only König cover of T_n is $\{\langle n, 0\rangle, \langle n, 1\rangle\}$. (The argument is similar to the one in the previous case. Not both $\langle n, i, 0\rangle$ and $\langle n, i, 1\rangle$ can be in the cover (for fixed $i=0$ or 1) by the disjointness requirement on the edges in the matching. Thus for both $n=0$ and 1, $\langle n, i\rangle$ must be in the cover by the definition of a cover. As each of the edges containing $\langle n \rangle$ contains one of the $\langle n, i\rangle$, $\langle n \rangle$ cannot be in the cover as the edges in the matching are disjoint and we can choose only one vertex from each edge.) ▌

As explained in Section 2, to get closure under the jump operator it suffices to prove closure under complementation and projection. Thus to establish Lemma 4.7 it suffices to prove that the set of predicates coded by recursive sequences of trees is closed under these operations.

LEMMA 4.9. *The predicates coded by recursive sequences of trees are closed under complementation.*

LEMMA 4.10. *The predicates coded by sequences of trees are closed under projections (or as one might say, recursive disjunctions), i.e., if there is a sequence coding the predicate $P(i, n)$, then there is one coding the predicate $P(i) \equiv \exists n P(i, n)$.*

Proof of Lemma 4.9. It clearly suffices to show that, if T codes a fact φ, we can uniformly produce a $\bar{T}$ coding $\neg\varphi$. We define $\bar{T}$ as the tree obtained by attaching two (disjoint) copies T_1 and T_2 of T to a new node which is then the root of $\bar{T}$. (In terms of the graph, we take two disjoint copies of T add on one new vertex, r, and two edges, e_1 and e_2, joining r to each copy (r_1 and r_2) of the original root node of T.) Consider now any matching F and corresponding König cover C of $\bar{T}$. C must contain one vertex from each of the edges e_1 and e_2. If it contains r, the matching F from which r is selected must contain exactly one of e_1 and e_2. Suppose for the sake of definiteness that it contains e_1. In this case F and C restricted to T_2 also constitute a maching and its corresponding König cover by Lemma 4.5. As C contains r and F contains e_1, this cover cannot contain r_1. Thus we have a König cover of T not containing its root node, i.e., $\neg\varphi$ holds as required. On the other hand, if $r \notin C$, then both r_1 and r_2 must belong to C but, again, at most one of e_1 and e_2 belong to F. Suppose e_2 does not belong to F. In this case, again by Lemma 4.5, the restriction of C and F to T_2 produces a matching and corresponding König cover. We then have a König cover of T containing its root node and so φ holds as required. ▌

Proof of Lemma 4.10. We are given a sequence $\langle T_{\langle i,n\rangle}\rangle$ of trees coding the predicate $P(i, n)$ and wish to define a sequence $\langle S_i \rangle$ of trees coding the predicate $\exists n P(i, n)$. For notational convenience we fix i and drop it from our subscripts. Our new tree S consists of a root node r to which we have attached, for each n, two copies $\bar{T}_{n,1}$ and $\bar{T}_{n,2}$ of $\bar{T}_n$ (the tree coding the complement of what T_n codes as constructed in the proof of Lemma 4.9). Consider now any matching F in S and corresponding König cover C of S. If $r \in C$, exactly one of the root nodes $r_{n,j}$ of these copies of $\bar{T}_n$, is such that $(r_{n,j}, r) \in F$. Suppose it is $r_{n,1}$, the root of $\bar{T}_{n,1}$; $r_{n,1}$ cannot be in C but, by Lemma 4.5, C restricted to $\bar{T}_{n,1}$ constitutes a König cover selected from the matching F restricted to the same subtree. Thus, by our choice of $\bar{T}_{n,1}$, $P(i, n)$ holds as required. On the other hand, if $r \notin C$, then every node $r_{m,j}$ must be in C, but at most one edge containing r, say $(r_{p,2}, r)$, can be in F. Thus, for every m, the restrictions of F and C to $\bar{T}_{n,1}$ give a matching and König cover containing $r_{m,1}$. Our chose of $\bar{T}_{m,1}$ then guarantees that $P(i, m)$ fails for every m as required. ∎

We have thus proven that König covers for recursive graphs necessarily have computational complexities that exhaust the hyperarithmetic sets:

THEOREM 4.11. *For every $e \in \mathcal{O}$ there is a recursive graph G_e such that H_e is recursive in any König cover of G_e.*

Indeed the uniformities necessarily present in our constructions, combined with an application of the recursion theorem in the form of effective transfinite induction, show a bit more.

THEOREM 4.12. *There is a recursive graph G such that every hyperarithmetic set is recursive in every König cover of G.*

Proof. The uniformities present in all of the lemmas of this section show that for each e we can uniformly construct a sequence $\langle T_{e,n} \rangle$ of trees such that, if $e \in \mathcal{O}$, the sequence codes H_e. Lemma 4.4 then produces a single graph $T_{g(e)}$ such that H_e is recursive in any König cover of $T_{g(e)}$. On the other hand, if e is any number, not necessarily in $\mathcal{O}$, the procedure inherent in these uniformities produces some index $g(e)$. Even if this is an index for a partial function we can uniformly interpret it as coding a bipartite (though possible finite or even empty) graph $T_{g(e)}$ in some systematic way. Thus, if we now take a disjoint union over all e, of the graphs $T_{g(e)}$ for each e, we obtain one large recursive bipartite graph G. As any König cover of G restricts to one of each of the $T_{g(e)}$, any König cover of G is Turing above every H-set. ∎

The next obvious question is whether one can find a recursive graph G such that $\mathcal{O}$ is recursive in any König cover of G. The answer to this question is no, on quite general grounds. The Kleene basis theorem says that one can always find a König cover for a recursive graph G which is recursive in $\mathcal{O}$; but the Gandy basis theorem says that one can find a cover of strictly smaller hyperdegree and so, in particular, not above $\mathcal{O}$ in Turing degree. (These basis theorems are described in System 2.5.) The known methods of directly constructing König covers, however, all require going all the way to $\mathcal{O}$, even for recursive graphs. To be precise we recall that, for example, the proof we presented in Section 3 established a stronger result than König's theorem by constructing covers with a certain maximality property expressed in Theorem 3.7, the extended König duality theorem. We now prove that such constructions necessarily capture all of $\mathcal{O}$.

THEOREM 4.13. *There is a recursive graph G such that $\mathcal{O}$ is recursive in any König cover for G that satisfies the conditions of Theorem* 3.7.

Proof. To calculate $\mathcal{O}$ or any Π_1^1 set it suffices to be able to decide if a given recursive tree (or even linear ordering) has an infinite descending path. Given a recursive tree T with root r we can uniformly construct a recursive tree T' with root r in which we have inserted a single node x' immediately after each node $x \neq r$ of T and immediately before its immediate successors in T (if any). Thus in T' every node $x \neq r$ of T has exactly one immediate successor x' and the immediate successors of x' in T' are just the immediate successors of x in T. Let G' be the bipartite graph corresponding to T' and let W' be the side of this graph containing r.

CLAIM. *There is an infinite path in T', and hence in T, if and only if there is an $X \subseteq W'$ containing r and a matching F in G' of X into $D(X)$.*

Proof of Claim. Suppose first that there are such X and F. We build the desired infinite path in T' by induction two levels at a time. We begin with $r \in X$. As F is a matching of X into $D(X)$, there is an immediate successor x_1 of r (in both T and T') in $D(X)$ to which r must be matched by F. By our constructing of T', x_1 has exactly one immediate successor x_1' in T'. The definition of $D(X)$ then guarantees that $x_1' \in X$. Now x_1' cannot be matched with x_1 by the disjointness of the matching F. Thus it must be matched with one of its immediate successors x_2 in T' (which must be one of the immediate successors of x_1 in T); $x_2 \in D(X)$ and so its immediate successor x_2' in T' is in X. The construction now continues in this way by induction.

Finally, suppose that there is an infinite path in T'. It must necessarily be of the form $r, x_1, x_1', x_2, x_2', \ldots$ with $r, x_1, x_2, \ldots$ forming an infinite path in T. Let $X = \{r, x_1', x_2', \ldots\}$. The set $\{x_1, x_2, \ldots\}$ is clearly contained in $D(X)$ and so the matching $\{(r, x_1), (x_1', x_2), (x_2', x_3), \ldots\}$ is the required matching of X into $D(X)$.

Returning now to the proof of Theorem 4.13, we see that T has an infinite path iff r is a member of every König cover of G' satisfying the conditions of Theorem 3.7. We can now form, for every partial recursive tree T_e with root node r_e, the corresponding graph (tree) T'_e. If we let T be a recursive disjoint union of the T'_e, then $\mathcal{O}$ is recursive in any König cover C of T satisfying the conditions of Theorem 3.7: T_e has an infinite path iff $r_e \in C$. ∎

If we turn now briefly to graphs on the reals $\mathcal{R}$ and consider trees with nodes in $\mathcal{R}$, branchings corresponding to the elements of $\mathcal{R}$ and sequences of such trees indexed by $\mathcal{R}$, all the arguments of this section (except Lemma 4.7, since we have not defined the jump of a set of reals) carry over straightforwardly. The crucial point is that in Lemma 4.10 we in fact prove that the coded predicates are closed under quantification over $\mathcal{R}$ as well as under ordinary disjunctions. We thus have proved that the sets of reals coded by recursive graphs on $\mathcal{R}$ contain far more than the projective sets:

THEOREM 4.14. *The sets of reals coded by recursive graphs on $\mathcal{R}$ are closed under complementation, recursive disjunction and quantification over $\mathcal{R}$.*

COROLLARY 4.15. *There are recursive graphs on $\mathcal{R}$ with no projective König covers.*

To conclude we shall apply our complexity theoretic results to measure the proof theoretic strength of König's duality theorem.

THEOREM 4.16. *König's duality theorem for graphs on $\mathbb{N}$ implies (over RCA_0) ATR_0 and so, in particular, it is not provable in ACA_0 nor equivalently in RCA_0 plus compactness or the full König's lemma.*

Proof. ATR_0 is just the existence of all H^X-sets for each set X. This is basically just the relativization of Theorem 4.12. To be precise we first note that the relativization of Lemma 4.7 and the closure of sets in RCA_0 under "recursive in" implies that the Turing jump is always defined and so we have ACA_0. For any given X we can form the graph G^X recursive in X as in a relativized version of Theorem 4.13. We can now for any $e \in \mathcal{O}^X$ argue by an induction on $<_e$ which is arithmetic in G^X that H_e^X is recursive in G^X. Thus H_e^X exists for every $e \in \mathcal{O}^X$ and we have proven ATR_0 from König's duality theorem. By the remarks at the end of System 2.4, the duality theorem is then strictly stronger than all of the other systems mentioned above. ∎

We now are faced with a gap in the proof theoretic strength of the theorem similar to the one between the hyperarithmetic sets and $\mathcal{O}$. We have deduced ATR_0 from the König duality theorem and have proved the theorem in Π^1_1-CA_0. The obvious question is, can we close the gap by proving Π^1_1-CA_0 from the theorem or by establishing it in ATR_0. As the theorem is clearly a true Π^1_2 sentence, the former is not possible on general grounds:

PROPOSITION 4.17 (Folklore). *Π^1_1-CA_0 is not provable (even in ATR_0) from any true Π^1_2 sentence P.*

Proof. By definition every true Π^1_2 sentence P is true in every β-model of ATR_0. (A β-model M of second-order arithmetic is one in which the natural numbers are standard and which is absolute for all Σ^1_1 sentences φ with parameters in M, i.e., φ is true iff $M \models \varphi$.) As there are β-models of ATR_0 which are not ones of Π^1_1-CA_0, Π^1_1-CA_0 cannot be a consequence of P in ATR_0. Indeed the minimum β-model of Π^1_1-CA_0 consists of the sets recursive in $\mathcal{O}^{(n)}$, the nth hyperjump of $\emptyset$, for some $n \in \omega$. On the other hand, the intersection of all the β-models of ATR_0 consists of just the hyperarithmetic sets. Thus there is a β-model of ATR_0 which is not one of Π^1_1-CA_0. (Each of these facts can be found, for instance, in Simpson [20 or 23, Chaps. VII, VIII].) ∎

On the other hand, the consequence of the known proofs of the König duality theorem for countable graphs that we have called the extended König duality theorem (3.7) does prove Π^1_1-CA_0.

As $\mathcal{O}$ is a complete Π^1_1 set, the relativization of Theorem 4.13 clearly gives Π^1_1-CA_0. Combining this with the provability of Theorem 3.7 in Π^1_1-CA_0 established in Section 3 gives a precise calibration of its strength in the sense of reverse mathematics.

THEOREM 4.18. *The extended König duality theorem* (3.7) *is equivalent to Π^1_1-CA_0.*

The proof of Theorem 4.13 actually shows that Lemma 3.2 (which asserts the existence, in any bipartite graph G, of a maximum $X \subseteq W$ with a matching into $D(X)$) implies the existence of $\mathcal{O}$ and so is itself equivalent to Π^1_1-CA_0.

THEOREM 4.19. *There is a recursive graph G such that $\mathcal{O}$ is recursive in the maximum element X of $\mathcal{F} = \{X \subseteq W \mid$ there is a matching in G of X into $D(X)\}$.*

Proof. Consider the G constructed in the proof of Theorem 4.13. That proof shows that T_e has an infinite path if and only if r_e belongs to the maximum X in $\mathcal{F}$. ∎

COROLLARY 4.20. *Lemma* 3.2 *is equivalent to* Π^1_1-CA_0.

Various results similar to Lemma 3.2 occur in all the known proofs of König's duality theorem and so we see that a proof of it in ATR_0 (if one exists) requires some entirely new approach. Some of these results are of independent interest in matching theory and the investigation of conditions for the existence of injective choice functions for given families. They often arise in attempts to extend Hall's theorem characterizing the finite families with injective choice functions to infinite families. These results are intimately connected with König's theorem, Menger's theorem, and other important results in finite and infinite combinatorics. Indeed, the solution by Podewski and Steffens [15] to König's theorem in the countable case came as a corollary to such a result for countable families via an application of Brualdi [3]. We shall describe just one result on choice functions from their paper which they combine with Brualdi's work to give a proof of König's theorem. With some effort we can see that their construction also gives a cover with enough of the maximality property of the extended König duality theorem (3.7) to carry out our proof of Π^1_1-CA_0. As the construction in Brualdi [3] is elementary, the result of Podewski and Steffens [15] also implies Π^1_1-CA_0.

DEFINITION 4.21. (i) A *family* F is a function from an index set I to a collection of non-empty sets $\{F(i) \mid i \in I\}$.

(ii) A *subfamily* F' of a family F is the restriction of F to some subset I' of I.

(iii) A *choice function* f for the family F is a function on I such that $f(i) \in F(i)$ for every $i \in I$.

(iv) A subfamily F' of a family F is *maximal representable* if there is an injective choice function for F' but not for any subfamily properly containing it.

The crucial result of Podewski and Steffens [15] which they combine with Brualdi [3] to prove König's theorem is the following:

LEMMA 4.22. *Every countable family has a maximal representable subfamily.*

This lemma easily gives the fact that Brualdi lacked:

LEMMA 4.23. *In every countable bipartite graph* G *there is a maximal* $P \subseteq W$ *such that there is a matching* F *from* P *into* M (*or in the terminology of the cited papers, such that there is a matching* F *which meets* W *in* P).

Proof. This is just a matter of translating the various definitions. We can just take P to be a maximal representable subfamily of the family F defined on W by $F(w) = N(w)$. ▌

Now Brualdi [3] proves by the well-known alternating paths proof of the finite version of König's that, if P and F are as in Lemma 4.23, then there is a covering of G consisting of one vertex from each edge in F, i.e., a König cover of G. A careful analysis of this construction shows that, in fact, there is one containing every $Y \subseteq W$ with a matching into $D(Y)$. Thus if G is a graph constructed from a recursive tree T as in our proof of Theorem 4.13 and T has an infinite path, then its root is in the cover constructed by Brualdi. On the other hand, we can also show that if T has no infinite path, then its root is not in this cover. Thus we can again prove the existence of $\mathcal{O}$. The only non-elementary part of the proof is Lemma 4.22 which therefore itself proves Π^1_1-CA_0. We shall now fill in the details of these claims.

Let P and F be as in Lemma 4.23 and let Q be $F[P]$, the image of P under F. Brualdi defines partitions of P and Q as follows:

$$P_1 = \{x \in P \mid (x, y) \in E \text{ for some } y \notin Q\}$$

$$Q_2 = \{y \in Q \mid (x, y) \in E \text{ for some } x \notin P\}$$

$$Q_1 = F[P_1], \quad P_2 = F[Q_2], \quad P_3 = P - (P_1 \cup P_2), \quad Q_3 = Q - (Q_1 \cup Q_2).$$

He then continues to define subsets of P_3 and Q_3:

$$P_3^1 = \{x \in P_3 \mid (x, y) \in E \text{ for some } y \in Q_1\}$$

$$Q_3^1 = \{y \in Q_3 \mid (x, y) \in E \text{ for some } x \in P_2\}$$

and for $i \geqslant 1$, he sets

$$P_3^{i+1} = \left\{x \in P_3 - \bigcup \{P_3^j \mid j \leqslant i\} \mid (x, y) \in E \text{ for some } y \in F[P_3^i]\right\}$$

$$Q_3^{i+1} = \left\{x \in Q_3 - \bigcup \{Q_3^j \mid j \leqslant i\} \mid (x, y) \in E \text{ for some } y \in F[Q_3^i]\right\}.$$

These sequences may or may not be infinite. In any case Brualdi defines $C_1 = P_1 \cup \bigcup \{P_3^i \mid i \geqslant 1\}$, $C_2 = Q_2 \cup \bigcup \{Q_3^i \mid i \geqslant 1\}$ and $C_3 = P_3 - \bigcup \{P_3^i \cup F[Q_3^i] \mid i \geqslant 1\}$. His result, in these terms, is that $C = C_1 \cup C_2 \cup C_3$ is a cover of G. For our purposes it suffices to prove two facts about this cover:

LEMMA 4.24. *For the graph G defined as in Theorem 4.13 from a tree T with root r and no infinite branches, if $r \in P$, then $r \notin C_1 \cup C_3$, i.e., $F(r) \in C_2$.*

Proof. We keep all the notational conventions from both Theorem 4.13 and the description of Brualdi's result above. Suppose that $r \in P$. If $x_1 = F(r) \in Q_2$, we are done. If $x_1 = F(r) \notin Q_2$, then, by definition of Q_2 and G, $x_1' \in P$. If $x_2 = F(x_1') \in Q_2$, then $x_1 \in Q_3^1$ as required. If not, $x_2' = F(x_2) \in P$. If $x_3 = F(x_3') \in Q_2$, then $x_1 \in Q_3^2$ as required. If not, $x_3' = F(x_3) \in P$. As T (and so T') has no infinite branches, we must eventually reach a stage at which $x_{n+1} = F(x_n') \in Q_2$ and so we obtain $x_1 \in Q_3^n$ as required.

LEMMA 4.25. *For any countable bipartite graph G and any $Y \subseteq W$ with a matching into $D(Y)$, $Y \subseteq C_1 \cup C_3$, i.e., $Y \subseteq P$ and $F[Y] \cap C_2 = \varnothing$.*

Proof. We first fix a matching H of Y into $D(Y)$. We next show that $Y \subseteq P$. If not let $y_0 \in Y - P$. Consider $z_1 = H(y_0) \in D(X)$. If it is not in Q, we immediately have a contradiction to the maximality of P. (We could simply add $(y_0, H(y_0))$ to the matching and also y_0 to P.) Define now the sequences $y_n = F(z_n) \in Y$ (as $z_n \in D(Y)$) and $z_{n+1} = H(y_n) \in D(Y)$ (by our choice of H). If these sequences terminate, they do so because some z_{n+1} is not in Q. In this case we may show that P is not maximal by replacing the edges $(y_1, z_1), \ldots, (y_n, z_n)$ in F by $(y_0, z_1), \ldots, (y_n, z_{n+1})$. On the other hand, if the sequences do not terminate, we can contradict the maximality of P by replacing all the edges (y_n, z_n) for $n \geq 1$ by (y_n, z_{n+1}) for $n \geq 0$.

For our final argument by contradiction, suppose that $y_0 \in Y$ (and so it is in P as well) but $z_0 = F(y_0) \in Q_3^m$ for the least possible m. (We set $Q_2 = Q_3^0$ so as to be able to treat all the cases simultaneously.) By definition of Q_3^m, there are sequences $z_{-i} \in Q_3^{m-i}$ and $y_{-i} \in P$ such that $(y_{1-i}, z_{-i}) \in E$ and $F(y_{-i}) = z_{-i}$ for $0 \leqslant i \leqslant m$. In particular, as $z_{-m} \in Q_3^0 = Q_2$, there is a $y_{-m-1} \notin P$ such that $(y_{-m-1}, z_{-m}) \in E$. As we chose m to be least, none of the y_{-i} $(i > 0)$ are in Y and so none of the z_{-i} are in $D(Y)$. Now let y_j and z_j be defined for $j > 0$ as in the proof that $y_0 \in P$. Once again we can contradict the maximality of P by replacing the edges $(y_{-m}, z_{-m}), \ldots, (y_0, z_0), (y_1, z_1), \ldots [, (y_n, z_n)$ if the above sequence is finite$]$ by $(y_{-m-1}, z_{-m}), \ldots, (y_0, z_1), (y_1, z_2), \ldots [, (y_n, z_{n+1})]$. ∎

We now see that the existence of maximal representable subfamilies is another example of a combinatorial principle that implies Π_1^1-CA_0.

THEOREM 4.26. *Lemma 4.22 proves Π_1^1-CA_0.*

Proof. As in the proof of Theorem 4.13, we let G be the disjoint union of the graphs G_e corresponding to the partial recursive trees T_e. Let C be the cover of G given by Brualdi's construction starting with any maximal P as guaranteed by Lemma 4.23. If T_e has an infinite path, then, by the argument in the proof of Theorem 4.13, there is a Y containing r_e and a matching of Y into $D(Y)$. So by Lemma 4.25, $r_e \in C$. On the other hand, if T has no infinite branches, then, by Lemma 4.24, $r_e \notin C$. Thus T_e has an infinite path if and only if $r_e \in C$. As before this means that $\mathcal{O}$ is recursive in C and so Lemma 4.22 proves Π^1_1-CA_0. ∎

References

1. R. Aharoni, On a duality principle in infinite bipartite graphs, *J. London Math. Soc. (3)* **28** (1983), 385–392.
2. R. Aharoni, König's duality theorem for infinite bipartite graphs, *J. London Math. Soc. (3)* **29** (1984), 1–12.
3. R. A. Brualdi, Strong transfinite version of König's duality theorem, *Monatsh. Math.* **75** (1971), 106–110.
4. L. A. Harrington, M. D. Morley, A. Scedrov, and S. G. Simpson (Eds.), "Harvey Friedman's Research on the Foundations of Mathematics," North-Holland, Amsterdam, 1985.
5. J. L. Hirst, "Combinatorics in Subsystems of Second-Order Arithmetic," Ph.D. thesis, Pennsylvania State University, 1987.
6. J. L. Hirst, Marriage theorems and reverse mathematics, *in* "Proceedings of the Workshop on Logic and Computation" (W. Sieg Ed.), AMS Contemporary Mathematics Series, Amer. Math. Soc., Providence, RI, 1990.
7. M. Holz, K. P. Podewski, and K. Steffens, "Injective Choice Functions," Lecture Notes in Mathematics, Vol. 1238, Springer-Verlag, Berlin, 1987.
8. C. G. Jockusch, Jr., Ramsey's theorem and recursion theory, *J. Symbolic Logic* **37** (1972), 81–89.
9. C. G. Jockusch and R. I. Soare, Π^0_1 classes and degrees of theories, *Trans. Amer. Math. Soc.* **173** (1972), 33–56.
10. H. A. Kierstead, An effective version of Dilworth's theorem, *Trans. Amer. Math. Soc.* **268** (1981), 63–77.
11. H. A. Kierstead, G. F. McNulty, and W. T. Trotter, A theory of recursive dimension for ordered sets, *Order* **1** (1984), 67–82.
12. D. König, "Theorie der endlichen und unendlichen Graphen" (1936), reprinted by Chelsea, New York, 1950.
13. L. Lovasz and M. D. Plummer, "Matching Theory," Annals of Discrete Mathematics, Vol. 29, North-Holland, Amsterdam, 1986.
14. A. Nerode and R. A. Shore (Eds.), "Recursion Theory," Proceedings of Symposia in Pure Mathematics, Vol. 42, Amer. Math. Soc., Providence, RI, 1985.
15. K. P. Podewski and K. Steffens, Injective choice functions for countable families, *J. Combin. Theory Ser. B* **21** (1976), 40–46.
16. H. Rogers, Jr., "Theory of Recursive Functions and Effective Computability," McGraw-Hill, New York, 1967.
17. J. H. Schmerl, Recursive colorings of graphs, *Canad. J. Math.* **21** (1980), 821–830.
18. J. H. Schmerl, The effective version of Brooks' theorem, *Canad. J. Math.* **34** (1982), 1036–1046.

19. S. G. SIMPSON, [1985], Reverse mathematics, *in* Ref. [14, pp. 461–472].
20. S. G. SIMPSON, Friedman's research on subsystems of analysis, *in* Ref. [4, pp. 137–160].
21. S. G. SIMPSON (Ed.), "Logic and Combinatorics," Contemporary Mathematics, Vol. 65, Amer. Math. Soc., Providence, RI, 1987.
22. S. G. SIMPSON, Subsystems of Z_2 and reverse mathematics, in Ref. [25, pp. 432–446].
23. S. G. SIMPSON, "Subsystems of Second Order Arithmetic," Springer-Verlag, Berlin, 1992.
24. R. L. SMITH, The consistency strength of some forms of Kruskal's theorems, *in* Ref. [4].
25. G. TAKEUTI, "Proof Theory," 2nd ed., North-Holland, Amsterdam, 1987.
26. H. TVERBERG, On Schmerl's effective version of Borrk's theorem *J. Combin. Theory Ser. B* **37** (1984), 27–30.