Contents lists available at SciVerse ScienceDirect

Annals of Pure and Applied Logic

journal homepage: www.elsevier.com/locate/apal

Expressive power of digraph solvability

Marc Bezem [a,*], Clemens Grabmayer [b], Michał Walicki [a]

[a] Department of Informatics, University of Bergen, P.O. Box 7803, 5020 Bergen, Norway
[b] Department of Philosophy, Utrecht University, P.O. Box 80125, 3508 TC Utrecht, The Netherlands

ARTICLE INFO

Article history:
Received 29 May 2011
Received in revised form 17 August 2011
Accepted 17 August 2011
Available online xxxx
Communicated by U. Kohlenbach

MSC:
03B30
03D99
03E25
05C63

Keywords:
Digraph
Kernel
Infinitary propositional logic
Reverse mathematics

ABSTRACT

A kernel of a directed graph is a set of vertices without edges between them, such that every other vertex has a directed edge to a vertex in the kernel. A digraph possessing a kernel is called solvable. Solvability of digraphs is equivalent to satisfiability of theories of propositional logic. Based on a new normal form for such theories, this equivalence relates finitely branching digraphs to propositional logic, and arbitrary digraphs to infinitary propositional logic. While the computational complexity of solvability differs between finite dags (trivial, since always solvable) and finite digraphs (NP-complete), this difference disappears in the infinite case. The existence of a kernel for a digraph is equivalent to the existence of a kernel for its lifting to an infinitely-branching dag, and we prove that solvability for recursive dags and digraphs is Σ^1_1-complete. This implies that satisfiability for recursive theories in infinitary propositional logic is also Σ^1_1-complete. We place several variants of the kernel problem in the axiomatic hierarchy and, in particular, prove as the main result that over RCA_0, solvability of finitely branching dags is equivalent to Weak König's Lemma. We then show that ZF proves solvability of trees and that solvability of forests requires at most a weak form of AC. Finally, a new equivalent of the full AC is formulated using solvability of complete digraphs.

1. Introduction

In a digraph (directed graph), a subset X of its vertices is called *independent* if the successors of vertices in X are not in X. A *kernel* of a digraph is an independent subset K of vertices such that there is an edge from every vertex outside of K to a vertex in K. Equivalently, a kernel of a digraph is a subset K of vertices such that a vertex is in K if and only if none of its successors is in K. This concept corresponds to, and originates from, the concept of *solution* of binary relations as introduced by von Neumann and Morgenstern in their book *Theory of Games and Economic Behavior* [16]. To see the connection with game theory, consider a two-player game with alternating moves, with vertices of a digraph G representing the positions and the edges the possible moves. Then any kernel of G describes a stable situation for one of the players: a player A in a position outside a kernel K of G can always choose to move to a position in K, forcing the opponent B to move out of K, and so on. Thus A can stay outside K for the rest of the game, whereas B is forced to stay inside K. Depending on the other rules of the game, this can be a winning strategy for A.

Today, kernel theory is an active research field in graph theory. Its main question concerns sufficient conditions for the existence of kernels in finite digraphs, e.g., [1,6,7,9], with a recent overview in [2].

* Corresponding author.
E-mail addresses: bezem@ii.uib.no (M. Bezem), clemens@phil.uu.nl (C. Grabmayer), michal@ii.uib.no (M. Walicki).

0168-0072/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.apal.2011.08.004

In this paper we aim at more general formulations. The expressive power in the title refers to, on the one hand, the recursion-theoretic complexity of the problem of kernel existence and, on the other hand, to the axiomatic strength of solvability of digraphs of various classes. These questions, apparently of purely graph theoretic flavor, have strong logical import. Section 2 starts with the definitions of kernels and solutions of digraphs, and introduces functions between digraphs and propositional theories mapping satisfiable theories to solvable digraphs and vice versa. The mappings relate infinitely branching digraphs to infinitary propositional logic, and finitely branching ones to the usual propositional logic. A new, simple proof of this otherwise known equivalence, [3,5], is given. Section 3 presents simple generalizations of two known results, useful for a general study of the digraph solvability and applied in later sections. Section 4 presents the first main result of the paper: Σ^1_1-completeness of the solvability of recursive digraphs and, as a consequence of the above equivalence, of satisfiability of recursive theories in infinitary propositional logic. Section 5 shows axiomatic strength of the solvability of some classes of digraphs. Section 5.1 presents the other main result of the paper: equivalence, over RCA_0, of solvability of finitely branching dags and countable compactness (or Weak König's Lemma). Section 5.2, noting solvablity of trees in ZF, shows the solvability of forests from a very weak form of the Axiom of Choice, AC(2), assuming a choice function for any collection of sets, all having 2 elements. It then gives a new equivalent of AC in terms of the solvability of complete digraphs.

2. Basic definitions and facts

A *directed graph* (a *digraph*) is a pair $\mathsf{G} = \langle G, E_{\mathsf{G}} \rangle$, where G is a set of vertices and $E_{\mathsf{G}} \subseteq G \times G$ is a binary relation representing the directed edges of G. When G is understood, we write E instead of E_{G}. A *directed acyclic graph* (a *dag*) is a digraph without cycles.

For a vertex $x \in G$, we denote by $E(x) := \{y \in G \mid E(x, y)\}$ the set of *successors of* x, and by $E^{\smile}(x) := \{y \in G \mid E(y, x)\}$ the set of *predecessors of* x with respect to the edge relation of G. This notation is extended to subsets of vertices, for example, for all $X \subseteq G$, we let $E^{\smile}(X) := \bigcup_{x \in X} E^{\smile}(x)$. A *sink* (*source*) in G is a vertex $x \in G$ without successors (predecessors) and $sinks(\mathsf{G}) = \{x \in G \mid E(x) = \varnothing\}$ denotes the set of sinks of G.

Given a digraph $\mathsf{G} = \langle G, E \rangle$, we denote by $\underline{\mathsf{G}} = \langle G, \underline{E} \rangle$ its *underlying* undirected graph, obtained by turning every directed edge $\langle x, y \rangle$ into an undirected one $\{x, y\}$, i.e., with $\underline{E} = \{\{x, y\} \mid \langle x, y \rangle \in E\}$.[1]

We give a general definition of path since we need both finite and infinite paths. Consider the digraph $\langle \mathbb{Z}, Succ \rangle$, where $\mathbb{Z}$ denotes the integers and $Succ = \{\langle n, n+1 \rangle \mid n \in \mathbb{Z}\}$. An *interval graph* is the subgraph induced by $I \subseteq \mathbb{Z}$ where for all $i, j \in I$ with $i < j$ we have $i + k \in I$ for all $0 < k < j - i$. A *digraph morphism* $h : \mathsf{F} \to \mathsf{G}$ is a mapping of vertices $h : F \to G$ preserving the edge relation, i.e., when extended pointwise to sets, satisfying $h(E_{\mathsf{F}}(x)) \subseteq E_{\mathsf{G}}(h(x))$ for all $x \in F$. A *path* in G is a digraph morphism h from an interval subgraph of $\mathbb{Z}$ to G. In particular, an ω-path is such a morphism from the interval graph consisting of all non-negative integers. Note that any cycle gives an ω-path. An *integer graph* is a digraph isomorphic to $\langle \mathbb{Z}, Succ \rangle$. An *ancestor* (*descendant*) of any vertex x of G is a vertex y such that there is a path in G from y to x (from x to y), and $E^{\smile *}_{\mathsf{G}}(x)$ ($E^*_{\mathsf{G}}(x)$) is the set of x's ancestors (descendants) in G.

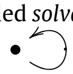
A *kernel* of a digraph $\mathsf{G} = \langle G, E \rangle$ is a subset of vertices $K \subseteq G$ such that:

(i) $G \setminus K \supseteq E^{\smile}(K)$ (K is an *independent set* in G), and
(ii) $G \setminus K \subseteq E^{\smile}(K)$ (K is *dominating*: from every non-kernel vertex there is at least one edge to a kernel vertex).

The equivalence between the existence of kernels and the satisfiability of propositional theories that we explore in this paper arises from an equivalent definition of kernels, the notion of *solution*. Let $\mathsf{G} = \langle G, E \rangle$ be a digraph. An assignment $\alpha \in \{\mathbf{0}, \mathbf{1}\}^G$ (of truth-values to the vertices of G) is a *solution* of G if for every $x \in G : \alpha(x) = \mathbf{1} \Leftrightarrow \alpha(E(x)) \subseteq \{\mathbf{0}\}$ or, equivalently, if for every $x \in G$:

$$\big(\alpha(x) = \mathbf{1} \ \wedge\ \alpha(E(x)) \subseteq \{\mathbf{0}\}\big) \quad \vee \quad \big(\alpha(x) = \mathbf{0} \ \wedge\ \mathbf{1} \in \alpha(E(x))\big). \tag{2.1}$$

The set of solutions of G is denoted by $sol(\mathsf{G})$. G is called *solvable* iff $sol(\mathsf{G}) \neq \varnothing$. By $\alpha^{\mathbf{1}}$ we denote the set $\{x \in G \mid \alpha(x) = \mathbf{1}\}$.

The simplest example of an unsolvable digraph is  For all digraphs G and all assignments $\alpha \in \{\mathbf{0}, \mathbf{1}\}^G$ it holds:

$$\alpha \in sol(\mathsf{G}) \iff \alpha^{\mathbf{1}} = \mathsf{G} \setminus E^{\smile}(\alpha^{\mathbf{1}}) \iff \alpha^{\mathbf{1}} \text{ is a kernel of } \mathsf{G}. \tag{2.2}$$

Now, a digraph G induces a (possibly infinitary) propositional theory $\mathcal{T}(\mathsf{G})$ by taking, for each $x \in G$, the formula $x \leftrightarrow E^{\neg}(x)$, where $E^{\neg}(x) = \bigwedge_{y \in E(x)} \neg y$ with the convention that $\bigwedge \varnothing = \mathbf{1}$.[2] Letting $mod(\mathsf{T})$ denote all models of a theory T, it is easy to see that (2.1) entails:

$$sol(\mathsf{G}) = mod(\mathcal{T}(\mathsf{G}))\ . \tag{2.3}$$

[1] Notations $E(x, y)$, $\langle x, y \rangle \in E$, $y \in E(x)$ and $x \in E^{\smile}(y)$ are used interchangeably for denoting that x is E-related to y.

[2] Satisfiability of such a theory is equivalent to the existence of solutions for the corresponding system of boolean equations. This motivates the name "solution", which was also used for kernels in the early days of kernel theory, e.g., [16], p. 588, or [13].

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004

As a consequence of (2.2) and (2.3), determining kernels of digraphs can be viewed as a special case of determining the models of theories in propositional logic. These theories are in ordinary, finitary propositional logic (PL), if G is finitely branching, and in infinitary propositional logic (PL^∞), otherwise.

PL^∞ denotes propositional logic with formulas of finite depth, formed over an arbitrary set of propositional variables by unary negation and (possibly) infinite conjunction.[3] PL^ω denotes the restriction of PL^∞ to propositions over a countable set of propositional variables and with conjunction of arity ω.

Conversely, consistency of propositional theories can be reduced to solvability of corresponding digraphs. Every PL^∞-theory T can be represented as a digraph $\mathcal{G}(\mathsf{T})$ whose solutions are in bijective correspondence to the models of T. This is particularly simple if a theory T is given as a set of equivalences in the *digraph normal form*

$$y \leftrightarrow \bigwedge_{i \in I_y} \neg x_i \ , \tag{2.4}$$

where all y, x_i are variables, and where every variable occurs at most once on the left of $\leftrightarrow$. The digraph $\mathcal{G}(\mathsf{T})$ is obtained by taking variables as vertices and, for every formula, introducing edges $\langle y, x_i \rangle$ for all $i \in I_y$. In addition, for every variable z *not* occurring on the left of any $\leftrightarrow$, we add a new vertex $\overline{z}$ and two edges $\langle z, \overline{z} \rangle$ and $\langle \overline{z}, z \rangle$. This last addition ensures that each variable z of T which would become a sink of $\mathcal{G}(\mathsf{T})$, and hence could only be assigned **1** by any solution of $\mathcal{G}(\mathsf{T})$, can now also be assigned **0** (when the respective $\overline{z}$ is assigned **1**). Letting $V(\mathsf{T})$ denote all variables of T, and $sol(\mathsf{X})|_Y$ the restriction of assignments in $sol(\mathsf{X})$ to the variables in Y, we have that

$$mod(\mathsf{T}) = sol(\mathcal{G}(\mathsf{T}))|_{V(\mathsf{T})}. \tag{2.5}$$

An arbitrary theory T, not in the above form (2.4), can be translated into an equisatisfiable theory in this form. The idea is to introduce new variables for every subformula, and to express the relation between the formula and its direct subformulas, using the normal form (2.4), in the graph structure. We assume vertices v for every propositional variable v (plus enough extra vertices used at every stage of the definition). We define a mapping $\mathcal{G}$ from formulas to digraphs in two steps. The first step is to define a rooted dag $\mathcal{G}'(\varphi)$, essentially the parse tree of φ, by recursion on φ:

(i) $\mathcal{G}'(\bigwedge_i \varphi_i)$ consists of a new vertex n, all $\mathcal{G}'(\neg\varphi_i)$ and new edges from n to the root of each $\mathcal{G}'(\neg\varphi_i)$;
(ii) $\mathcal{G}'(\neg\varphi)$ consists of a new vertex n', $\mathcal{G}'(\varphi)$ and a new edge from n' to the root of $\mathcal{G}'(\varphi)$;
(iii) $\mathcal{G}'(v)$ consists of the vertex v as a sink.

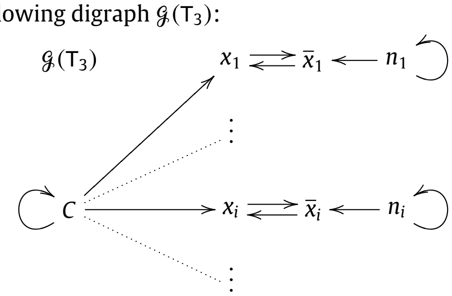
Then we extend the rooted dag $\mathcal{G}'(\varphi)$ to digraph $\mathcal{G}(\varphi)$ in the following way. First, add a new vertex n'' to $\mathcal{G}'(\varphi)$ with a loop and an edge to the root of $\mathcal{G}'(\varphi)$. This enforces solutions in which the root of $\mathcal{G}'(\varphi)$ is **1**. Second, for every sink v, add a new vertex $\overline{v}$ and one new edge from v to $\overline{v}$ and one from $\overline{v}$ to v. As explained above, this makes it possible to assign any truth value to v. The extended graph has neither sources nor sinks. This completes the description of $\mathcal{G}(\varphi)$.

Some obvious simplifications of $\mathcal{G}(\varphi)$ can be made. The first one is to remove double negations by putting $\mathcal{G}'(\neg\neg\varphi) = \mathcal{G}'(\varphi)$. The second one is to remove a possible double negation from the vertex with the loop to the root of $\mathcal{G}(\neg\varphi)$, that is, when the main formula is negative. The third one is to put $\mathcal{G}(\neg v) = \overline{v}$, in combination with appropriate edges between v and $\overline{v}$.

Finally, we collect the set of all $\mathcal{G}(\varphi)$, $\varphi \in \mathsf{T}$, to obtain $\mathcal{G}(\mathsf{T})$, satisfying Eq. (2.5) by construction. We illustrate the translation with some examples.

Example 2.6. Simplifying the digraphs for $\mathsf{T}_1 = \{\neg x\}$ and $\mathsf{T}_2 = \{\neg x \vee y\}$, we obtain:

The theory T_3 with literals $\neg x_i$ for all $i \in \mathbb{N}$ and one infinitary disjunction $C = \bigvee_{i \in \mathbb{N}} x_i = \neg \bigwedge_{i \in \mathbb{N}} \neg x_i$, gives rise to the following digraph $\mathcal{G}(\mathsf{T}_3)$:



[3] Many variations are possible, e.g., one could allow well-founded formulas or take disjunction instead of conjunction. Binary connectives, such as $\leftrightarrow$, are assumed to be encoded, but could have been added.

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004

$\mathcal{G}(\mathsf{T}_3)$ can be obtained from the finite subgraph G_1 induced by $\{C, x_1, \bar{x}_1, n_1\}$ by replicating the subgraph induced by these vertices except C. The inconsistency of T_3 is reflected by the unsolvability of $\mathcal{G}(\mathsf{T}_3)$ which, in turn, reduces to the unsolvability of the finite subgraph G_1. This suggests a possibility of reducing satisfiability of theories in PL^∞ to solvability of finite graphs, instead of to satisfiability of finite subtheories. Such an investigation, however, is not the topic of the present paper.

Eq. (2.3) for digraphs, and Eq. (2.5) for propositional theories establish a back-and-forth correspondence between satisfiable propositional theories and solvable digraphs. Various statements of sufficient conditions for the existence of kernels, e.g., [1,2,6,7,9], can be now applied for determining satisfiability of PL theories and vice versa. The following investigation of the placement of variants of the kernel problem in the recursive and axiomatic hierarchy, invokes this equivalence — sometimes, merely for facilitating the proof, and at other times for drawing a conclusion in one field, having obtained it in the other.

3. Some general facts about solvability

This section presents two results on solvability that are of independent, general interest. They are not new but only generalize earlier known facts by discharging some unnecessary assumptions. Section 3.1 shows that every digraph has a sinkless subgraph with essentially the same solution set. The proof also yields the well-known fact that every finite dag, and even every dag without ω-paths, has a unique solution, since the relevant sinkless subgraphs of such dags are empty. Section 3.2 shows that solutions for arbitrary digraphs can be represented as solutions for appropriate, infinitely branching dags.

3.1. Induced assignment

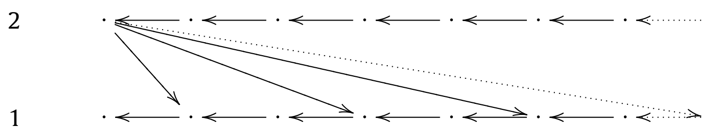
This subsection uses induction on the set of ordinals with cardinality at most the cardinality of the graph in question. All quantifications are relative to this set of ordinals and we use κ to denote such ordinals (λ for limits). The construction sequentially removes vertices from the graph until a fixed-point, a sinkless subgraph with essentially the same solution set, is reached.

Assigning **1** to $sinks(\mathsf{G})$ may force values at some other vertices. This was implicitly used already in the proof of Richardson's theorem (finitely branching digraph without odd cycles is solvable, [13]), and then formulated more generally in [11] for irreflexive graphs. Since irreflexivity is unnecessary, we spell out and justify the construction in full generality. It is based on repeatedly removing sinks and their predecessors. The induced (partial) assignment σ is defined by ordinal recursion as follows:

$$\left.\begin{array}{l} C_0 = G, \quad \text{for the given digraph } \mathsf{G} = \langle G, E\rangle \\ \mathsf{C}_\kappa \text{ is the subgraph induced by } C_\kappa \\ \sigma^{\mathbf{1}}_\kappa = sinks(\mathsf{C}_\kappa) \\ \sigma^{\mathbf{0}}_\kappa = E^{\smile}(\sigma^{\mathbf{1}}_\kappa) \cap C_\kappa \\ C_{\kappa+1} = C_\kappa \setminus (\sigma^{\mathbf{1}}_\kappa \cup \sigma^{\mathbf{0}}_\kappa) \quad \text{and} \quad C_\lambda = \bigcap_{\kappa<\lambda} C_\kappa \text{ for limit } \lambda \\ \\ G^\circ = \bigcap_\kappa C_\kappa \text{ and } \mathsf{G}^\circ \text{ is the induced subgraph} \\ \sigma^{\mathbf{v}} = \bigcup_\kappa \sigma^{\mathbf{v}}_\kappa, \ \text{ for } \mathbf{v} \in \{\mathbf{0}, \mathbf{1}\} \\ \text{The induced assignment is given by } \sigma = \{\langle x, \mathbf{v}\rangle \mid x \in \sigma^{\mathbf{v}}\}. \end{array}\right\} \tag{3.1}$$

Note that σ is well-defined since there is no overlap between the sets $\sigma^{\mathbf{v}}_\kappa$, when κ or $\mathbf{v}$ varies. For finitely branching digraphs ω iterations suffice. In general, even if any path to a sink is finite, one may need transfinite ordinals to reach a fixed-point, but one never needs ordinals with cardinality larger than that of the graph. In the following example the (empty) fixed-point is reached in $\omega + \omega$ iterations, while the infinitely branching graph is countable.

Example 3.2. In the digraph below, after ω iterations only vertices at level 1 have induced values. The digraph has the induced (unique) solution when, after $\omega + \omega$ iterations, G° becomes empty.



The example is an instance of a general fact, namely, the solvability of digraphs without ω-paths. The latter follows from the next proposition, allowing the reduction of many solvability questions to solvability of sinkless digraphs.

Proposition 3.3. *For any* G, *with* σ, C_κ *and* G° *as defined in* (3.1)*:*

1. $G^\circ = C_\kappa = C_{\kappa+1}$ *for some* κ *with cardinality at most* $|G|$
2. $sinks(\mathsf{G}^\circ) = \varnothing$
3. $sol(\mathsf{G}) = \{\alpha \cup \sigma \mid \alpha \in sol(\mathsf{G}^\circ)\}$, *in particular,* $sol(\mathsf{G}) \neq \varnothing \Leftrightarrow sol(\mathsf{G}^\circ) \neq \varnothing$.

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004

Proof. (1). For finite graphs this is obvious, so let G be infinite and assume by contradiction that $C_\kappa \setminus C_{\kappa+1}$ is non-empty for all κ with cardinality at most $|G|$. Then there would be an injection $\{\kappa : |\kappa| \le |G|\} \to G$, which is impossible.

(2). This follows directly from the previous point, since $C_\kappa = C_{\kappa+1}$ implies that there are no sinks in $\mathsf{C}_\kappa = \mathsf{G}^\circ$.

(3) . Let $\alpha \in sol(\mathsf{G})$. By induction we show that for all κ, $\sigma^{\mathbf{1}}_\kappa \subseteq \alpha^{\mathbf{1}}$ and $\sigma^{\mathbf{0}}_\kappa \subseteq \alpha^{\mathbf{0}}$. This is obvious for $\sigma^{\mathbf{1}}_0 = sinks(\mathsf{G})$ and, consequently, also for $\sigma^{\mathbf{0}}_0 = E^{\smile}(sinks(\mathsf{G}))$. Inductively, if $x \in \sigma^{\mathbf{1}}_\kappa = sinks(\mathsf{C}_\kappa)$, then $E(x) \subseteq \bigcup_{\kappa'<\kappa} \sigma^{\mathbf{0}}_{\kappa'}$ (since $y \in E(x) \cap \sigma^{\mathbf{1}}_{\kappa'}$ would imply $x \in \sigma^{\mathbf{0}}_{\kappa'}$ and hence $x \notin \sigma^{\mathbf{1}}_\kappa$). By the induction hypothesis we get $E(x) \subseteq \alpha^{\mathbf{0}}$, and so $\alpha(x) = \mathbf{1}$. If $x \in \sigma^{\mathbf{0}}_\kappa$ then $x \in E^{\smile}(\sigma^{\mathbf{1}}_\kappa) \subseteq E^{\smile}(\alpha^{\mathbf{1}})$, so $\alpha(x) = \mathbf{0}$. Hence any $\alpha \in sol(\mathsf{G})$ extends σ.

Now let $x \in G^\circ$ and $y \in E(x)$. If $y \notin G^\circ$, then $y \in \sigma^{\mathbf{0}}$, since $y \in \sigma^{\mathbf{1}}$ would imply $x \notin G^\circ$. In other words, all successors of x outside G° have $\alpha(x) = \mathbf{0}$, which means that α restricted to G° is a solution of G°. By similar arguments, any solution of G° can be extended to a solution of G by joining σ. □

When $G^\circ = \varnothing$, $sol(\varnothing) = \{\varnothing\} \neq \varnothing$ and, by point (3), G has only one solution σ. This is the case, for instance, for finite dags, which appears to be the first theorem in kernel theory from [16]. More generally, Proposition 3.3 has the following corollary. The absence of ω-paths means that the digraph is well-founded in the forward direction and, in particular, contains no cycles.

Corollary 3.4. *Every digraph without an ω-path has a unique solution.*

3.2. Lifting digraphs to dags

Every digraph G (with at least one edge) can be transformed into an infinitely branching dag G^ω – preserving and reflecting the solutions – as follows.

The *(dag-)lifting* of a digraph $\mathsf{G} = \langle G, E\rangle$ is the digraph $\mathsf{G}^\omega = \langle G^\omega, E^\omega\rangle$ with:

$$\begin{aligned} G^\omega &:= G \times \omega \\ E^\omega &:= \{\langle n_i, m_j\rangle \mid \langle n, m\rangle \in E \wedge i < j\} \end{aligned} \tag{3.5}$$

where, here and below, the vertices of G^ω, pairs in $G \times \omega$, are denoted by indexing the vertex in the first component, that is, a pair $\langle n, i\rangle$ is written as n_i. The graph G^ω is indeed a dag: it contains no cycles, since there can be a path of positive length from y_i to y_j only when $i < j$. Also, $sinks(\mathsf{G}^\omega) = sinks(\mathsf{G}) \times \omega$ and G has an ω-path iff G^ω has an ω-path.

For every function $f : G \to X$, its *lifting* $f^\omega : G^\omega \to X$ is given by:

$$f^\omega(n_i) := f(n) \qquad (\text{for all } n \in G \text{ and } i \in \omega)\,. \tag{3.6}$$

For a set (of functions) F we denote $F^\omega = \{f^\omega \mid f \in F\}$.

Lemma 3.7. *For every G, $(sol(\mathsf{G}))^\omega \subseteq sol(\mathsf{G}^\omega)$.*

Proof. By definition, for every vertex $x \in G$ and for all $i \in \omega$:

$$E^\neg(x_i) = \bigwedge_{m \in E(x), j>i} \neg m_j \qquad (\text{so } E^\neg(x_i) = \mathbf{1} \text{ for all } x \in sinks(\mathsf{G})).$$

Let $\alpha \in sol(\mathsf{G})$, then $\alpha(x) = \alpha(E^\neg(x))$. By (3.6) we have $\alpha^\omega(x_i) = \alpha(x) = \alpha(E^\neg(x)) = \alpha^\omega(E^\neg(x_i))$ for all x, i. It follows that $\alpha^\omega \in sol(\mathsf{G}^\omega)$. □

We say that a $\beta \in sol(\mathsf{G}^\omega)$ is *stable on a vertex* $n \in G$ if $\forall i \forall j\ (\beta(n_i) = \beta(n_j))$ and call β *stable* if β is stable on every vertex of G.

Lemma 3.8. *For every G, every $\beta \in sol(\mathsf{G}^\omega)$ is stable.*

Proof. G^ω has the property that $\forall n \in G\ \forall i \forall j > i\ (E^\omega(n_j) \subseteq E^\omega(n_i))$. Now, if $\beta(n_i) = \mathbf{1}$, that is, $\beta(E^\omega(n_i)) \subseteq \{\mathbf{0}\}$, then also $\beta(n_k) = \mathbf{1}$ for all $k \ge i$. If $\beta(n_i) = \mathbf{0}$, there is an $m_j \in E^\omega(n_i)$ with $\beta(m_j) = \mathbf{1}$ and, by the previous case, $\beta(m_{j'}) = \mathbf{1}$ for all $j' \ge j$. Hence $\beta(n_k) = \mathbf{0}$ for all $k \ge i$. □

The immediate corollary of the two lemmata is the following:

Theorem 3.9. *For every G, $(sol(\mathsf{G}))^\omega = sol(\mathsf{G}^\omega)$.*

In particular, G is solvable if and only if G^ω is. A special case of the above gives, for a finite cyclic G, its infinite, acyclic counterpart. The paradigmatic example is lifting a single loop to the infinite Yablo dag, the digraph $\langle \mathbb{N}, <\rangle$, [17]. The special case of finite, sinkless graphs was addressed in [4] and we merely generalized it allowing infinite graphs and sinks. When digraphs are infinitely branching, the theorem allows us to equate the problem of solvability of arbitrary digraphs and the

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004

problem of solvability of dags. Consequently, many results characterizing the solvability of arbitrary digraphs, also hold for the solvability of arbitrary dags.

4. Recursion-theoretic complexity

This section contains the first main result of the paper, namely, that solvability of recursive digraphs is Σ^1_1-complete and that, as a consequence, this also holds for satisfiability of (clausal) recursive PL^{ω}-theories. We begin with a simple argument showing that even binary recursive trees (which are always solvable in systems at least as strong as WKL_0, by the main result in Section 5.1) may fail to have recursive solutions.

A consistent, recursive, propositional theory may have no recursive models. In terms of digraphs, a solvable, recursive digraph may have no recursive solutions. Since lifting (3.5) of a recursive digraph yields a recursive dag, there are recursive dags with no recursive solutions. The following gives a direct proof of this fact, even for binary trees, using a variation of the Kleene tree (as explained to us by Dag Normann). Note that a tree can be viewed as a dag (with unique paths from the root to each vertex).

Proposition 4.1. *There exists a recursive binary tree T without recursive solutions.*

Proof. The argument is based on the existence of two recursively enumerable but recursively inseparable sets A and B. This means that $A \cap B = \varnothing$ and there is no recursive set C such that $A \subset C$ and $B \subset \overline{C}$. Let recursive functions a and b enumerate these sets A and B, respectively. We define, uniformly recursive in n, linear trees T_n consisting of all sequences $0^0, 0^1, 0^2, \ldots, 0^k$ where $0^0 = \epsilon$ and k is such that:

(1) $a(i) \neq n \wedge b(i) \neq n$ for all $i < k/2$,
(2) $k = 2i$ if i is minimal such that $a(i) = n$, and
(3) $k = 2i + 1$ if i is minimal such that $b(i) = n$.

This means that $T_n = 0^*$ if $n \notin A \cup B$. Otherwise, T_n is a finite path with an even number of edges if $n \in A$ and an odd number if $n \in B$. The recursive tree T consists now of all prefixes of sequences 0^n10^k for all $n \in \mathbb{N}$ and $0^k \in T_n$. If there exists a recursive $\alpha \in sol(T)$, then the set $C = \{n \in \mathbb{N} \mid \alpha(0^n1) = \mathbf{1}\}$ is recursive and separates A and B. Contradiction. □

Before stating the main results of this section, we briefly recall Theorem XX from Rogers [14, Section 16.3]. Let FPT be the following subset of $\mathbb{N}$:

$$\mathrm{FPT} = \{z \mid \varphi_z \text{ is the characteristic function of a finite-path tree}\}.$$

Here φ_z is the *partial recursive function* with Kleene index z. A *tree* in [14] is a prefix-closed set of finite sequences of natural numbers encoded by so-called *sequence numbers*. It is convenient to assume that every natural number is a sequence number and that 0 encodes the empty sequence. For brevity, we will say that *z encodes a tree* if φ_z is the characteristic function of a tree. In this setting, Theorem XX states that FPT is a Π^1_1-complete set. Consequently, the complement of FPT is a Σ^1_1-complete set:

$$\overline{\mathrm{FPT}} = \{z \mid \text{if } z \text{ encodes a tree, then this tree has an } \omega\text{-path}\}.$$

The set $\overline{\mathrm{FPT}}$ is instrumental in proving other sets Σ^1_1-complete by means of a so-called *many-one reduction*. Let $A, B \subseteq \mathbb{N}$. We write $A \leq_m B$ to denote the fact that there is a many-one reduction from A to B, that is, a total recursive function $f : \mathbb{N} \to \mathbb{N}$ such that $n \in A$ if and only if $f(n) \in B$, for every $n \in \mathbb{N}$. If $A \leq_m B$ and A is Σ^1_1-complete and B is Σ^1_1, then B is Σ^1_1-complete as well.

The set that we will prove to be Σ^1_1-complete is GSOL defined by

$$\mathrm{GSOL} = \{z \mid \text{if } z \text{ encodes a digraph, then this digraph is solvable}\}.$$

We must first make clear what it means for a Kleene index z to encode a digraph. We take this to mean that φ_z is the characteristic function of a set of pairs of natural numbers, where the pairs represent the edges of the digraph. We assume a primitive recursive, surjective encoding of such pairs as natural numbers.

Before we give the details of a many-one reduction from $\overline{\mathrm{FPT}}$ to GSOL we provide an intuitive sketch of this reduction. It is a small step to view a tree in the sense of Rogers as a digraph: the vertices are finite sequences of natural numbers and the edges point from any σ to each finite sequence σx, extending σ by x, in the tree. Recall from Corollary 3.4 that any finite-path tree, that is, a tree without ω-paths, has a unique solution. We can standardize this solution by splitting every edge in two, adding intermediate vertices and appropriate edges. Then a finite-path tree leads to a digraph in which all sinks have even distance to the root. Because of the absence of ω-paths, the unique solution now assigns $\mathbf{1}$ to the root. We can spoil this solution by adding an edge from the root to itself (a single loop): the resulting digraph is no longer solvable.

If the tree has an ω-path, the effect of splitting the edges in two is different. To analyze this case, assume the tree has an ω-path and split all edges as in the previous paragraph. Let I be the set of new, intermediate vertices and O be the set of other vertices, (the old vertices of the tree). The resulting digraph is still a tree and still has the solution assigning $\mathbf{1}$ to all vertices in O and $\mathbf{0}$ to all vertices in I. But there are now other solutions as well. These solutions are among the solutions given in

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004

Proposition 3.3, point 3, but one is most easily described here directly. Distinguish between vertices not on any ω-path and all others. Obviously the root is on an ω-path. For vertices not on any ω-path we keep the truth values as in the previous paragraph, that is, **1** for all vertices in O and **0** for those in I. This is correct for sinks that are all in O, but also for all other vertices not on any ω-path. The reason is simply that if a vertex is not on any ω-path , then none of its successors are on any ω-path . On the other hand, for vertices on an ω-path we swap the above truth values, that is, we assign **0** if such a vertex is in O and **1** if such a vertex is in I. This is correct for the following reason. If a vertex $x \in O$ is on an ω-path , then x must have at least one successor on an ω-path , and all successors of x are in I. This means that at least one successor of x gets assigned **1**, justifying $x = \mathbf{0}$. Furthermore, if a vertex $x \in I$ is on an ω-path , then its unique successor in O is also on an ω-path and gets assigned **0**, justifying $x = \mathbf{1}$. So due to the ω-path, there exists a solution assigning **0** to the root. This assignment is still a solution when we add a single loop to the root.

In summary, if T is a tree and $G(T)$ is the digraph extending T by splitting edges in two and adding a single loop at the root, then we have:

$$T \text{ has an } \omega\text{-path} \iff G(T) \text{ is solvable.}$$

We are now in a position to give the details of the many-one reduction of $\overline{\text{FPT}}$ to GSOL. Readers who are already convinced by the informal explanation above may skip this rather technical paragraph. We multiply sequence numbers by two so that odd numbers become available for intermediate vertices. Let φ_z be a partial recursive function. We define the partial recursive function ψ as follows[4]:

$$\begin{array}{ll}
\psi(\langle 0, 0\rangle) = \varphi_z(0) & \text{(the single loop at the root if the tree is non-empty)} \\
\psi(\langle 2x{-}1, 2x\rangle) = \varphi_z(x) & \text{if } x > 0 \text{ (} 2x - 1 \text{ represents an intermediate vertex)} \\
\psi(\langle 2x, 2y{-}1\rangle) = \varphi_z(y) & \text{if } x \text{ is } y \text{ without the last element (} 2y - 1 \text{ intermediate)}.
\end{array}$$

Otherwise (if none of the above cases apply):

$$\begin{array}{ll}
\psi(\langle x', y'\rangle) = 0 & \text{if } x' \text{ is odd or } y' \text{ is odd} \\
\psi(\langle 2x, 2y\rangle) = 0 & \text{if } \varphi_z(y) \le \varphi_z(y') \text{ for all prefixes } y' \text{ of } y \\
\psi(\langle 2x, 2y\rangle) = \uparrow & \text{in all other cases (} \uparrow \text{ denotes divergence).}
\end{array}$$

Recall that characteristic functions are total, binary functions. The last two cases are designed to ensure that ψ is encoding a digraph only if φ_z is the characteristic function of a tree.

Since the definition of ψ is uniformly recursive in z, it follows that $\psi = \varphi_{f(z)}$ for some recursive function f. Given the informal explanation in the previous paragraph, it is not difficult to verify that f is the desired many-one reduction of $\overline{\text{FPT}}$ to GSOL.

Theorem 4.2. *The solvability problem for digraphs,* GSOL*, is Σ^1_1-complete.*

Proof. Since we have $\overline{\text{FPT}} \le_m$ GSOL by the above reduction, it suffices to prove that GSOL is Σ^1_1. We first note that encoding a digraph means being a total recursive function on all pairs of natural numbers, which has only arithmetical complexity. Next, we give a Σ^1_1-formula in z that expresses solvability for digraphs encoded by z:

$$\exists K\ \forall n\big[\, n \in K \ \leftrightarrow\ \forall n'(\, EdgeIn(n, n', z) \ \rightarrow\ n' \notin K\,)\,\big].$$

Here n, n' vary over natural numbers, and K over sets of natural numbers, and $EdgeIn(n, n', z)$ is a Σ^0_1-formula equivalent to $\varphi_z(\langle n, n'\rangle) = 1$. □

Natural questions at this point are: what is the complexity of digraph solvability for finitely branching digraphs? And for dags? Define the following sets

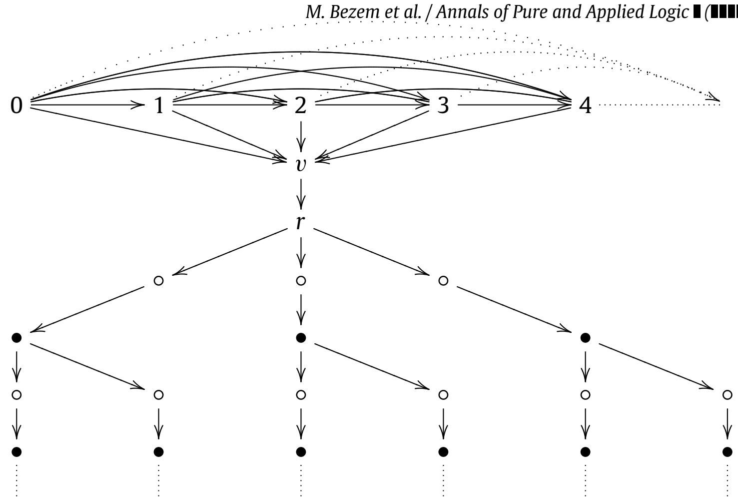
FBGS = $\{z \mid$ if z encodes an fb digraph, then this digraph is solvable$\}$

DSOL = $\{z \mid$ if z encodes a dag, then this dag is solvable$\}$.

Recall that solvability of fb digraphs is equivalent to satisfiablility of theories in ordinary propositional logic, by Eqs. (2.3) and (2.5). This means that the 'solvability part' of FBGS is rather easy and that the complexity of FBGS comes from the encoding of digraphs, complicated by the 'fb' requirement. However, this complexity is still arithmetical and does not exceed Π^0_3.

Regarding dags, we note that adding a single loop to the root of a tree, such as done in $G(T)$, results in a digraph and not in a dag. In order to get a dag $D(T)$ rather than a digraph, instead of the single loop we add a new vertex v with an edge to the root r of the tree (with the edges split in two as before), plus a Yablo dag $\langle \mathbb{N}, <\rangle$, [17], with edges from every vertex $n \in \mathbb{N}$ to v.

[4] The arguments $\langle x, y\rangle$ of ψ denote codes of pairs of numbers. Arguments of φ_z, on the other hand, are understood as sequence numbers. These two codings are unrelated.



This has the same effect as the single loop at r, namely spoiling solutions assigning **0** to v, that is, **1** to r. This informal argument proves $\overline{\mathrm{FPT}} \leq_m$ DSOL. We now get that DSOL is Σ^1_1-complete since GSOL is Σ^1_1 and acyclicity is of arithmetical complexity.

Eq. (2.3) relates satisfiability of infinitary propositional logic with digraph solvability. One would expect both to have the same analytical complexity. This will turn out to be the case, but we must first define what it means for a Kleene index to encode a theory in infinitary propositional logic. There are many possibilities here and we minimize technicalities by considering *infinitary clausal theories*.

A *literal* is a propositional variable or its negation. A *clause* is a (possibly infinite) set of literals representing a disjunction. A PL^ω-theory in infinitary clausal form can be encoded by a ternary predicate C on $\mathbb{N}$ representing the clauses:

$$C_i := \{p_j \mid C(i,j,1)\} \cup \{\neg p_j \mid C(i,j,2)\}, \quad \text{for all } i \in \mathbb{N}.$$

We take a Kleene index z to encode the theory consisting of all clauses C_i if φ_z is the characteristic function for the predicate C. We remark that the complexity of encoding a theory is arithmetical. We define the set ISAT by:

$$\mathrm{ISAT} = \{z \mid \text{if } z \text{ encodes a } \mathrm{PL}^\omega\text{-theory, then this theory is satisfiable}\}$$

Satisfiability can be expressed by a Σ^1_1-formula:

$$\exists M \forall i \exists j\, [(j \in M \wedge \mathit{LitIn}(z,i,j,1)) \vee (j \notin M \wedge \mathit{LitIn}(z,i,j,2))],$$

where M varies over sets of natural numbers and $\mathit{LitIn}(z,i,j,k)$ is a Σ^0_1-formula equivalent to $\varphi_z(i,j,k) = 1$. This establishes $\mathrm{ISAT} \in \Sigma^1_1$, and we now sketch the many-one reduction GSOL $\leq_m$ ISAT.

Given a graph $\mathsf{G} = \langle \mathbb{N}, E\rangle$, the construction yielding Eq. (2.3) gives rise to formulas $i \leftrightarrow \bigwedge_{j\in E(i)} \neg j$ in the corresponding theory. Writing this equivalence in clausal form gives:

(a) one positive clause $i \vee \bigvee_{j\in E(i)} j$ and
(b) binary negative clauses $\neg i \vee \neg j$ for each $j \in E(i)$.

We use the even numbers to represent clauses under (a), and a subset of the odd numbers to represent clauses under (b). For the latter we use some familiar injective pairing function such as $f(i,j) = (i+j)(i+j+1) + 2i + 1$, which is surjective on the odd numbers. Now we define the ternary predicate C as follows, where C is taken to be false in all cases in which it is not defined to be true.

(a)	$C(2i, i, 1)$	for all $i \in \mathbb{N}$
	$C(2i, j, 1)$	for all $i \in \mathbb{N}$ and $j \in E(i)$
(b)	$C(f(i,j), i, 2)$	for all $i \in \mathbb{N}$ and $j \in E(i)$
	$C(f(i,j), j, 2)$	for all $i \in \mathbb{N}$ and $j \in E(i)$
	$C(f(i,j), 0, k)$	for all $i, j, k \in \mathbb{N}$ with $j \notin E(i)$

The last clauses are deliberately tautological, including $k = 1$ and $k = 2$. Without these, the theory would contain empty clauses $C_{f(i,j)}$ for $j \notin E(i)$, and would hence be unsatisfiable. Clearly we have that the clausal theory encoded by C is satisfiable if and only if $\mathsf{G} = \langle \mathbb{N}, E\rangle$ has a kernel. Also, C is recursive when E is, yielding GSOL $\leq_m$ ISAT. We thus obtain:

Corollary 4.3. *Consistency of PL^ω-theories,* ISAT, *is Σ^1_1-complete.*

5. Axiomatic strength

The next subsection contains the main result of the paper: equivalence, over RCA_0, of countable compactness and the solvability of finitely branching dags. Section 5.2 shows that ZF proves solvability of trees, while the solvability of forests can be proven in ZF with only a weak form of AC. It also gives a new equivalent of AC in terms of digraph solvability.[5]

[5] Not even full ZF is necessary, but we have refrained from analyzing this in detail.

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004

5.1. Solvability of fb dags over RCA_0

The transformations between digraphs and propositional theories from Section 2 suggest an analogy where ω-paths correspond to infinite theories while infinite branching corresponds to infinitary formulae. Infinitary propositional theories can be much more complex and expressive than infinite theories in finitary propositional logic. Consequently, one can expect that bounding the branching degree really makes solvability results less demanding from the axiomatic point of view. On the other hand, bounding the length of paths to be finite cannot be expected to simplify solvability results much in this respect.

As an important special case, solvability of trees, or more generally dags, without ω-paths (but with arbitrary branching), leaves much axiomatic strength intact. The result was given in Corollary 3.4 and Friedman states its equivalence over RCA_0 to ATR_0, [8]. On the other hand, it is not difficult to see that already RCA_0 proves solvability of every rooted tree with no finite path but with finite branching. (Assuming the tree is given by an adjacency list for every node, the distance to the root can be defined recursively, and nodes can be assigned the value given by the parity of this distance.) In this section we prove the solvability of finitely branching (fb) dags in the weakest possible subsystem of second-order arithmetic in which this result can be proved, namely, the system WKL_0. Recall from [15] that the weakest of these systems is RCA_0, in which only Δ^0_1-comprehension and Σ^0_1- and Π^0_1-induction are allowed (in addition to first-order arithmetic). The system WKL_0 extends RCA_0 by Weak König's Lemma. Since solvability of a finitely branching digraph G is equivalent to consistency of the corresponding propositional theory $\mathsf{T} = \mathcal{T}(\mathsf{G})$, we find it convenient to use an equivalent of WKL_0, namely the extension of RCA_0 by the axiom stating the compactness of countable propositional theories (see [15, Thm.IV.3.3]). We henceforth call the latter axiom *countable compactness*.

Our proof consists of two parts. The easy part is to show that countable compactness is sufficient to prove solvability of fb dags in RCA_0. This result is not new, but is not well-known and we have not found any reference. (E.g., [12] applies propositional compactness to obtain solvability of certain fb digraphs, but doesn't state the general result explicitly.) What is truly new here is the converse, that solvability of countable fb dags proves countable compactness and that this proof can be carried out in RCA_0. This is the hard part, which can be seen as a contribution to Friedman's programme of Reverse Mathematics. We start with the easy part.

Lemma 5.1. *Countable compactness implies, over* RCA_0*, solvability of fb dags (represented by adjacency lists).*

Proof. Let $\mathsf{G} = \langle \mathbb{N}, E\rangle$ be an fb dag. The corresponding propositional theory $\mathcal{T}(\mathsf{G})$ consists of all formulas $x \leftrightarrow \bigwedge_{y\in E(x)} \neg y$. If x is a sink then the above formula reads $x \leftrightarrow \mathbf{1}$, or simply x. At this point, care must be exercised when reasoning in RCA_0. First, the propositional formulas must be encoded as numbers. Second, the set of codes representing $\mathcal{T}(\mathsf{G})$ must be definable by Δ^0_1-comprehension. In order to achieve this we require that the graph G is given by a (neighbourhood) function $E : \mathbb{N} \to \mathbb{N}^*$, that is, as a function of nodes to finite sequences of nodes. These finite sequences are called *adjacency lists*. For a sink x the sequence $E(x)$ is empty. These adjacency lists make it possible to define in RCA_0 the theory $\mathcal{T}(\mathsf{G})$ as the set of all codes of formulas $x \leftrightarrow \bigwedge_{y\in E(x)} \neg y$. As noted after Proposition 3.3, every finite dag has a unique kernel. This fact can actually be proved with only finite combinatorics. Now, any finite subset S of $\mathcal{T}(\mathsf{G})$ can easily be strengthened by adding propositions y for all y occurring in S only on the righthand side of a formula in S. (The reason for doing this is that such y become sinks in the graph corresponding to S, and hence get assigned truth value $\mathbf{1}$.) Call the extended set of formulas S'. Taking G' to be the finite subgraph of G induced by the nodes/variables occurring in S, we then have $S' = \mathcal{T}(\mathsf{G}')$. The solution of G' is a model of S' by Eq. (2.3), and hence S has a model. It follows by countable compactness that $\mathcal{T}(\mathsf{G})$ has a model, which is a solution of G by, again, Eq. (2.3). □

For the difficult part, let $\Sigma = \{p_1, p_2, \ldots\}$ be a countable set of variables, $\mathcal{C}$ the set of all (finite) clauses over Σ without complementary pairs of literals. For any theory $T \subseteq \mathcal{C}$, we will define a graph G_T. These graphs G_T will be fb dags whose solutions represent models of T provided that every *finite* subtheory of T has a model. In this way we will prove that solvability of countable fb dags implies countable compactness.

Let a theory T be given by an enumeration $t_0, t_1, \ldots$ of its clauses. Let T_i denote the finite subtheory of T consisting of $t_0, t_1, \ldots, t_i$. For every $i \in \mathbb{N}$, let $\mathcal{C}_i \subset \mathcal{C}$ contain all clauses with maximal literal index i. Clauses are denoted as disjunctions of literals with increasing indices, but are actually finite sets of literals. This means that we may write, for example, $\neg p_1 \in C$ for a clause C. For every i, $\mathcal{C}_i$ is finite and we denote its cardinality by $|\mathcal{C}_i|$. For example, $\mathcal{C}_2$ consists of the six clauses $p_1 \vee p_2$, $p_1 \vee \neg p_2$, $\neg p_1 \vee p_2$, $\neg p_1 \vee \neg p_2$, p_2, $\neg p_2$. (Note that the enumerations of T and $\mathcal{C}$ may be totally unrelated, for example, both $t_0 = p_{99}$ and $t_{99} = p_1$ are possible.)

The set $\mathbb{N}\times\mathbb{N}$ is the set of nodes of the graph G_T. In order to be compatible with Lemma 5.1, the graph must be represented by an adjacency list for every node. We allow ourselves a graphical representation which is easier to grasp, and leave it to the reader to verify that the set of adjacency lists actually can be obtained by Δ^0_1-comprehension. The nodes at even levels $2i$ represent the literals, in such a way that for all k, $\langle 2i, 2k\rangle$ corresponds to p_i and $\langle 2i, 2k+1\rangle$ to $\neg p_i$. The odd levels $2i-1$ are used to represent clauses from $\mathcal{C}_i$. The level $2i-1$ is thought to be divided into intervals of length $|\mathcal{C}_i|$, with nodes $\langle 2i-1, s + |\mathcal{C}_i| * n\rangle$, for all $n \geq 0$ and $0 \leq s < |\mathcal{C}_i|$, representing the $(s+1)$-th clause $C^s_i \in \mathcal{C}_i$. The n in the second element of the pair determines whether this node has edges (in and possibly out), and this depends on whether C^s_i follows from T_n or not. With the exception of edges $\langle 2i, k\rangle \to \langle 2i, k+1\rangle$, there are only edges $\langle a_1, a_2\rangle \to \langle b_1, b_2\rangle$ with $b_i < a_i$ ($i = 1, 2$). This ensures both fb and the fact that adjacency lists can be computed from the edge relation that we will define now.

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004

Definition 5.2. Given the enumerated theory $T = \{t_0, t_1, t_2, \dots\}$, where no clause t_i contains a complementary pair, recall that $T_i = \{t_0, \dots, t_i\}$. The nodes of G_T are pairs of natural numbers:

for every $i > 0$, $p_i \in \Sigma : \{2i\} \times \mathbb{N}$, written p_i^k,
for every $i > 0 : \{2i-1\} \times \mathbb{N}$, written c_i^k.

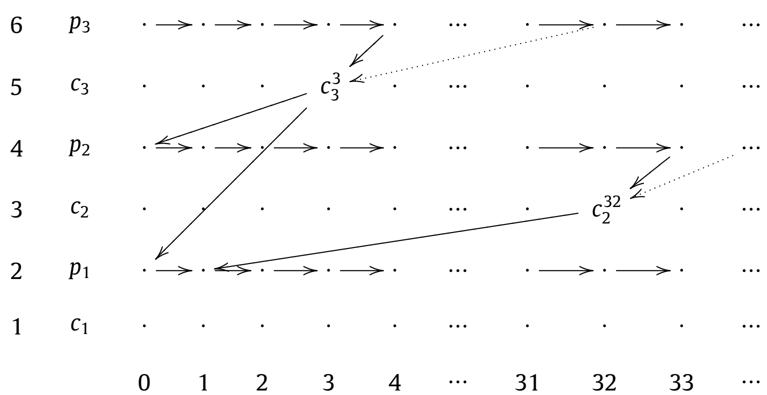
There is an edge $\langle a_1, a_2\rangle \to \langle b_1, b_2\rangle$ in G_T in each of the three cases:

1. $p_i^k \to p_i^{k+1}$, i.e., $a_1 = b_1 = 2i$, $a_2 = k$ and $b_2 = a_2 + 1$.

2. $p_i^k \to c_i^{s+|\mathcal{C}_i|*n}$, i.e., $a_1 = 2i = b_1 + 1$, $a_2 = k > b_2 = s + |\mathcal{C}_i| * n$, provided that $T_n \models C_i^s$ for clause $C_i^s \in \mathcal{C}_i$ $(0 \le s < |\mathcal{C}_i|)$. Moreover we require that $odd(k) \leftrightarrow p_i \in C_i^s$. The latter requirement means that if $\neg p_i \in C_i^s$, then the edges depart from p_i^k with k even, that is, departing from nodes representing p_i. If the other literals in C_i^s are false, then for C_i^s to be true p_i must be false, and this is exactly what this edge does. The case $p_i \in C_i^s$ and odd k is perfectly dual. This motivates the last part of the definition, where we define the outgoing edges of $c_i^{s+|\mathcal{C}_i|*n}$ to the remaining literals (if any) in C_i^s.

3. $c_i^{s+|\mathcal{C}_i|*n} \to p_j^k$, i.e., $a_1 = 2i-1$, $a_2 = s + |\mathcal{C}_i| * n$, $b_1 = 2j$, $1 \le j < i$, provided that $T_n \models C_i^s$ for a clause $C_i^s \in \mathcal{C}_i$ $(0 \le s < |\mathcal{C}_i|)$ and either $p_j \in C_i^s$ and $b_2 = 0$, or $\neg p_j \in C_i^s$ and $b_2 = 1$.

Example 5.3. Let $T_0 = \{C_3^3\}$, where $C_3^3 = p_1 \vee p_2 \vee \neg p_3$ is, say, the fourth clause in $\mathcal{C}_3$, and $T_5 \models \neg p_1 \vee p_2 = C_2^2 \in \mathcal{C}_2$.



For $C_3^3 \in \mathcal{C}_3$, we have $T_0 \models C_3^3$, so it appears for the first time in the "interval $n = 0$" of the c_3-level, at position $3 + |\mathcal{C}_i| * 0$, i.e., in the node c_3^3. Its maximal literal p_3 occurs negatively, so there are edges from all p_3^k with even $k > 3$ to c_3^3. Since p_1 and p_2 occur positively in C_3^3, we have edges $c_3^3 \to p_1^0$ and $c_3^3 \to p_2^0$.

Further, there are six clauses in $\mathcal{C}_2$, i.e., $|\mathcal{C}_2| = 6$, and we have assumed $T_5 \models C_2^2$ (and $T_4 \not\models C_2^2$). Hence $C_2^2 \in \mathcal{C}_2$ appears at position $2 + 6 * 5 = 32$ of the c_2 level. Since the maximal literal p_2 in C_2^2 occurs positively, there are edges from all p_2^k with odd $k > 32$ to c_2^{32}. Since C_2^2 contains $\neg p_1$, there is an edge from c_2^{32} to p_1^1.

The following theorem gives the desired connection between solutions of G_T and models of T. Given a binary function α on $\mathbb{N} \times \mathbb{N}$, we view α as a valuation on Σ by putting $\alpha(p_i) = \alpha(p_i^0)$ (recall that $p_i^0 = \langle 2i, 0\rangle$). We may then extend α to literals and clauses by putting $\alpha(\neg p_i) = \neg\alpha(p_i)$ and $\alpha(C) = \mathbf{1}$ iff $\alpha(l) = \mathbf{1}$ for some literal $l \in C$.

Theorem 5.4. *Let G_T be as in Definition 5.2 and let α be a solution of G_T. If T_n has a model for all n, then we have for all $j > 0$:*
(1) $\forall k \ge 0\ (\alpha(p_j^k) = \neg\alpha(p_j^{k+1}))$ and
(2) $\forall n \ge 0\ \forall i \le j\ \forall C \in \mathcal{C}_i\ [T_n \models C \Rightarrow \alpha(C) = \mathbf{1}]$.

Proof. Before we prove the theorem by induction on j, observe that (1) and (2) are Π_1^0-formulae, so that the induction can be carried out in RCA_0.

Base case $j = 1$. The nodes c_1^m are all sinks. If we do not have $T_n \models p_1$ or $T_n \models \neg p_1$ for some n, then there are no edges $p_1^k \to c_1^n$ and (1) and (2) hold trivially. Note that $\mathcal{C}_1$ consists of $p_1 = C_1^0$ and $\neg p_1 = C_1^1$. If $T_n \models p_1$ ($T_n \models \neg p_1$) then there are edges $p_1^k \to c_1^{2n}$ ($p_1^k \to c_1^{1+2n}$) for all odd (even) $k > 2n$ ($k > 1 + 2n$). Using $T_n \models p_1 \wedge \neg p_1$ for no n, one proves (1) and (2) by observing that the k above have the correct parity in the respective cases.

Step case $j+1$. Assume we have proved (1) and (2) for j and below. We first prove (1) by contradiction. If $\alpha(p_{j+1}^k) = \alpha(p_{j+1}^{k+1})$ for some k, then it must be that $\alpha(p_{j+1}^k) = \mathbf{0} = \alpha(p_{j+1}^{k+1})$ since α is a solution of the graph G_T in which $p_{j+1}^k \to p_{j+1}^{k+1}$ is an edge. Hence there must be $c_{j+1}^q = \mathbf{1}$ with an edge $p_{j+1}^k \to c_{j+1}^q$ for some q. By Definition 5.2.2 this is the case when $q = s + |\mathcal{C}_{j+1}| * n$, $T_n \models C_{j+1}^s \in \mathcal{C}_{j+1}$ for certain s, n. Moreover we have $k > |\mathcal{C}_{j+1}| * n + s$, with k odd iff p_{j+1} occurs positively in C_{j+1}^s. But then there is also an edge $p_{j+1}^{k+2} \to c_{j+1}^q$, so also $p_{j+1}^{k+2} = \mathbf{0}$. As a consequence there must also be a $c_{j+1}^r = \mathbf{1}$ with an edge $p^{k+1}_{j+1} \to c^r_{j+1}$ based on a clause $C^t_{j+1} \in \mathcal{C}_{j+1}$ with $T_m \models C^t_{j+1}$ for certain t, m. As k and $k+1$ have different parity, the literals in C^s_{j+1} and C^t_{j+1} with maximal index $j+1$ are complementary. As the situation is perfectly symmetric, we may assume without loss of generality that k is odd, that is, $p_{j+1} \in C^s_{j+1}$ and $\neg p_{j+1} \in C^t_{j+1}$, and that $n \geq m$. Then we have that $T_n \models R = (C^s_{j+1} - \{p_{j+1}\}) \cup (C^t_{j+1} - \{\neg p_{j+1}\})$, where the resolvent R consists entirely of (one or more) literals with index $\leq j$. It follows by the induction hypothesis (2) that $\alpha(R) = \mathbf{1}$. However, since $c^s_{j+1} = c^r_{j+1} = \mathbf{1}$ and α is a solution, all successors of these nodes are assigned value $\mathbf{0}$ by α. Since these successors represent the literals in R we get $\alpha(R) = \mathbf{0}$, which is a plain contradiction. This completes the proof of (1) in the induction case. For proving (2), assume that $T_n \models C$ for some clause C that consists entirely of literals with index $\leq j+1$. Without loss of generality we may assume that n is minimal. If the literal with highest index in C has index $\leq j$ we can apply the induction hypothesis (1). Otherwise, $C = C^s_{j+1} \in \mathcal{C}_{j+1}$ for suitable s. It follows that C is represented by the node c^m_{j+1} with $m = s + n * |\mathcal{C}_{j+1}|$ (and by such nodes with $n+1, n+2, \ldots$, but one suffices). Then we have an edge $p^k_{j+1} \to c^m_{j+1}$ for $k = m+1$ or $k = m+2$, as well as edges to the nodes representing the (zero or more) literals in C with index $\leq j$. (At this point it may be helpful to revisit Example 5.3 and to look at the nodes c^3_3 and c^{32}_2.) We are in a situation in which we have (1) for all levels up to and including level $j+1$. This means that all nodes p^k_i with k even have the value $\alpha(p_i)$, and those with k odd the value $\alpha(\neg p_i)$ $(1 \leq i \leq j+1)$. Now, by the definition of G_T and the assumption that α is a solution, we get that $\alpha(C) = \mathbf{1}$: if all literals in C with index $\leq j$ have value $\mathbf{0}$, then c^m_{j+1} has value $\mathbf{1}$ and hence p^k_{j+1} has value $\mathbf{0}$. By Definition 5.2 and (1), the latter node represents the *complement* of the literal with index $j+1$ in C, and hence $\alpha(C) = \mathbf{1}$. This completes the induction step. □

Theorem 5.5. *The solvability of countable fb dags (given by adjacency lists) is equivalent to* WKL *over* RCA_0.

Proof. Over RCA_0, countable compactness is equivalent to WKL [15, Thm.IV.3.3]. Theorem 5.4 above and its converse in Lemma 5.1 give the equivalence. □

5.2. *Choice principles and solvability over* ZF

We start by showing solvability of arbitrary trees in ZF. The proof suggests that solvability of forests may require the Axiom of Choice. Surprisingly, a very weak version – AC(2) or, for countable forests, van Douwen's Choice Principle – suffices. Finally, we give an equivalent of full AC over ZF, in terms of solvability of complete digraphs.

Recall some basic definitions. Given an indexed family of sets $X = \{X_i \mid i \in I\}$, its *disjoint union* is the set

$$\biguplus X = \biguplus_{i \in I} X_i := \left\{\langle i, x\rangle \mid i \in I,\ x \in X_i\right\},$$

while its *cartesian product* is the set

$$\prod X = \prod_{i \in I} X_i := \left\{f \subseteq \biguplus X \mid f \text{ is a function with domain } I\right\}.$$

Unless stated otherwise, we assume that all sets X_i are non-empty. Then, a *choice function* on a set X is any $f \in \prod X$. The *Axiom of Choice*, AC, is the statement: for every set X (with all $X_i \neq \varnothing$), there exists a choice function, i.e., $\prod X \neq \varnothing$. AC(2) states that a choice function exists for every set X with cardinality $|X_i| = 2$ for every $X_i \in X$.

For an indexed family of digraphs $\mathcal{G} = \{\mathsf{G}_i \mid i \in I\}$, with $\mathsf{G}_i = \langle G_i, E_i\rangle$ for all $i \in I$, its *disjoint union* is defined by $\biguplus_{i \in I} \mathsf{G}_i := \langle \biguplus_{i \in I} G_i, E\rangle$ with $E := \{\langle\langle i, v\rangle, \langle i, v'\rangle\rangle \mid i \in I, v \in G_i, v' \in E_i(v)\}$.

5.2.1. *Trees, forests and* AC(2).

An acyclic digraph is a *forest* if every node has at most one predecessor. Ancestors of every node in a forest are thus totally ordered by the transitive closure of the predecessor relation $E^{\smile}$. A *tree* is a forest where every two nodes have a common ancestor. A tree's (unique) source, if any, is called its *root*.

The construction from (3.1) and Proposition 3.3 can be carried out in ZF by transfinite recursion on ordinals with cardinality not exceeding the cardinality of the considered graph. This allows us to establish the following proposition. The proof uses the notion of a *tight* digraph morphism which not only preserves but also reflects the edge relation, i.e., a mapping of vertices $h : F \to G$ such that $h(E_{\mathsf{F}}(x)) = E_{\mathsf{G}}(h(x))$. A tight morphism reflects solutions: whenever $\alpha \in sol(\mathsf{G})$, then $\alpha \circ h \in sol(\mathsf{F})$, where, for all $x \in F : (\alpha \circ h)(x) = \alpha(h(x))$.

Proposition 5.6. ZF $\vdash$ *every tree is solvable.*

Proof. Given a tree $\mathsf{T} = \langle T, E\rangle$, define σ as in (3.1). If the resulting σ leaves a non-empty $T^\circ \subseteq T$ unassigned, Proposition 3.3 ensures that T° has no sinks. (For any $X \subseteq T$, X denotes the subgraph of T induced by X.) It is possible that T° is a forest but not a tree. We argue that all trees in the forest T°, with at most one exception, U, are rooted. Let $R = \{r \in T^\circ \mid \neg\exists x \in T^\circ\ r \in E(x)\}$ be the set of sources (roots of trees) in T°. For each $r \in R$, $\mathsf{T}^\circ_r = E^*_{\mathsf{T}^\circ}(r)$ is a tree with root r. The trees T°_r, $r \in R$, are mutually disjoint (each containing all T° descendants of its root r). It is possible that $U = T^\circ \setminus \bigcup_{r \in R} T^\circ_r$ is not empty, but then U is a tree: every pair $x, y \in U$ has a common ancestor in T which is in U as well (since every node in U has a predecessor in U, it has the same ancestors in U as in T.) Also, all trees are sinkless, since T° is sinkless.

Define an equivalence on the nodes T° by

$$x \sim y \Leftrightarrow \exists v \in T^\circ \exists n \in \mathbb{N}\, \{x, y\} \subseteq E^n_{\mathsf{T}^\circ}(v), \tag{5.7}$$

i.e., if the nodes are at the same distance n from some common ancestor v in some tree of the forest T°. (Transitivity of $\sim$ follows since ancestors of every node are totally ordered.) The quotient $\mathsf{Q} = \mathsf{T}^\circ/_\sim = \langle T^\circ/_\sim, E_1\rangle$, where $E_1([x]) = \{[y] \mid y \in E(x)\}$, is then a collection of disjoint digraphs, each isomorphic to $\mathsf{N} = \langle \mathbb{N}, Succ\rangle$, with the possible exception of one integer graph (the quotient of the rootless tree U). Each digraph isomorphic to N is solvable by assigning $\alpha([r]) = \mathbf{1}$ to its root and propagating the values $\alpha(Succ([x])) = \neg\alpha([x])$. The integer graph is solvable by choosing its arbitrary element $[u]$, assigning $\alpha([u]) = \mathbf{1}$ and propagating the values "downwards", $\alpha(Succ([x])) = \neg\alpha([x])$, and "upwards", $\alpha(Succ^{\smile}([x])) = \neg\alpha([x])$, starting with $[x] = [u]$. Thus, Q is solvable.

The quotient mapping $\mathsf{T}^\circ \rightarrow \mathsf{Q}$, sending x onto $[x]$, is a tight morphism, so any solution for Q can be reflected into a solution for T° which, combined with σ, gives a solution to T by Proposition 3.3. □

If all but finitely many trees in a forest are rooted or have no ω-paths, the proof can be still done in ZF (using the roots, when needed). But for unrooted, sinkless trees, the proof relies on the choice of an u, suggesting the plausible conjecture that solvability of arbitrary forests requires some form of AC. The following lemma allows to show that full AC is not needed, and that a very weak form suffices.

Proposition 5.8. ZF $\vdash$ *every forest is solvable iff every collection of disjoint integer graphs is solvable.*

Proof. Implication to the right is obvious since every disjoint union of integer graphs is a forest. For the opposite, as noted above, we only have to show the claim for a forest consisting of unrooted trees without sinks. Given such a forest $\mathsf{F} = \langle F, E\rangle$, define an equivalence relation on its nodes by (5.7). The quotient $\mathsf{Z} = \mathsf{F}/_\sim = \langle F/_\sim, E_1\rangle$, where $E_1([x]) = \{[y] \mid y \in E(x)\}$, is then a collection of disjoint integer graphs. The quotient mapping $\mathsf{F} \rightarrow \mathsf{Z}$ is a tight morphism so any solution for Z can be reflected into a solution for F. □

Solvability of forests follows now in ZF extended with AC(2). Define an equivalence on $Z = F/_\sim$ by $z \simeq u \Leftrightarrow \exists n \in \mathbb{N}\, [z \in E_1^{2n}(u) \vee u \in E_1^{2n}(z)]$, i.e., if the nodes are at an even distance from each other. The quotient $\mathsf{L} = \mathsf{Z}/_\simeq = \langle Z/_\simeq, E_2\rangle$, where $E_2([z]^\simeq) = \{[y]^\simeq \mid y \in E_1(z)\}$ is isomorphic to a disjoint collection of graphs $\bullet \leftrightarrows \bullet$. Solvability of such collections is equivalent to AC(2). The quotient mapping $\mathsf{Z} \rightarrow \mathsf{L}$ is a tight morphism, so a solution for L, existing by AC(2), can be reflected into a solution for Z, and then for F.

For countable forests, another version of choice suffices. Van Douwen's Choice Principle, vDCP (FORM 119 in [10]), is the assertion that a countable family of non-empty disjoint sets $X = \{X_i \mid i \in \omega\}$, for which there is a function f such that for every $i \in \omega : \langle X_i, f(i)\rangle$ is an integer graph, has a choice function. Having a choice function for every such X, allows to solve any countable collection of disjoint integer graphs which, by Lemma 5.8, implies solvability of any countable forest. Over ZF, vDCP does not imply the axiom of choice for countable collections of sets with two elements, AC($\aleph_0$, 2), [10]. Consequently, since solvability of countable forests is implied by vDCP, it does not imply AC($\aleph_0$, 2).

Note that solvability of disjoint integer digraphs amounts to a specific partition principle, splitting each $\mathbb{Z}$-isomorphic subset into its "odd" and "even" vertices. This appears significantly weaker than vDCP, and we conjecture that ZF does not prove vDCP from forest solvability.

5.2.2. *Complete digraphs and full* AC.

To formulate an equivalent of AC in terms of digraph solvability, we will use a specific property of solutions of weakly complete graphs, which we describe first.

We call a digraph G *strongly complete* if $E = \{\langle x, y\rangle \mid x, y \in G, x \neq y\}$, i.e., if each pair of distinct vertices is connected by two directed edges, each in one direction. We call it *weakly complete* if for each pair of distinct vertices $x \neq y$ either $\langle x, y\rangle \in E$ or $\langle y, x\rangle \in E$. (The latter allows loops, the former does not.) Equivalently, digraph G is weakly complete if its underlying, undirected graph $\underline{\mathsf{G}}$ is *complete*, namely, $\underline{E} \supseteq \{\{x, y\} \mid x, y \in G, x \neq y\}$.

For a kernel $\alpha \in sol(\mathsf{G})$, $\alpha^{\mathbf{1}}$ must be independent and dominating in the underlying graph $\underline{\mathsf{G}}$. These two properties are equivalent to $\alpha^{\mathbf{1}}$ being a maximal independent subset of $\underline{\mathsf{G}}$. A simple fact follows from this observation.

Proposition 5.9. *For a weakly complete digraph* G*, the following holds:*

$$\alpha \in sol(\mathsf{G}) \Longleftrightarrow \exists x \in G\, [\alpha^{\mathbf{1}} = \{x\} \wedge E^{\smile}(x) = G \setminus \{x\}].$$

Proof. The implication to the left holds for any digraph: if x satisfies $\{x\} = G \setminus E^{\smile}(x)$, then $\{x\}$ is a kernel. Conversely, if $\alpha \in sol(\mathsf{G})$ then, since $\alpha^{\mathbf{1}}$ must be a maximal independent subset of the complete $\underline{\mathsf{G}}$, $|\alpha^{\mathbf{1}}| = 1$. So assume a solution with $\alpha^{\mathbf{1}} = \{x\}$ for some x. Then $x \notin E^{\smile}(x)$ and if $y \neq x$, then $\alpha(y) = \mathbf{0}$ and so $y \in E^{\smile}(x)$, i.e., $E^{\smile}(x) = G \setminus \{x\}$. □

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004

Example 5.10. The digraph C_3 is a cycle with three vertices. C_3 is weakly complete and unsolvable, having no vertex x as required by the proposition. But adding a single reverse edge makes it solvable.

Every strongly complete digraph G is solvable: every solution α of G picks a vertex u making $\alpha(u) = \mathbf{1}$ and $\alpha(v) = \mathbf{0}$ for all $v \neq u$, Hence strongly complete digraphs have precisely as many solutions as vertices.

The Yablo dag, $\langle \mathbb{N}, < \rangle$, is unsolvable: it is weakly complete, but does not contain a vertex x as required by the proposition. The argument applies unchanged to generalizations of the Yablo dag to any total ordering without greatest element, e.g., dags over rationals or reals, $\langle \mathbb{Q}, < \rangle$ or $\langle \mathbb{R}, < \rangle$, are unsolvable.

Theorem 5.11. *Over* ZF, AC *is equivalent with the following statement:*

GS*: For every indexed family* $\{\mathsf{G}_i \mid i \in I\}$ *of solvable digraphs, the disjoint union* $\biguplus_{i\in I} \mathsf{G}_i$ *is solvable.*

Proof. AC $\Rightarrow$ GS). Let $\{\mathsf{G}_i = \langle G_i, E_i \rangle \mid i \in I\}$ be an indexed family of solvable digraphs. By AC it follows that the product $\prod_{i\in I} sol(\mathsf{G}_i)$ is non-empty. But every $f \in \prod_{i\in I} sol(\mathsf{G}_i)$ defines a solution $\alpha_f : \biguplus_{i\in I} G_i \to \{\mathbf{0}, \mathbf{1}\}$ of $\biguplus_{i\in I} \mathsf{G}_i$ by letting $\alpha_f(\langle i, v \rangle) := f(i)(v)$ for all $\langle i, v \rangle \in \biguplus_{i\in I} G_i$.

GS $\Rightarrow$ AC). Let $\{X_i : i \in I\}$ be a collection of non-empty sets. For each $i \in I$, let G_i be the strongly complete digraph with X_i as its set of vertices. By Proposition 5.9 G_i has $|X_i|$ solutions, each picking one element of X_i. For every solution α of the disjoint union $\biguplus_{i\in I} \mathsf{G}_i$ it holds that the restriction $\alpha|_{G_i}$ of α to G_i is a solution of G_i. Consequently, every solution α of $\biguplus_{i\in I} \mathsf{G}_i$ induces a function $f : I \to \bigcup_{i\in I} X_i$ in $\prod_{i\in I} X_i$ by defining, for every $i \in I$, $f(i)$ as the $x \in X_i$ such that $\alpha|_{G_i}(x) = \mathbf{1}$ – which x is unique by Proposition 5.9. □

By employing dag-lifting and Theorem 3.9, this result extends to dags.

Corollary 5.12. *Over* ZF, AC *is equivalent with the following statement:*

DS*: For every indexed family* $\{\mathsf{D}_i \mid i \in I\}$ *of solvable dags, the disjoint union* $\biguplus_{i\in I} \mathsf{D}_i$ *is solvable.*

Proof. Since GS $\Rightarrow$ DS is obvious, in view of Theorem 5.11 it suffices to show DS $\Rightarrow$ GS. For this, assume DS and let $\{\mathsf{G}_i \mid i \in I\}$ be an indexed family of solvable digraphs. By Theorem 3.9, the dag-lifting G_i^ω of G_i is solvable for every $i \in I$. Then it follows by DS that the dag $\mathsf{D} := \biguplus_{i\in I} \mathsf{G}_i^\omega$ is solvable. Since, as is easy to prove in ZF, D is isomorphic to G^ω for $\mathsf{G} := \biguplus_{i\in I} \mathsf{G}_i$, it follows that G^ω is solvable, and hence, by Theorem 3.9 again, that G is solvable. □

6. Conclusion

Kernel theory is an active research field of graph theory; a recent overview can be found in [2]. Unlike most of the research in kernel theory, we have studied graph kernels from the point of view of mathematical logic. We have elaborated constructions for the following:

1. For every digraph G a (possibly infinitary) propositional theory $\mathcal{T}(\mathsf{G})$, the model class of which corresponds to the set of kernels of G.
2. For every propositional theory T a digraph $\mathcal{G}(\mathsf{T})$ the set of kernels of which corresponds to model class of T.
3. For every digraph an infinite dag having essentially the same kernels.
4. For every binary relation R a digraph which has a kernel if and only if R is not well-founded.

All constructions preserve recursiveness. These constructions yield, among other results, the following insights, of which only the first has been noticed before:

1. Propositional SAT and the existence of kernels of finitely branching digraphs are equivalent problems. In the finite case, both are NP-complete.
2. The problem of the existence of a kernel of recursive digraph is Σ^1_1-complete.
3. Since SAT of recursive theories in infinitary logic is equivalent to the existence of kernels of recursive, infinitely branching digraphs, this version of SAT is Σ^1_1-complete, too.
4. The problem of the existence of a kernel is equally difficult for (recursive) dags and for (recursive) digraphs.
5. The existence of kernels of finitely branching dags is equivalent over RCA_0 to countable compactness.
6. The existence of kernels for trees is provable in ZF. The existence of kernels for arbitrary forests requires at most AC(2), while for countable forests also vDCP suffices.
7. The existence of kernels for disjoint unions of digraphs (or respectively, of dags) that have kernels is equivalent over ZF to AC.

Acknowledgements

We thank Dag Normann, Sjur Dyrkolbotn, Vincent van Oostrom and Albert Visser for comments and discussion of various issues related to the results presented. An anonymous referee conjectures that, when fb dags are represented by edge sets instead of adjacency lists, an analogue of Theorem 5.5 can be obtained by replacing WKL_0 by ACA_0. This would imply that Theorem 5.5 really depends on the representation of fb dags. We are grateful for this suggestion for future research.

References

[1] Martine Anciaux-Mendeleer, Pierre Hansen, On kernels in strongly connected graphs, Networks 7 (3) (1977) 263–266.
[2] Endre Boros, Vladimir Gurvich, Perfect graphs, kernels and cooperative games, Discrete Mathematics 306 (2006) 2336–2354.
[3] Vašek Chvátal, On the computational complexity of finding a kernel, Technical Report CRM-300, Centre de Recherches Mathématiques, Université de Montréal, 1973. http://users.encs.concordia.ca/~chvatal.
[4] Roy Cook, Patterns of paradox, The Journal of Symbolic Logic 69 (3) (2004) 767–774.
[5] Nadia Creignou, The class of problems that are linearly equivalent to Satisfiability or a uniform method for proving NP-completeness, Theoretical Computer Science 145 (1995) 111–145.
[6] Pierre Duchet, Graphes noyau-parfaits, II, Annals of Discrete Mathematics 9 (1980) 93–101.
[7] Pierre Duchet, Henry Meyniel, Une généralisation du theorème de Richardson sur l'existence de noyaux dans les graphes orientés, Discrete Mathematics 43 (1) (1983) 21–27.
[8] Harvey Friedman, Kernel tower theory, I. FOM 407 (email list); http://cs.nyu.edu/pipermail/fom/2010-March/014507.html, 2010.
[9] Hortensia Galeana-Sánchez, Victor Neumann-Lara, On kernels and semikernels of digraphs, Discrete Mathematics 48 (1) (1984) 67–76.
[10] Paul Howard, Jean E. Rubin, Consequences of the axiom of choice, American Mathematical Society, Providence, R.I, 1998.
[11] John R. Isbell, On a theorem of Richardson, Proceedings of the AMS 8 (5) (1957) 928–929.
[12] Eric C. Milner, Robert E. Woodrow, On directed graphs with an independent covering set, Graphs and Combinatorics 5 (1989) 363–369.
[13] Moses Richardson, Solutions of irreflexive relations, The Annals of Mathematics, Second Series 58 (3) (1953) 573–590.
[14] Hartley J. Rogers, Theory of Recursive Functions and Effective Computability, MacGraw–Hill, 1967.
[15] Stephen G. Simpson, Subsystems of Second Order Arithmetic, in: Perspectives in Logic, 2nd edition, Cambridge University Press, 2009.
[16] John von Neumann, Oskar Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, 1944 (1947).
[17] Stephen Yablo, Paradox without self-reference, Analysis 53 (4) (1993) 251–252.

Please cite this article in press as: M. Bezem, et al., Expressive power of digraph solvability, Annals of Pure and Applied Logic (2011), doi:10.1016/j.apal.2011.08.004