Expressive Power of Digraph Solvability

Michał Walicki[1,*], Clemens Grabmayer[2], and Marc Bezem[1]

[1] University of Bergen, Department of Informatics,
P.O. Box 7803, 5020 Bergen, Norway
([*] on leave at UTL, IST, Lisbon)
michal,bezem@ii.uib.no
[2] Universiteit Utrecht, Department of Philosophy,
Heidelberglaan 6, 3584 CS Utrecht, The Netherlands
clemens@phil.uu.nl

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. We show that 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. Furthermore, we show that the existence of a kernel for a digraph is equivalent to the existence of a kernel for its lifting to an infinitely-branching dag. 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: we prove that solvability for recursive dags and digraphs is Σ^1_1-complete. This implies that satisfiability for recursive theories of infinitary propositional logic is also Σ^1_1-complete. Finally, using solvability of dags we formulate a new equivalent of the axiom of choice.

1 Introduction

In a digraph (directed graph), a subset X of its vertices is called 'independent' if the predecessors 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 famous book *Theory of Games and Economic Behavior* [14]. In order to explain the connection with game theory, consider a two-player game with alternating moves, let the vertices of a digraph G represent the positions and the edges the possible moves. Then kernels of G describe 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 that draws inspirations from, and finds applications in, game theory as well as in logic. Its main question concerns sufficient conditions for the existence of kernels in finite digraphs, e.g., [1, 7–9], with a recent overview in [2].

In this paper we aim at more general formulations, asking about the set-theoretical status of the solvability of the main classes of digraphs. We recall, from [4, 6], the equivalence between the existence of kernels of digraphs and the satisfiability of corresponding theories of propositional logic, and give a new, simple proof. This correspondence relates infinitely branching digraphs to infinitary propositional logic, and finitely branching ones to the usual propositional logic.

Overview. In Section 2 we start with the definitions of kernels and solutions of digraphs, and introduce correspondence functions between digraphs and propositional theories that map satisfiable theories to solvable digraphs and vice versa. In Section 3 we establish some independent results, useful for a general study of the digraph solvability, which are then used in Section 4, presenting the main results of the paper: Σ^1_1-completeness of the solvability of recursive dags, as well as of satisfiability of recursive theories in infinitary propositional logic, and a new equivalent of the axiom of choice, in terms of the solvability of certain digraphs.

2 Basic definitions and facts

A *directed graph* (a *digraph*) is a pair $\mathsf{G} = \langle G, E \rangle$, where G is a set of vertices and $E \subseteq G \times G$ is a binary relation representing the directed edges of G.[3] 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* in G is a vertex $x \in G$ without successors. Then $\mathit{sinks}(\mathsf{G}) = \{x \in G \mid E(x) = \varnothing\}$ denotes the set of sinks of 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) $G \setminus K \subseteq E^{\smile}(K)$ (from every non-kernel vertex there is at least one edge to a kernel vertex).

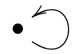
The equivalence between the existence of kernels and the satisfiability of propositional theories that we explore in this paper arises from an equivalent definition

[3] Directed graphs as defined here correspond to 'simple' ones according to a more general notion of 'directed graph' in which some vertices u and v may be connected by more than one edge with source u and target v; 'simple' directed graphs do not contain such *multiple edges*.

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:

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

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 $\bigwedge \varnothing = \mathbf{1}$.[4] Letting $mod(\mathsf{T})$ denote all models of a theory T, it is easy to see that (2.1) entails:

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

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 (and hence digraph solvability can be reduced to consistency of propositional theories). 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 formed over an arbitrary set of propositional variables by unary negation, and infinite conjunction[5]. PL^ω denotes the restriction of PL^∞ to propositions over a countable set of propositional variables and with conjunction of arity ω.

Conversely, also consistency of propositional theories can be reduced to solvability of corresponding digraphs. In fact, every PL^∞-theory T can be represented as a digraph $\mathcal{G}(\mathsf{T})$ such that $mod(\mathsf{T}) = sol(\mathcal{G}(\mathsf{T}))$. First, assume a theory T given as a set of equivalences of the 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

[4] Satisfiability of such a theory is equivalent to the existence of solutions for the corresponding system of boolean equations. This motivates the use of the name "solution". This was also the name used for kernels in the early days of kernel theory, e.g., in [14], p.588, or [12].

[5] By de Morgan, infinite disjunction can be used instead of infinite conjunction.

T which would become a sink of $\mathcal{G}(\mathsf{T})$, and hence could only be assigned $\mathbf{1}$ by any solution of $\mathcal{G}(\mathsf{T})$, can actually be also assigned $\mathbf{0}$ (when the respective $\overline{z}$ is assigned $\mathbf{1}$). 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

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

Now, an arbitrary theory T can be transformed to the above form. It can be done in many ways, and we give only one example, assuming T to be given as a set of clauses. A clause is a set of literals. A literal is an atom or a negated atom. Logically, a clause is the disjunction of its literals, with the convention that $\bigvee \varnothing = \mathbf{0}$.

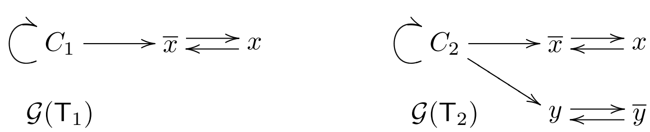
Given a clause C, we denote the subset of its positive (negative) literals by C^+ (C^-), and define the set P (the set N) as the index set of the positive (negative) literals in C. In other words, $C^+ = \{x_p \mid p \in P\}$ and $C^- = \{\neg x_n \mid n \in N\}$. First, let a_C be a new variable. The formula $C' : a_C \leftrightarrow \neg a_C \wedge \neg C$ is equivalent to $\neg a_C \wedge C$, and C and C' are equisatisfiable with $mod(C') = mod(C) \cup \{\langle a_C, \mathbf{0}\rangle\}$. Substituting for $\neg C$, we obtain a more explicit form of $C' : a_C \leftrightarrow \neg a_C \wedge \bigwedge_{p\in P} \neg x_p \wedge \bigwedge_{n\in N} x_n$. In addition, for every variable in the intial theory $x \in V(\mathsf{T})$, we introduce a new variable $\overline{x}$. For every such pair we introduce the two formulae (i), and for every clause C the formula (ii):

(i) $x \leftrightarrow \neg\overline{x}$ and $\overline{x} \leftrightarrow \neg x$.
(ii) $a_C \leftrightarrow \neg a_C \wedge \bigwedge_{p\in P} \neg x_p \wedge \bigwedge_{n\in N} \neg \overline{x}_n$

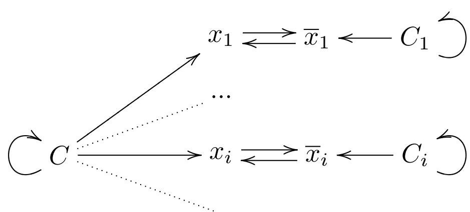
The theory C'' containing formulae (i) and (ii) is equisatisfiable to C, with $mod(C) = mod(C'')|_{V(C)}$. Defining $\mathsf{T}' = \bigcup_{C\in\mathsf{T}} C''$ and letting $\mathcal{G}(\mathsf{T}) = \mathcal{G}(\mathsf{T}')$, the equality (2.5) remains valid.

When no confusion can arise, we use the denotation of the clause C itself for the node representing the new propositional variable a_C.

Example 2.1. For $\mathsf{T}_1 = \{\neg x\}$ and $\mathsf{T}_2 = \{\neg x \vee y\}$, we obtain the digraphs:



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



$\mathcal{G}(\mathsf{T}_3)$ can be obtained from the finite (sub)graph G_1 induced by $\{C, x_1, \overline{x}_1, C_1\}$ by replicating the subgraph induced by these vertices except C. The inconsistency of T_3 is reflected by the lack of solutions for $\mathcal{G}(\mathsf{T}_3)$ which, in turn, reduces to the lack of solutions of the finite subgraph G_1. This suggests a possibility of reducing satisfiability of theories in PL$^\infty$ to solvability of finite graphs, instead of to satisfiability of finite subtheories. Such an investigation, however, is not the topic of the present paper.

The equation (2.3) for digraphs, and equation (2.5), for propositional theories establish a back-and-forth correspondence between satisfiable propositional theories and solvable digraphs that enables the transfer of results from propositional logic to digraphs and vice versa. For instance, that kernel existence of finite graphs is NP-complete [4] follows now from the fact that $\mathcal{G}$ defines a polynomial reduction. (As an aside: also $\mathcal{T}$ defines a polynomial reduction.) Conversely, for every theory T in our normal form, i.e., with all formulae in the form (2.4) and such that $\mathsf{T} = \mathcal{T}(\mathcal{G}(\mathsf{T}))$, we obtain the following equivalent of satisfiability in terms of kernels. Each $x \in V(\mathsf{T})$ occurs on the left of one $x \leftrightarrow E^{\neg}(x)$, and $V(E^{\neg}(x))$ denotes the variables in the formula $E^{\neg}(x)$. We have that

$$\begin{aligned} mod(\mathsf{T}) \neq \varnothing \overset{(2.5)(2.2)}{\iff} & \exists K \subseteq V(\mathsf{T}): \\ & K \cap \bigcup_{x \in K} V(E^{\neg}(x)) = \varnothing \wedge \forall y \notin K : V(E^{\neg}(y)) \cap K \neq \varnothing\,. \end{aligned} \tag{2.6}$$

The first conjunct reflects (i) and the second one (ii) of the definition of a kernel. Every model of T is uniquely determined by such a subset K, giving the variables to be assigned **1**. Thus, various statements of sufficient conditions for the existence of kernels, e.g., [1, 2, 7–9], can be now applied for determining satisfiability of PL theories.

3 Some general facts about solvability

This section presents some results on solvability that are of independent, general interest; some of them will play a role in the next section as well. We show in Section 3.1 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 has a unique solution, since the appropriate sinkless subgraph of a finite dag is empty. In Section 3.2 we give a simple characterization of the solutions of weakly complete digraphs, and in 3.3 we show that solutions for arbitrary digraphs can be represented as solutions for appropriate, infinitely branching dags.

3.1 Induced assignment

In this subsection we will use induction on the set of ordinals with cardinality at most the cardinality of the graph in question. All quantifications etc. are relative to this set of ordinals and we use κ to denote such ordinals (λ for limits). We

$$C_0 = G, \quad \text{for the given digraph } \mathsf{G} = \langle G, E\rangle$$
$$\sigma^{\mathbf{1}}_\kappa = sinks(\mathsf{C}_\kappa)$$
$$\sigma^{\mathbf{0}}_\kappa = E^{\smile}(\sigma^{\mathbf{1}}_\kappa) \cap C_\kappa$$
$$C_{\kappa+1} = C_\kappa \setminus (\sigma^{\mathbf{1}}_\kappa \cup \sigma^{\mathbf{0}}_\kappa) \quad \text{and} \quad C_\lambda = \bigcap_{\kappa<\lambda} C_\kappa \text{ for limit } \lambda$$
$$\mathsf{C}_\kappa \text{ is the subgraph induced by } C_\kappa \text{ for } \kappa > 0$$

$$G^\circ = \bigcap_\kappa C_\kappa \text{ and } \mathsf{G}^\circ \text{ is the induced subgraph}$$
$$\sigma^{\mathbf{v}} = \bigcup_\kappa \sigma^{\mathbf{v}}_\kappa, \text{ for } \mathbf{v} \in \{\mathbf{0}, \mathbf{1}\}$$
$$\text{The induced assignment is given by } \sigma = \{\langle x, \mathbf{v}\rangle \mid x \in \sigma^{\mathbf{v}}\}.$$

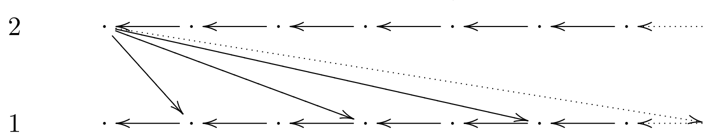
Fig. 1: Definition by ordinal induction of an induced sinkless subgraph G°, for every given digraph G.

will inductively remove 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 observed already in [10] 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 induction in Figure 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 3.1. In the digraph below, after ω iterations only vertices at level 1 obtain induced values. The digraph obtains the induced (unique) solution after $\omega + \omega$ iterations. Note that G° is empty here.



The following proposition allows one to consider sinkless digraphs only.

Proposition 3.2. *For any* G*, with* σ*,* C_κ *and* G° *as defined in Figure 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,* $sol(\mathsf{G}) \neq \varnothing \Leftrightarrow sol(\mathsf{G}^\circ) \neq \varnothing$*.*

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 $\{\kappa : |\kappa| \leq |G|\} \to G$, 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 $\kappa : \sigma^{\mathbf{1}}_\kappa \subseteq \alpha^{\mathbf{1}}$ 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 $\alpha(x) = \mathbf{0}$. This proves that 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 [14]. The above formulation implies also the existence of a unique solution, for instance, for arbitrary well-founded dags (i.e., having no infinite path of connected edges $\langle x_0, x_1\rangle, \langle x_1, x_2\rangle, \langle x_2, x_3\rangle, ...$).

3.2 Weakly complete digraphs

An *undirected graph* (or, shorter, a *graph*) is a pair $\mathsf{G} = \langle G, E\rangle$, where G is a set of vertices and E is a set of *undirected edges* between the vertices in G, that is, a set $E \subseteq \{\{u, v\} : u, v \in G\}$ consisting of unordered pairs of vertices in G. A graph $\mathsf{G} = \langle G, E\rangle$ is called *complete* if $E \supseteq \{\{u, v\} : u, v \in G, u \neq v\}$, that is, if every pair of distinct vertices of G is connected by an edge. Note that a complete undirected graph may, but need not, contain *self-loops*, that is, undirected edges that connect vertices with themselves.

Properties of undirected graphs can be used to define ‘weak’ counterparts for directed graphs by using the ‘forgetful projection’ from digraphs to graphs in which the direction of the edges is removed. Given a digraph G, let $\underline{\mathsf{G}}$ denote the undirected graph obtained by turning every directed edge $\langle x, y\rangle$ into an undirected one $\{x, y\}$. For every property P of undirected graphs, we say that G is *weakly* P if $\underline{\mathsf{G}}$ is P. For example, G is *weakly* connected, complete, etc., if $\underline{\mathsf{G}}$ is connected, complete, etc.

Note that distinct vertices u and v in a weakly complete digraph are connected by at least one of the directed edges $\langle u, v\rangle$ and $\langle v, u\rangle$. In addition to weak completeness, there is also the following stronger notion of completeness: a digraph $\mathsf{G} = \langle G, E\rangle$ is called *complete* if $E = \{\langle u, v\rangle : u, v \in G, u \neq v\}$, that is, if the edges of G connect each pair of distinct vertices of G in both directions. Note that a complete digraph does not contain any *self-loop*, that is, a directed edge that connects a vertex with itself.

For a kernel $\alpha \in sol(\mathsf{G}), \alpha^{\mathbf{1}}$ must be independent and dominating in $\underline{\mathsf{G}}$. These two properties are equivalent to $\alpha^{\mathbf{1}}$ being a maximal independent subset of $\underline{\mathsf{G}}$. Conversely, given such a maximal independent subset $K \subseteq \underline{\mathsf{G}}$, we determine if

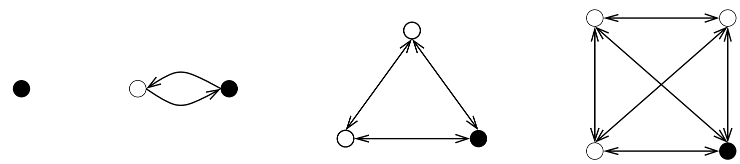


Fig. 2: Solutions of complete digraphs assign **1** to precisely one of the vertices.

it is a kernel of G by verifying that every vertex $x \in G \setminus K$ has a directed edge into K (an edge in $\underline{\mathsf{G}}$ might be only to x.)

Consequently, a trivial (and inefficient) algorithm for finding a kernel is to unorient the digraph G, find $\underline{\mathsf{G}}$'s maximal independent subsets, and for each such check if every vertex outside it has a directed edge to the subset. A simple fact follows from this observation.

Proposition 3.3. *For a weakly complete digraph* G*, the following hold:*

1. *for any* $\alpha \in sol(\mathsf{G}) : |\alpha^{\mathbf{1}}| = 1$.
2. $sol(\mathsf{G}) \neq \varnothing \;\Leftrightarrow\; \exists x \in G : E^{\smile}(x) = G \setminus \{x\}$
   (each such x *gives a distinct solution).*

Proof. Part 1 follows directly from the preceding observations: $\alpha^{\mathbf{1}}$ must be a maximal independent subset of a complete $\underline{\mathsf{G}}$. As to part 2, the implication to the left holds for any digraph since, if such an x exists, it satisfies $\{x\} = G \setminus E^{\smile}(x)$, i.e., is a kernel. Conversely, if $\alpha \in sol(\mathsf{G})$ then, by 1, $|\alpha^{\mathbf{1}}| = 1$. So assume a solution with $\alpha(x) = \mathbf{1}$ for some x. Then $x \notin E^{\smile}(x)$. Also, if $y \neq x$, then $\alpha(y) = \mathbf{0}$ and so $y \in E^{\smile}(x)$. □

Example 3.4. The digraph C_3, a cycle with 3 vertices, which is weakly complete, is unsolvable due to item 2 of the proposition. But, according to the same item, adding a single reverse edge makes it solvable.

Every complete digraph G is solvable: every solution α of G picks a vertex u such that $\alpha(u) = 1$ and $\alpha(v) = 0$ for all $v \neq u$, as illustrated by the examples in Figure 2. Hence complete digraphs have precisely as many solutions as vertices.

The infinite Yablo dag [15], the digraph $\langle \mathsf{N}, < \rangle$, is unsolvable: it is weakly complete, but it does not contain a vertex as required in item 2 of the proposition.

3.3 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:

$$\begin{aligned} G^{\omega} &:= G \times \omega \\ E^{\omega} &:= \{\langle n_i, m_j \rangle \mid \langle n, m \rangle \in E \wedge i < j\} \end{aligned} \tag{3.7}$$

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 in case that $i < j$. Also, $sinks(\mathsf{G}^\omega) = sinks(\mathsf{G}) \times \omega$ and G is not well-founded (has a cycle or an ω-path) iff G^ω has an ω-path.

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

$$f^\omega(n_i) := f(n) \qquad (\text{for all } n \in G \text{ and } i \in \omega)\,. \tag{3.8}$$

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

Lemma 3.6. *For every* $\mathsf{G} : (sol(\mathsf{G}))^\omega \subseteq sol(\mathsf{G}^\omega)$.

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

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

Let $\alpha \in sol(\mathsf{G})$, then $\alpha(x) = \alpha(E^\neg(x))$. By (3.8) 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* $n \in \mathsf{G}$ if $\forall i, j : \beta(n_i) = \beta(n_j)$ and call β *stable* if β is stable on every vertex of G.

Lemma 3.7. *For every* G*, every* $\beta \in sol(\mathsf{G}^\omega)$ *is stable.*

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

The immediate corollary of the two lemmata is the following:

Theorem 3.8. *For every* $\mathsf{G} : (sol(\mathsf{G}))^\omega = sol(\mathsf{G}^\omega)$.

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. This special case was addressed in [5] and we merely generalized its results allowing infinite graphs with sinks. When digraphs are infinitely branching, the theorem allows us to equate the problem of solvability of arbitrary digraphs and the problem of solvability of dags, as will be done in the following section.

4 Solvability of digraphs and dags

This section contains the main results of the paper concerning the complexity of solvability of digraphs and dags. In Section 4.1, we briefly comment on the universal solvability of finite, well-founded and finitely branching dags and NP-completeness of solvability of finite digraphs. In Section 4.2, we show that solvability of recursive digraphs and dags is Σ^1_1-complete and that, as a consequence, this also holds for satisfiability of recursive PL$^\omega$-theories. In Section 4.3, we give a new equivalent of the axiom of choice in terms of solvability of infinitely branching digraphs and dags.

4.1 The finite case

As noted after Proposition 3.2, every finite dag has a unique kernel and such dags are obviously less expressive than arbitrary finite digraphs. E.g., a kernel for a dag can be constructed in linear time while the problem of kernel existence for general (finite) digraphs is NP-complete, [4]. The linear reductions $\mathcal{T}$ and $\mathcal{G}$ from Section 2 give a new proof of this fact, which has been noticed earlier in [6].

Two kinds of infinite dags, always possessing solutions, can be seen under the current heading of finiteness. (1) A well-founded dag may possess infinite branchings but has only finite paths and is solvable by Proposition 3.2.3. (2) A finitely branching dag, on the other hand, may possess infinite paths but no infinitely branching vertices. It, too, is always solvable, as we now explain.

Any finitely branching digraph G represents a propositional theory $\mathsf{T} = \mathcal{T}(\mathsf{G})$ and, by (2.3), its solvability is equivalent to the consistency of T. G is infinite iff T is, and we consider this case. Every finite subtheory of T corresponds to an appropriate subgraph of G (with additional 2-cycles added at new sinks s, arising by removing parts of the graph; these 2-cycles can be viewed as definitional extensions of T by $s' \leftrightarrow \neg s$). By compactness, G is solvable iff all such finite subgraphs of G are solvable. Since every subgraph of a dag is a dag and all finite dags are solvable, there is no need to add 2-cycles in the case of finitely branching dags (we get models of finite subsets of T strengthened by some axioms s for new sinks s). Compactness thus implies consistency of T and hence solvability of G. This is the gist of the argument for solvability of finitely branching dags, which can be found in [11].

4.2 Recursive dags

A consistent, recursive, propositional theory may have no recursive models or, in terms of digraphs, a solvable, recursive digraph may have no recursive solutions. Since lifting (3.7) 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 finitely branching dags, using (a variation of) the Kleene-tree.

Theorem 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 inseparable sets A and B. This means that $A \cap B = \varnothing$ and for no recursive set C we have $A \subset C$ 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 $0^0, 0^1, 0^2, ..., 0^k$ 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 consists of all sequences 0^k 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 $0^n 1 0^k$ for all $n \in \mathsf{N}$ and $0^k \in T_n$. If there exists a recursive $\alpha \in sol(T)$, then the set $C = \{n \in \mathsf{N} \mid \alpha(0^n 1) = \mathbf{1}\}$ is recursive and separates A and B. Contradiction. □

The theorem above concerns finitely branching trees, and a fortiori, finitely branching dags. As we will see in Theorem 4.3 below, the complexity of solvability of arbitrary, possibly infinitely branching dags, leaves the arithmetical hierarchy and is Σ^1_1-complete. So let us observe that going over now to digraphs with infinite branching, the difference in expressive power between dags and general digraphs disappears. On the one hand, dags with infinite branching may be unsolvable (e.g., the Yablo dag), while on the other hand, for an arbitrary digraph G it holds that the solutions of G correspond bijectively to the solutions of the dag-lifting of G, by Theorem 3.8. Therefore we will be able to show the results in the sequel both for arbitrary dags, and for arbitrary digraphs.

For the statement of our complexity results, we now specify the digraph solvability problem GSOL, and the Σ^1_1-complete problem to which we will link it, the non-well-foundedness problem NWF for binary relations. In both cases we employ encodings $\ulcorner R \urcorner$ of recursive binary relations $R \subseteq \mathsf{N} \times \mathsf{N}$ as natural number codes of Turing machines that compute the characteristic function of R (assuming standard constructions for these codes, we do not specify the details).

Digraph Solvability Problem GSOL
Instance: An encoding $\ulcorner E \urcorner$ of the edge relation E of a rec. digraph $\mathsf{G} = \langle \mathsf{N}, E \rangle$
Question: Is G solvable?
Recognition problem: $\{\ulcorner E \urcorner : \mathsf{G} = \langle \mathsf{N}, E \rangle$ is a recursive dag that is solvable$\}$

A binary relation $R \subseteq A \times A$ on a set A is called *well-founded* if there are no infinitely descending R-chains, that is, no infinite sequences $x_0, x_1, x_2, \ldots$ such that for all $n \in \mathsf{N} : \langle x_{n+1}, x_n \rangle \in R$.

Non-Well-Foundedness Problem NWF
Instance: an encoding $\ulcorner R \urcorner$ of a recursive binary relation R on N
Question: Is R not well-founded?
Recognition problem: $\{\ulcorner R \urcorner : R$ is a rec., not well-founded, binary relation on $\mathsf{N}\}$

Let $A, B \subseteq \mathsf{N}$. 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 computable function $f : \mathsf{N} \to \mathsf{N}$ such that, for every $n \in \mathsf{N}$, $n \in A$ if and only if $f(n) \in B$.

Lemma 4.2. NWF $\leq_m$ GSOL.

Proof. For every binary relation $R \subseteq \mathsf{N} \times \mathsf{N}$ we call $G(R) = \langle \mathsf{N}, R^{\smile} \rangle$, where $R^{\smile}$ is the converse of R, the digraph *associated* with R; and we say, for $x, y \in \mathsf{N}$, that *y is reachable from x via an R-chain* if there are $z_0, z_1, \ldots, z_n \in \mathsf{N}$ such that $y = z_n \mathrel{R} z_{n-1} \mathrel{R} \ldots \mathrel{R} z_1 \mathrel{R} z_0 = x$.

In the construction below of a computable many-one reduction from NWF to GSOL we will assume that given recursive relations $R \subseteq \mathsf{N} \times \mathsf{N}$ are such that:

$$\langle 0,0 \rangle \notin R \;\wedge (\forall \langle i,j \rangle \in R)\big[\, j \text{ (and hence also } i) \text{ is reachable from 0 via an } R\text{-chain}\big]. \tag{4.9}$$

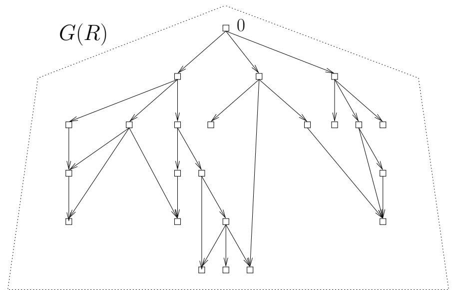


Fig. 3: Left: example of a well-founded relation $R \subseteq \mathsf{N} \times \mathsf{N}$ such that (4.9) holds, illustrated via the digraph $G(R)$ associated with R. Right: the tree unravelling $T(R) := T_{G(R);0}$ of $G(R)$ at vertex 0.

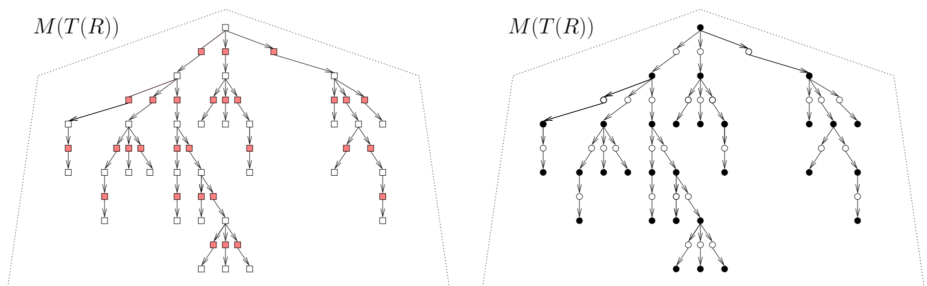


Fig. 4: Left: the modification $M(T(R))$ of the tree unravelling $T(R)$ of $G(R)$ as in Fig. 3, which is obtained by splitting the edges in $T(R)$, thereby creating intermediate vertices. Right: its unique solution, which maps each of the intermediate vertices to $\mathbf{0}$.

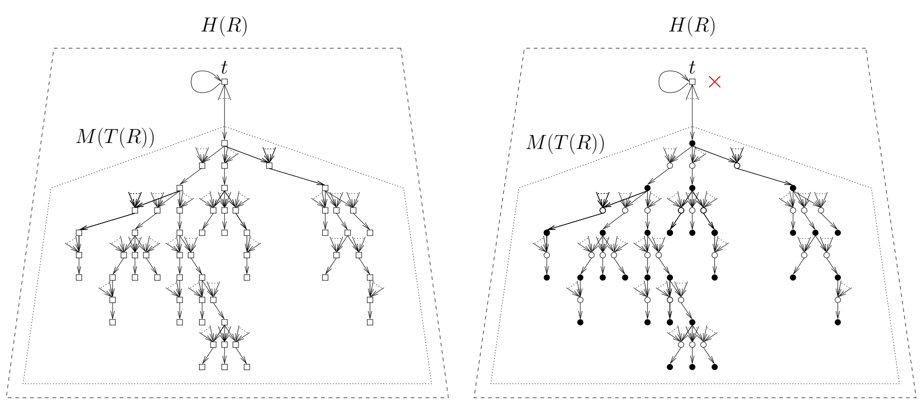


Fig. 5: Left: the digraph $H(R)$ constructed for R as chosen for $G(R)$ in Fig. 3 by extending $M(T(R))$ (see Fig. 4) with a new vertex t on top, an edge that is a self-loop on t, and edges from t to all of the intermediate vertices in $M(T(R))$. Right: the unique solution of its subgraph $M(T(R))$ *cannot* be extended to a solution of $H(R)$.

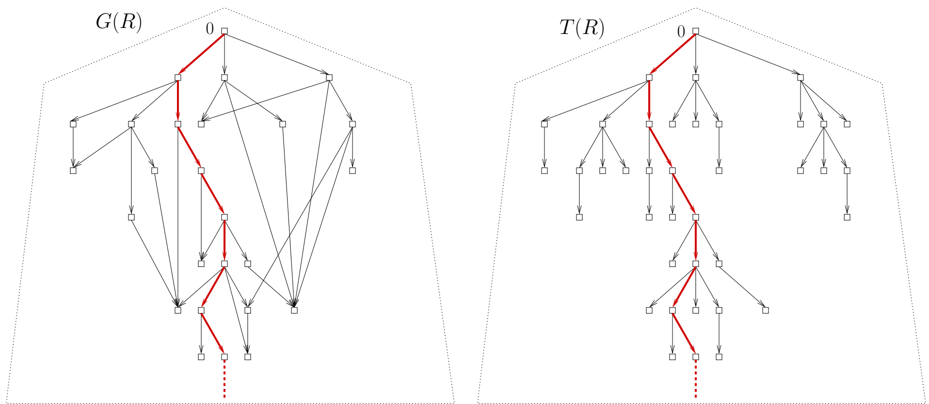


Fig. 6: Left: example of a non-well-founded relation $R \subseteq \mathsf{N} \times \mathsf{N}$ such that (4.9) holds, illustrated via the digraph $G(R)$ associated with R. Right: the tree unravelling $T(R) := T_{G(R);0}$ of $G(R)$ at vertex 0. Both digraphs contain an infinite path.

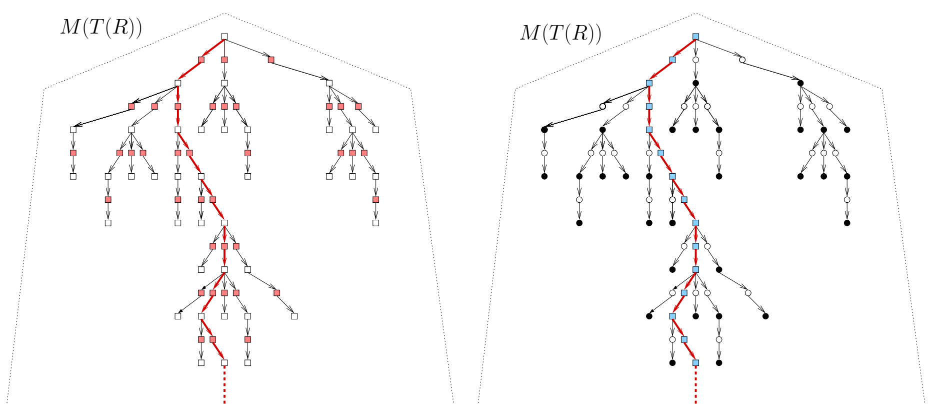


Fig. 7: Left: the modification $M(T(R))$ of the tree unravelling $T(R)$ of $G(R)$ as in Fig. 6 which is obtained by splitting the edges in $T(R)$. Right: solutions are uniquely determined on the well-founded part of $M(T(R))$, but not on the set of vertices that are situated on, or are the starting points of, infinite paths.

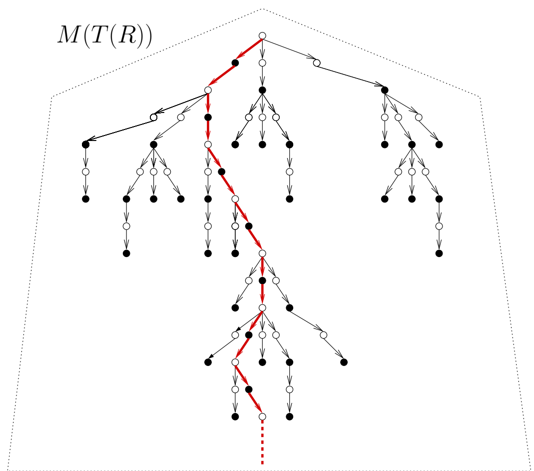


Fig. 8: There is a solution of $M(T(R))$ as in Fig. 7 that assigns $\mathbf{1}$ to all those intermediate vertices in $M(T(R))$ that are situated on infinite paths.

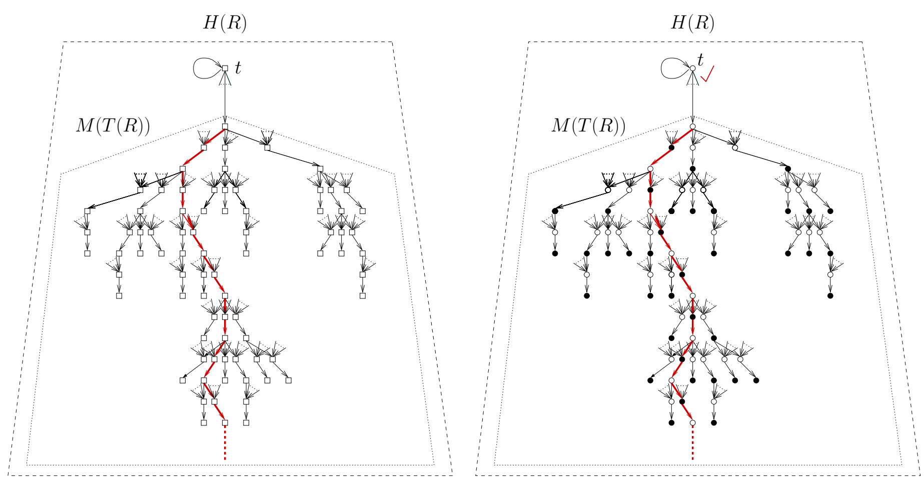


Fig. 9: Left: the digraph $H(R)$ constructed for R as chosen for $G(R)$ in Fig. 6 by extending $M(T(R))$ (see Fig. 7) with a new vertex t on top, an edge that is a self-loop on t, and edges from t to all of the intermediate vertices in $M(T(R))$. Right: The solution of its subgraph $M(T(R))$ as illustrated in Fig. 8 can be extended to a solution of $H(R)$ by assigning $\mathbf{0}$ to the vertex t on the top.

We are justified in doing so, because every binary recursive relation R on N can be embedded into a binary recursive relation R' on N such that (4.9) holds for $R := R'$, and such that well-foundedness is preserved and reflected: given an arbitrary recursive binary relation $R \subseteq \mathsf{N} \times \mathsf{N}$, the binary relation:

$$R' := \{\langle i, 0\rangle \mid i > 0\} \cup \{\langle i+1, j+1\rangle \mid \langle i, j\rangle \in R\} \subseteq \mathsf{N} \times \mathsf{N}$$

is recursive, satisfies (4.9) for $R := R'$, and furthermore, R' is well-founded if and only if R is well-founded.

Let $\mathsf{H} = \langle \mathsf{N}, P\rangle$ be a digraph, and $v \in \mathsf{N}$ a vertex of H. By the *tree unravelling* of H at v we mean the digraph $T_{\mathsf{H};v} = \langle \Pi_{\mathsf{H};v}, E\rangle$ with:

$$\Pi_{\mathsf{H};v} := \{v_0 v_1 \dots v_n \in \mathsf{N}^* \mid v_i \in \mathsf{N},\, v_0 = v,\, \forall i \in \{0, 1, \dots, n-1\}.\langle v_i, v_{i+1}\rangle \in P\}$$
$$E := \{\langle wv, wvv'\rangle \in \Pi_{\mathsf{H};v} \times \Pi_{\mathsf{H};v} \mid w \in \mathsf{N}^*,\, v, v' \in \mathsf{N},\, \langle v, v'\rangle \in P\}$$

That is, the set of vertices of $T_{\mathsf{H};v}$ is the set of paths in H from v, and the edge relation of $T_{\mathsf{H};v}$ is obtained as a lifting from the edge relation of H. It is easy to see that $T_{\mathsf{H};v}$ is a tree with root v, which has a recursive edge-relation if this holds for H. Note that if a binary relation $R \subseteq \mathsf{N} \times \mathsf{N}$ has the property (4.9), then every path in $G(R)$ extends to a path that starts at the vertex 0; it follows that R is well-founded if and only if $T_{G(R);0}$ does not contain an infinite path.

Now we define a computable many-one reduction from NWF to GSOL as follows. For a given recursive binary relation $R \subseteq \mathsf{N} \times \mathsf{N}$ with (4.9), a recursive digraph $H(R)$ is constructed in a recursive way such that $H(R)$ consists of:

- a single vertex t with a loop on the top;
- a modification $M(T(R))$ of the tree $T(R) := T_{G(R);0}$, the tree unravelling at 0 of the digraph $G(R)$ associated with R; the modification is obtained by splitting each edge of $T(R)$ into two consecutive edges, inserting a new, *intermediate* vertex between the edge-connected old vertices;
- an edge from the top vertex t to every new, intermediate vertex in $M(T(R))$.

Figure 5 shows, on the left, the example of the digraph $H(R)$ that is constructed for a well-founded binary relation R on N such that (4.9) holds with associated digraph $G(R)$ as in Figure 3. The tree unravelling $T(R)$ of $G(R)$ is well-founded (see Figure 3, on the right). Consequently, this also holds for its modification $M(T(R))$, the tree that resulted by splitting the edges of $T(R)$ (see Figure 4, on the left). The unique solution of $M(T(R))$ assigns **1** to the old vertices symbolized by black filled circles, and **0** to the new, intermediate vertices symbolized by unfilled circles (see Figure 4, on the right). The digraph $H(R)$ itself is unsolvable: All edges from t to vertices in its subgraph $M(T(R))$ target intermediate vertices, and there always meet the value **0**. Hence unsolvability of $H(R)$ follows from the unsolvability of its t-loop by Proposition 3.2.3.

Figure 9 shows the example of the digraph $H(R)$ that is constructed for a non-well-founded relation R such that (4.9) with associated digraph $G(R)$ as in Figure 6, on the left. In this case both of $T(R)$ and $M(T(R))$ are non-well-founded trees (see Figure 6 on the right, and Figure 7 on the left). In particular,

$M(T(R))$ contains an infinite path. As a consequence there exists a solution of $M(T(R))$ that assigns **1** to all intermediate vertices on the infinite path, and **0** to intermediate vertices with no infinite paths below them (see Figure 8). This solution of $M(T(R))$ induces **0** at the top vertex t with the loop, and thereby extends to a solution for $H(R)$ (see Figure 9). Hence $H(R)$ is solvable.

Reasoning as above for the examples in Figures 3–5 and Figures 6–9 justifies the first of the following four steps, which hold for every binary relation R on N with (4.9):

$$\begin{aligned} H(R) \text{ is solvable} &\iff M(T(R)) \text{ has an infinite path} \\ &\iff T(R) = T_{G(R);0} \text{ has an infinite path} \\ &\iff G(R) \text{ has an infinite path} \\ &\iff R \text{ is not well-founded} \end{aligned}$$

(The last three steps are easy to verify.) Together with the facts that for every recursive binary relation R on N a recursive relation R' with (4.9) such that R' well-founded if and only if R is well-founded can be constructed, that the digraphs $T(R')$, $M(T(R'))$, and $H(R')$ are recursive, and that encodings of the edge-relation in each of these digraphs can be obtained from the code $\ulcorner R \urcorner$ of R by a recursive procedure, it follows that the described transformation from recursive relations R to recursive digraphs $H(R')$ constitutes a computable many-one reduction from NWF to GSOL. □

Theorem 4.3. *The solvability problem* GSOL *for digraphs is* Σ^1_1*-complete.*

Proof. Since NWF is Σ^1_1-complete [13, 3], and NWF $\leq_m$ GSOL by Lemma 4.2, it remains to show that GSOL $\in \Sigma^1_1$. For this we describe a Σ^1_1-formula $\psi(m)$ that expresses solvability for recursive digraphs, and more precisely, has the property:

$$\psi(\ulcorner E \urcorner) \iff \mathsf{G} \text{ is solvable} \qquad (\text{for all recursive digraphs } \mathsf{G} = \langle \mathsf{N}, E \rangle).$$

We let $\psi(m)$ be the Σ^1_1-formula:

$$\exists K\, \forall n \big[\, n \in K \;\leftrightarrow\; \forall n' (\, \mathit{EdgeIn}(n, n', m) \;\to\; n' \notin K\,)\,\big]$$

(n, n' vary over natural numbers, and K over sets of natural numbers), where $\mathit{EdgeIn}(n, n', m)$ is a Σ^0_0-formula with the property that:

$$\mathit{EdgeIn}(n, n', \ulcorner E \urcorner) \iff E(n, n') \qquad (\text{for all } n, n' \in \mathsf{N})$$

holds for all recursive digraphs $\mathsf{G} = \langle \mathsf{N}, E \rangle$ (based on the chosen Turing maching encoding, it is routine work to construct such a formula EdgeIn). □

The following corollary states Σ^1_1-completeness also for two problems that are closely related to GSOL: the solvability problem DSOL for dags, and the satisfiability problem for recursive PL$^\omega$-theories, ISAT. Here we call a PL$^\omega$-theory *recursive* if it contains a countable set of propositions, the set of (the codes of) its propositions is recursive, and if each proposition (typically an infinite object) has itself a recursive code.[6]

Corollary 4.4. *The following problems are* Σ^1_1*-complete:*
GSOL*: Is a given recursive digraph solvable?*
DSOL*: Is a given recursive dag solvable?*
ISAT*: Is a given, recursive,* PL^ω *theory satisfiable?*

Proof. GSOL $\leq_m$ DSOL is a consequence of Theorem 3.8: the dag-lifting G^ω of a recursive digraph G is a recursive dag that is solvable if and only if G is solvable. Since DSOL is a sub-problem of GSOL, this implies Σ^1_1-completeness of DSOL in view of the theorem.

Together with ISAT $\in \Sigma^1_1$, which is not difficult to show for concrete encodings[7] of recursive PL^ω-theories, Σ^1_1-completeness of ISAT follows from the fact that GSOL $\leq_m$ ISAT holds: this in turn can be proved by applying the reduction used in Section 2 to show (2.3). ☐

Note that a direct proof for dags can be obtained showing NWF $\leq_m$ DSOL by replacing, in the proof of Lemma 4.2, the loop at the top of $H(R)$ by a Yablo dag, with edges from all its vertices to all the intermediate vertices in $M(T(R))$.

4.3 Arbitrary digraphs

We proceed now to arbitrary, infinitely branching digraphs, and formulate a new equivalent to the axiom of choice in terms of digraph solvability.

Let $\mathcal{F} = \{X_i \mid i \in I\}$ be an indexed family of sets. The *cartesian product* of sets in $\mathcal{F}$ is defined by $\prod_{i\in I} X_i := \left\{f : I \to \bigcup_{i\in I} \mid \forall i \in I : f(i) \in X_i\right\}$ and the *disjoint union* by $\biguplus_{i\in I} X_i := \left\{\langle x, i\rangle \mid i \in I,\, x \in X_i\right\}$.

Let X be a set. A *choice function on* X is a function $f : X\backslash\{\emptyset\} \to \bigcup X$ such that $f(x) \in x$ for all $x \in X\backslash\{\emptyset\}$. The *axiom of choice* (AC) is the statement: for every set X, there exists a choice function on X.

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

---

[6] The complexity result for ISAT in Corollary 4.4 is invariant under 'reasonable' precise definitions of the concept 'recursive PL^ω-theory' and under 'reasonable' encodings. One possibility is to limit considerations to PL^ω-theories in infinitary clausal form, in which infinite disjunction is restricted to sets of literals. A recursive PL^ω-theory T in infinitary clause form can be encoded by a recursive ternary predicate C on N such that the set of clauses $\mathcal{C}_i := \{p_j \mid C(i, j, 1)\} \cup \{\neg p_j \mid C(i, j, 2)\}$ for $i \in \mathsf{N}$ represents the infinitary clause form of T.

[7] For every recursive PL^ω-theory T in infinitary clausal form that is encoded through a ternary predicate C_T as described in footnote 6, satisfiability of T can be expressed, using a Σ^1_1-formula, as: $\exists M \forall i \exists j (j \in M \wedge R(\ulcorner C_\mathsf{T}\urcorner, i, j, 1) \vee j \notin M \wedge R(\ulcorner C_\mathsf{T}\urcorner, i, j, 2))$, where M varies over sets of natural numbers, and where R is a recursive predicate such that, for all recursive ternary predicates P, and for all i, j, k, it holds: $R(\ulcorner P\urcorner, i, j, k) \Leftrightarrow P(i, j, k)$.

Theorem 4.5. *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. We use the well-known fact that, over ZF, AC is equivalent to:

(CP) For every indexed family $\{X_i : i \in I\}$ of non-empty sets, the cartesian product $\prod_{i\in I} X_i$ is non-empty.

Hence it suffices to show the two implications in (CP) $\Leftrightarrow$ (GS).

(CP) $\Rightarrow$ (GS). Let $\{\mathsf{G}_i = \langle G_i, E_i\rangle \mid i \in I\}$ be an indexed family of solvable digraphs. By (CP) it follows that the product $\prod_{i\in I} sol(\mathsf{G}_i)$ is non-empty. Since 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 $\alpha_f(\langle v, i\rangle) := f(i)(v)$ for all $\langle v, i\rangle \in \biguplus_{i\in I} G_i$, it follows that $\biguplus_{i\in I} \mathsf{G}_i$ is solvable.

(GS) $\Rightarrow$ (CP). Let $\{X_i : i \in I\}$ be a collection of non-empty sets. For each $i \in I$, let G_i be the complete digraph with X_i as its set of vertices. By Proposition 3.3 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 $\alpha|_{\mathsf{G}_i}$ of α to G_i is a solution of G_i. Consequently, every solution α of the disjoint union of the G_i's 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 unique $x \in X_i$ such that $\alpha|_{\mathsf{G}_i}(x) = \mathbf{1}$. □

By employing dag-liftings and Theorem 3.8, this result can be 'lifted' to dags.

Corollary 4.6. *Over* ZF*,* AC *is equivalent with the following statements:*

(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 4.5 it suffices to show (DS) $\Rightarrow$ (GS). For this, let us assume (DS), and let $\{\mathsf{G}_i \mid i \in I\}$ be an indexed family of solvable digraphs. By Theorem 3.8, the dag-lifting G_i^ω of G_i is solvable for every $i \in I$. Then it follows by (DS) that the dag $D := \biguplus_{i\in I} \mathsf{G}_i^\omega$ 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 3.8 again, that G is solvable. □

See Figure 10 for a picture that suggests a direct proof of (DS)$\Rightarrow$(AC): Every solution of the dag-lifting G^ω of a disjoint union $\bigcup_{i\in I} \mathsf{G}_i$ of complete digraphs G_i induces a choice function on the vertices of the disjoint union. In the picture this is explained for the dag-lifting H^ω of the complete two-vertex digraph H.

5 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:

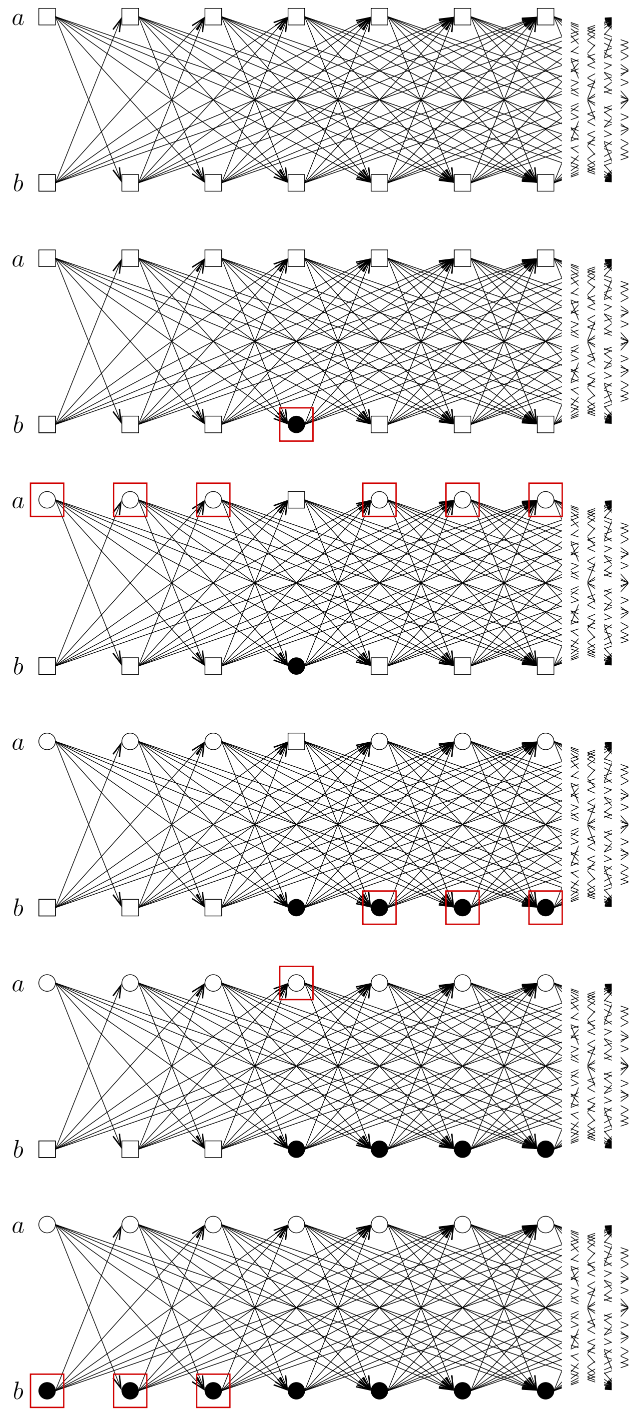


Fig. 10: Every solution of the dag-lifting H^ω of the complete two-vertex digraph $\mathsf{H} = \langle\{a,b\},\{\langle a,b\rangle,\langle b,a\rangle\}\rangle$ induces a choice between a and b. (For the first step, note that a solution of H^ω cannot assign $\mathbf{0}$ to every vertex of H^ω.)

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 a 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 for disjoint unions of digraphs (or respectively, disjoint unions of dags) that have kernels is equivalent to the Axiom of Choice.

Acknowledgments. The proof of Theorem 4.1 is due to Dag Normann, who deserves thanks also for discussion of several other points addressed in the paper. We thank also Sjur Dyrkolbotn, Ulrich Kohlenbach, Albert Visser, and Vincent van Oostrom for comments and discussion of various issues related to the presented results.

References

1. Martine Anciaux-Mendeleer and Pierre Hansen. On kernels in strongly connected graphs. *Networks*, 7(3):263–266, 1977.
2. Endre Boros and Vladimir Gurvich. Perfect graphs, kernels and cooperative games. *Discrete Mathematics*, 306:2336–2354, 2006.
3. J. Castro and F. Cucker. Non-deterministic ω-computations and the analytical hierarchy. *Zentralblatt f. Math. Logik u. Grundl. d. Mathematik*, 35:333–342, 1989.
4. Vašek Chvátal. On the computational complexity of finding a kernel. Technical Report CRM-300, Centre de Recherches Mathématiques, Univeristé de Montréal, 1973. `http://users.encs.concordia.ca/~chvatal`.
5. Roy Cook. Patterns of paradox. *The Journal of Symbolic Logic*, 69(3):767–774, 2004.
6. Nadia Creignou. The class of problems that are linearly equivalent to Satisfiability or a uniform method for proving NP-completeness. *Theoretical Computer Science*, 145:111–145, 1995.
7. Pierre Duchet. Graphes noyau-paraits, II. *Annals of Discrete Mathematics*, 9:93–101, 1980.

8. Pierre Duchet and Henry Meyniel. Une jeneralization du theoreme de Richardson sur l'existence de noyaux dans les graphes orientes. *Discrete Mathematics*, 43(1):21–27, 1983.
9. Hortensia Galeana-Sánchez and Victor Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76, 1984.
10. John R. Isbell. On a theorem of Richardson. *Proceedings of the AMS*, 8(5):928–929, 1957.
11. Eric C. Milner and Robert E. Woodrow. On directed graphs with an independent covering set. *Graphs and Combinatorics*, 5:363–369, 1989.
12. Moses Richardson. Solutions of irreflexive relations. *The Annals of Mathematics, Second Series*, 58(3):573–590, 1953.
13. H. Rogers. *Theory of Recursive Functions and Effective Computability*. MacGraw–Hill, 1967.
14. John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944 (1947).
15. Stephen Yablo. Paradox without self-reference. *Analysis*, 53(4):251–252, 1993.