A Tableau-Based Decision Procedure for a Fragment of Graph Theory Involving Reachability and Acyclicity*

Domenico Cantone[1] and Calogero G. Zarba[2]

[1] University of Catania
[2] University of New Mexico

Abstract. We study the decision problem for the language **DGRA** (*directed graphs with reachability and acyclicity*), a quantifier-free fragment of graph theory involving the notions of reachability and acyclicity.
We prove that the language **DGRA** is decidable, and that its decidability problem is *NP*-complete. We do so by showing that the language enjoys a *small model property*: If a formula is satisfiable, then it has a model whose cardinality is polynomial in the size of the formula.
Moreover, we show how the small model property can be used in order to devise a tableau-based decision procedure for **DGRA**.

1 Introduction

Graphs arise naturally in many applications of mathematics and computer science. For instance, graphs arise as suitable data structures in most programs. In particular, when verifying programs manipulating pointers [3], one needs to reason about the reachability and acyclicity of graphs.

In this report we introduce the language **DGRA** (*directed graphs with reachability and acyclicity*), a quantifier-free many-sorted fragment of directed graph theory. The language **DGRA** contains three sorts: node for nodes, set for sets of nodes, and graph for graphs. In the language **DGRA** graphs are modeled as binary relations over nodes or, alternatively, as sets of pairs of nodes. The language **DGRA** contains the set operators $\cup$, $\cap$, $\setminus$, $\{\cdot\}$ and the set predicates $\in$, and $\subseteq$. It also contains:

- a predicate $\mathit{reachable}(a, b, G)$ stating that there is a nonempty path going from node a to node b in the graph G;
- a predicate $\mathit{acyclic}(G)$ stating that the graph G is acyclic.

We prove that the language **DGRA** is decidable, and that its decidability problem is *NP*-complete. We do so by showing that the language enjoys a *small model property*: If a formula is satisfiable, then it has a model $\mathcal{A}$ whose cardinality is polynomial in the size of the formula.

* This work is partly supported by grants NSF ITR CCR-0113611 and NSF CCR-0098114.

More precisely, let φ be a satisfiable formula in the language **DGRA**, and let m and g be, respectively, the number of variables of sort node and graph occurring in φ. Then there exists a model $\mathcal{A}$ of φ such that its associated domain A_{node} has cardinality less than or equal to $m + m^2 \cdot g^2$.

At first sight, it seems that the small model property only suggests a brute force decision procedure for **DGRA**, consisting in enumerating all models up to a certain size. However, the bound on the cardinality of A_{node} can be cleverly exploited in order to devise a tableau-based decision procedure for **DGRA**.

Roughly speaking, the idea is as follows. Suppose that T is a tableau for the formula φ. We devise the tableau rules in such a way that at most $m^2 \cdot g^2$ fresh variables of sort node are added to any branch B of T. Furthermore, the tableau rules need to ensure that these fresh variables are to be interpreted as distinct from each other, and distinct from every old variable of sort node already occurring in φ.

We use the above intuition in order to devise a tableau calculus for **DGRA** that is terminating, sound, and complete. Consequently, we obtain a decision procedure for **DGRA** that is, at least potentially, more efficient than a naive brute force approach.

Organization of the report. In Section 2 we define a notion of paths that will be used in the rest of the report. In Section 3 we define the syntax and semantics of the language **DGRA**. In Section 4 we present our tableau calculus for **DGRA**. In Section 5 we show one example of our tableau calculus in action. In Section 6 we prove that our tableau calculus is terminating, sound, and complete, and therefore it yields a decision procedure for **DGRA**. In Section 7 we survey on related work. In Section 8 we draw final conclusions.

2 Paths

Definition 1 (Paths and cycles). Let A be a set. A (simple) PATH π over A is a sequence

$$\pi = \langle \nu_1, \ldots, \nu_n \rangle$$

such that

(a) $n \geq 2$;
(b) $\nu_i \in A$, for each $1 \leq i \leq n$;
(c) $\{\nu_1, \nu_n\} \cap \{\nu_2, \ldots, \nu_{n-1}\} = \emptyset$;
(d) $\nu_i \neq \nu_j$, for each $1 < i < j < n$.

A CYCLE is a path $\pi = \langle \nu_1, \ldots, \nu_n \rangle$ such that $\nu_1 = \nu_n$. □

Note that, according to Definition 1, the sequence $\langle a, b, b, c \rangle$ is *not* a path.

We denote with *paths*(A) the set of all paths over A. Let $\pi = \langle \nu_1, \ldots, \nu_n \rangle$ be a path in *paths*(A), and let $R \subseteq A \times A$ be a binary relation. We write $\pi \subseteq R$ when $(\nu_i, \nu_{i+1}) \in R$, for each $1 \leq i < n$.

If $\pi = \langle \nu_1, \ldots, \nu_n \rangle$ is a path, we let $nodes(\pi) = \{\nu_1, \ldots, \nu_n\}$. Given a path $\pi = \langle \nu_1, \ldots, \nu_n \rangle$, we define a function—which for simplicity we continue to denote with π—from $nodes(\pi)$ to $nodes(\pi)$ as follows:

$$\begin{aligned} \pi(\nu_i) &= \nu_{i+1}\,, && \text{for each } 1 \leq i < n\,, \\ \pi(\nu_n) &= \nu_n\,, && \text{if } \nu_n \neq \nu_1\,. \end{aligned}$$

Note that this function is well-defined because of conditions (c) and (d) of Definition 1.

Let $\pi = \langle \nu_1, \ldots, \nu_n \rangle$ be a path in $paths(A)$, let $X \subseteq A$, and assume that $\nu_i \in X$. Then we write

$$first(\nu_i, \pi, X) = \nu_j\,,$$

whenever j is the unique index such that:

- $i \leq j$;
- $\nu_j \in X$;
- $\nu_k \notin X$, for each $i \leq k < j$.

Definition 2 (Basic paths). Let $\pi = \langle \nu_1, \ldots, \nu_n \rangle$ be a path in $paths(A)$, and let $X \subseteq A$. We say that π is BASIC with respect to X if the following conditions hold:

- $\nu_1 \in X$ and $\nu_n \in X$;
- $\nu_i \notin X$, for each $1 < i < n$. $\square$

3 The language DGRA

3.1 Syntax

The language **DGRA** (*directed graphs with reachability and acyclicity*) is a quantifier-free many-sorted language with equality [6]. Its sorts and symbols are depicted in Figure 1. Note that some symbols of the language are overloaded.

Definition 3. A **DGRA**-FORMULA is a well-sorted many-sorted formula constructed using:

- the function and predicate symbols in Figure 1;
- variables of sort τ, for $\tau \in \{\mathsf{node}, \mathsf{set}, \mathsf{graph}\}$;
- the equality predicate $=$;
- the propositional connectives $\neg$, $\wedge$, $\vee$, and $\rightarrow$. $\square$

Given a **DGRA**-formula φ, we denote with $vars_\tau(\varphi)$ the set of τ-variables occurring in φ. Moreover, we let $vars(\varphi) = vars_{\mathsf{node}}(\varphi) \cup vars_{\mathsf{set}}(\varphi) \cup vars_{\mathsf{graph}}(\varphi)$.

To increase readability, in the rest of the report we will use the abbreviations depicted in Figure 2.

Sorts

node	nodes
set	sets of nodes
graph	graphs, modeled as sets of pairs of nodes

Symbols

	Function symbols	Predicate symbols
Sets	$\emptyset_{\mathsf{set}} : \mathsf{set}$ $\cup, \cap, \setminus : \mathsf{set} \times \mathsf{set} \to \mathsf{set}$ $\{\cdot\} : \mathsf{node} \to \mathsf{set}$	$\in : \mathsf{node} \times \mathsf{set}$ $\subseteq : \mathsf{set} \times \mathsf{set}$
Binary relations	$\emptyset_{\mathsf{graph}} : \mathsf{graph}$ $\cup, \cap, \setminus : \mathsf{graph} \times \mathsf{graph} \to \mathsf{graph}$ $\{(\cdot,\cdot)\} : \mathsf{node} \times \mathsf{node} \to \mathsf{graph}$	$(\cdot,\cdot) \in \cdot : \mathsf{node} \times \mathsf{node} \times \mathsf{graph}$ $\subseteq : \mathsf{graph} \times \mathsf{graph}$
Reachability		$\mathit{reachable} : \mathsf{node} \times \mathsf{node} \times \mathsf{graph}$ $\mathit{acyclic} : \mathsf{graph}$

Figure 1: The language **DGRA**.

Syntactic sugar	Official formula
$a \notin x$	$\neg(a \in x)$
$G(a,b)$	$(a,b) \in G$
$\neg G(a,b)$	$\neg((a,b) \in G)$
$G^+(a,b)$	$\mathit{reachable}(a,b,G)$
$\neg G^+(a,b)$	$\neg\mathit{reachable}(a,b,G)$

Figure 2: Syntactic sugar for the language **DGRA**.

3.2 Semantics

Definition 4. Let V_τ be a set of τ-variables, for $\tau \in \{\mathsf{node}, \mathsf{set}, \mathsf{graph}\}$, and let $V = V_{\mathsf{node}} \cup V_{\mathsf{set}} \cup V_{\mathsf{graph}}$.

A **DGRA**-INTERPRETATION over V is a many-sorted interpretation satisfying the following conditions:

– each sort τ is mapped to a set A_τ such that:
  - $A_{\mathsf{node}} \neq \emptyset$;
  - $A_{\mathsf{set}} = \mathcal{P}(A_{\mathsf{node}})$;
  - $A_{\mathsf{graph}} = \mathcal{P}(A_{\mathsf{node}} \times A_{\mathsf{node}})$;

– each variable $u \in V$ of sort τ is mapped to an element $u^{\mathcal{A}} \in A_\tau$;

– the set symbols $\emptyset_{\mathsf{set}}$, $\cup$, $\cap$, $\setminus$, $\{\cdot\}$, $\in$, and $\subseteq$ are interpreted according to their standard interpretation over sets of nodes;

- the binary relation symbols $\emptyset_{\mathsf{graph}}$, $\cup$, $\cap$, $\setminus$, $\{(\cdot,\cdot)\}$, $(\cdot,\cdot) \in \cdot$, and $\subseteq$ are interpreted according to their standard interpretation over sets of pairs of nodes;
- $[\mathit{reachable}(a,b,G)]^{\mathcal{A}} = \mathit{true}$ if and only if there exists a path $\pi \in \mathit{paths}(A_{\mathsf{node}})$ such that $\pi \subseteq G^{\mathcal{A}}$;
- $[\mathit{acyclic}(G)]^{\mathcal{A}} = \mathit{true}$ if and only if there is no cycle $\pi \in \mathit{paths}(A_{\mathsf{node}})$ such that $\pi \subseteq G^{\mathcal{A}}$. □

If $\mathcal{A}$ is a **DGRA**-interpretation, we denote with $\mathit{vars}_\tau(\mathcal{A})$ the set of variables of sort τ that are interpreted by $\mathcal{A}$. Moreover, we let $\mathit{vars}(\mathcal{A}) = \mathit{vars}_{\mathsf{node}}(\mathcal{A}) \cup \mathit{vars}_{\mathsf{set}}(\mathcal{A}) \cup \mathit{vars}_{\mathsf{graph}}(\mathcal{A})$. If $V \subseteq \mathit{vars}(\mathcal{A})$, we let $V^{\mathcal{A}} = \{u^{\mathcal{A}} \mid u \in V\}$.

Definition 5. A **DGRA**-formula φ is **DGRA**-SATISFIABLE if there exists a **DGRA**-interpretation $\mathcal{A}$ such that φ is true in $\mathcal{A}$. □

3.3 Examples

The following are examples of valid statements over graphs that can be expressed in the language **DGRA**:

$$(G^+(a,b) \ \wedge \ G^+(b,c)) \ \rightarrow \ G^+(a,c) \tag{1}$$

$$(G \subseteq H \ \wedge \ \mathit{acyclic}(H)) \ \rightarrow \ \mathit{acyclic}(G) \tag{2}$$

$$(G^+(a,b) \ \wedge \ H^+(b,a)) \ \rightarrow \ \neg\mathit{acyclic}(G \cup H) \tag{3}$$

$$\neg\mathit{acyclic}(\{(a,b)\}) \ \rightarrow \ a = b \tag{4}$$

In particular:

- (1) expresses the transitivity property of the reachability relation.
- (2) states that if a graph H is acyclic, then any of its subgraphs is also acyclic.
- (3) states that if it is possible to go from node a to node b in a graph G, and from node b to node a in a graph H, then the graph $G \cup H$ contains a cycle.
- (4) states that if a graph contains only the edge (a,b), and it is not acyclic, then a and b must be the same node.

3.4 Normalized literals

Definition 6. A literal is FLAT if it is of the form $x = y$, $x \neq y$, $x = f(y_1,\dots,y_n)$, $p(y_1,\dots,y_n)$, and $\neg p(y_1,\dots,y_n)$, where $x, y, y_1, \dots, y_n$ are variables, f is a function symbol, and p is a predicate symbol. □

Definition 7. A **DGRA**-literal is NORMALIZED if it is a flat literal of the form:

$$\begin{array}{lll} a \neq b\,, & & \\ x = y \cup z\,, & x = y \setminus z\,, & x = \{a\}\,, \\ G = H \cup L\,, & G = H \setminus L\,, & G = \{(a,b)\}\,, \\ G^+(a,b)\,, & \neg G^+(a,b)\,, & \mathit{acyclic}(G)\,. \end{array}$$

where a, b are node-variables, x, y, z are set-variables, and G, H, L are graph-variables. □

Lemma 8. *The problem of deciding the* **DGRA**-*satisfiability of* **DGRA**-*formulae is equivalent to the problem of deciding the* **DGRA**-*satisfiability of conjunctions of normalized* **DGRA** *literals. Moreover, if the latter problem is in NP, so is the former.* □

PROOF. Clearly, if we can decide the **DGRA**-satisfiability of **DGRA**-formulae, we can decide the **DGRA**-satisfiability of conjunctions of normalized **DGRA**-literals.

Vice versa, assume that we can decide the **DGRA**-satisfiability of conjunctions of normalized **DGRA**-literals, and let φ be a **DGRA**-formula. In order to check φ for **DGRA**-satisfiability, we can translate φ in a DNF $\Gamma_1 \vee \cdots \vee \Gamma_n$ such that:

- each Γ_i is a conjunction of normalized **DGRA**-literals;
- φ is **DGRA**-satisfiable if and only if at least one of the Γ_i is **DGRA**-satisfiable.

This translation can be done with the help of the following satisfiability-preserving rewrite rules:[3]

$$\begin{array}{rcl}
x \neq y & \Longrightarrow & a \in x \setminus y \;\vee\; a \in y \setminus x \\
x = \emptyset_{\mathsf{set}} & \Longrightarrow & x = x \setminus x \\
x = y \cap z & \Longrightarrow & x = (y \cup z) \setminus ((y \setminus z) \cup (z \setminus y)) \\
a \in x & \Longrightarrow & \{a\} \subseteq x \\
x \subseteq y & \Longrightarrow & y = x \cup y \\
G \neq H & \Longrightarrow & (a,b) \in G \setminus H \;\vee\; (a,b) \in G \setminus H \\
G = \emptyset_{\mathsf{graph}} & \Longrightarrow & G = G \setminus G \\
G = H \cap L & \Longrightarrow & G = (H \cup L) \setminus ((H \setminus L) \cup (L \setminus H)) \\
G(a,b) & \Longrightarrow & \{(a,b)\} \subseteq G \\
G \subseteq H & \Longrightarrow & H = G \cup H \\
\neg acyclic(G) & \Longrightarrow & G^{+}(a,a)\,.
\end{array}$$

Clearly, if we can check each of the Γ_i for **DGRA**-satisfiability in nondeterministic polynomial time, then we can also check φ for **DGRA**-satisfiability in nondeterministic polynomial time. ■

3.5 The small model property

Definition 9. Let $\mathcal{A}$ be a **DGRA**-interpretation, let $V \subseteq vars_{\mathsf{node}}(\mathcal{A})$ be a set of node-variables, and let $k \geq 0$. We say that $\mathcal{A}$ is k-SMALL with respect to V if $A_{\mathsf{node}} = V^{\mathcal{A}} \cup A'$, for some set A' such that $|A'| \leq k$. □

[3] Note that some of these rewrite rules introduce new node-variables.

Lemma 10 (Small model property). *Let Γ be a conjunction of normalized* **DGRA**-*literals, and let $V_\tau = vars_\tau(\Gamma)$, for each sort τ. Also, let $m = |V_{\mathsf{node}}|$ and $g = |V_{\mathsf{graph}}|$. Then the following are equivalent:*

1. *Γ is* **DGRA**-*satisfiable;*
2. *Γ is true in a* **DGRA**-*interpretation $\mathcal{A}$ that is $(m^2 \cdot g^2)$-small with respect to V_{node}.* □

PROOF. $(2 \Rightarrow 1)$. Immediate.

$(1 \Rightarrow 2)$. Let $\mathcal{B}$ be a **DGRA**-interpretation satisfying Γ. We want to use $\mathcal{B}$ in order to construct a **DGRA**-interpretation $\mathcal{A}$ that satisfies Γ, and that is also $(m^2 \cdot g^2)$-small with respect to V_{node}.

For each $a, b \in V_{\mathsf{node}}$ and for each $G \in V_{\mathsf{graph}}$ such that the literal $G^+(a, b)$ is in Γ, we associate a shortest path $\pi_{a,b,G} = \langle \nu_1, \ldots, \nu_n \rangle$ satisfying the following conditions:

- $\nu_1 = a^{\mathcal{B}}$;
- $\nu_n = b^{\mathcal{B}}$;
- $\pi_{a,b,G} \subseteq R^{\mathcal{B}}$.

We define the following sets of paths:

$$\begin{aligned}
\Pi_1 &= \{\langle \nu, \nu \rangle \mid \nu \in V^{\mathcal{B}}_{\mathsf{node}}\}\,,\\
\Pi_2 &= \left\{ \pi_{a,b,G} \mid \begin{pmatrix} \text{the literal } G^+(a,b) \text{ is in } \Gamma \text{ and } G \in V_{\mathsf{graph}} \text{ and} \\ \pi_{a,b,G} \text{ is basic w.r.t. } V^{\mathcal{B}}_{\mathsf{node}} \end{pmatrix} \right\}\,,\\
\Pi &= \Pi_1 \cup \Pi_2\,.
\end{aligned}$$

For each path $\pi = \langle \nu_1, \ldots, \nu_n \rangle \in \Pi$, we select a minimal set X_π satisfying the following "selection conditions":

(a) $\nu_1, \nu_n \in X_\pi$;
(b) $\nu_2 \in X_\pi$;
(c) if $\pi \in \Pi_2$ and $\pi \not\subseteq G^{\mathcal{B}}$, for some $G \in V_{\mathsf{graph}}$, then there must exist an index i such that:
   - $1 \leq i < n$;
   - $\nu_i \in X_\pi$;
   - $\langle \nu_i, \nu_{i+1} \rangle \notin G^{\mathcal{B}}$.

Let D be the following set:

$$D = \{(\pi, \nu) \mid \pi \in \Pi \text{ and } \nu \in X_\pi\}\,.$$

We define the following equivalence relation $\sim$ over D:

$$(\pi_1, \nu) \sim (\pi_2, \mu) \qquad \Longleftrightarrow \qquad \nu = \mu \in V^{\mathcal{B}}_{\mathsf{node}}\,.$$

We let $\mathcal{A}$ be the unique **DGRA**-interpretation over $vars(\mathit{\Gamma})$ defined by letting

$$A_{\mathsf{node}} = D/\sim,$$

and

$$\begin{aligned}
a^{\mathcal{A}} &= \left[\left(\left\langle a^{\mathcal{B}}, a^{\mathcal{B}}\right\rangle, a^{\mathcal{B}}\right)\right]_{\sim}, && \text{for each } a \in V_{\mathsf{node}}, \\
x^{\mathcal{A}} &= \left\{[(\pi,\nu)]_{\sim} \mid \nu \in x^{\mathcal{B}}\right\}, && \text{for each } x \in V_{\mathsf{set}}, \\
G^{\mathcal{A}} &= \left\{([(\pi,\nu)]_{\sim}, [(\pi,\mu)]_{\sim}) \mid \begin{pmatrix} (\nu,\pi(\nu)) \in G^{\mathcal{B}} \text{ and} \\ \mu = \mathit{first}(\nu,\pi,X_{\pi}) \end{pmatrix}\right\}, && \text{for each } G \in V_{\mathsf{graph}}.
\end{aligned}$$

An example of this construction is depicted in Figure 3.

By construction, we have

$$\begin{aligned}
|A_{\mathsf{node}}| &\le |V_{\mathsf{node}}| + |\mathit{\Pi}| \cdot |V_{\mathsf{graph}}| \\
&\le |V_{\mathsf{node}}| + |V_{\mathsf{node}}|^2 \cdot |V_{\mathsf{graph}}|^2 \\
&= m + m^2 \cdot g^2 .
\end{aligned}$$

This implies that $\mathcal{A}$ is $(m^2 \cdot g^2)$-small with respect to V_{node}. We prove now that $\mathcal{A}$ satisfies all literals in $\mathit{\Gamma}$.

Literals of the form $a \neq b$. Immediate.

Literals of the form $x = \{a\}$. We have

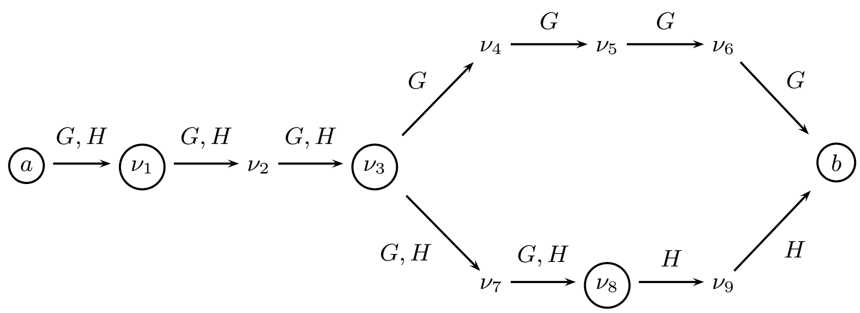
$$\begin{aligned}
x^{\mathcal{A}} &= \left\{[(\pi,\nu)]_{\sim} \mid \nu \in x^{\mathcal{B}}\right\} \\
&= \left\{\left[\left(\pi, a^{\mathcal{B}}\right)\right]_{\sim}\right\} \\
&= \left\{a^{\mathcal{A}}\right\} .
\end{aligned}$$

Literals of the form $x = y \cup z$. We have

$$\begin{aligned}
x^{\mathcal{A}} &= \left\{[(\pi,\nu)]_{\sim} \mid \nu \in x^{\mathcal{B}}\right\} \\
&= \left\{[(\pi,\nu)]_{\sim} \mid \nu \in y^{\mathcal{B}} \cup z^{\mathcal{B}}\right\} \\
&= \left\{[(\pi,\nu)]_{\sim} \mid \nu \in y^{\mathcal{B}}\right\} \cup \left\{[(\pi,\nu)]_{\sim} \mid \nu \in z^{\mathcal{B}}\right\} \\
&= y^{\mathcal{A}} \cup z^{\mathcal{A}}
\end{aligned}$$

Literals of the form $x = y \setminus z$. We have

$$\begin{aligned}
x^{\mathcal{A}} &= \left\{[(\pi,\nu)]_{\sim} \mid \nu \in x^{\mathcal{B}}\right\} \\
&= \left\{[(\pi,\nu)]_{\sim} \mid \nu \in y^{\mathcal{B}} \setminus z^{\mathcal{B}}\right\} \\
&= \left\{[(\pi,\nu)]_{\sim} \mid \nu \in y^{\mathcal{B}}\right\} \setminus \left\{[(\pi,\nu)]_{\sim} \mid \nu \in z^{\mathcal{B}}\right\} \\
&= y^{\mathcal{A}} \setminus z^{\mathcal{A}}
\end{aligned}$$



$$
\begin{aligned}
V_{\mathsf{node}} &= \{a, b\} \\
\pi_1 &= \langle a, \nu_1, \nu_2, \nu_3, \nu_4, \nu_5, \nu_6, b \rangle \\
\pi_2 &= \langle a, \nu_1, \nu_2, \nu_3, \nu_7, \nu_8, \nu_9, b \rangle \\
X_{\pi_1} &= \{a, \nu_1, \nu_3, b\} \\
X_{\pi_2} &= \{a, \nu_1, \nu_8, b\}
\end{aligned}
$$

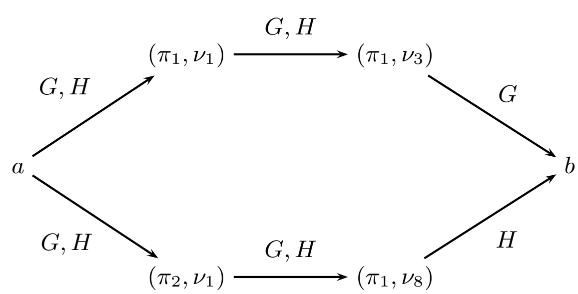


Figure 3: Example of construction of a “small” model.

Literals of the form $G = \{(a, b)\}$. We want to prove that $G^{\mathcal{A}} = \{(a^{\mathcal{A}}, b^{\mathcal{A}})\}$. Assume first that $e \in G^{\mathcal{A}}$. Then there exist $\pi \in \Pi$ and $\nu, \mu \in X_\pi$ such that:

$$\begin{gathered} e = ([(\pi,\nu)]_\sim, [(\pi,\mu)]_\sim) \\ (\nu, \pi(\nu)) \in G^{\mathcal{B}} \\ \mu = \mathit{first}(\nu, \pi, X_\pi)\,. \end{gathered}$$

It follows that $\nu = a^{\mathcal{B}}$ and $\pi(\nu) = b^{\mathcal{B}}$. Moreover, $\mu = \mathit{first}(\nu, \pi, X_\pi) = \pi(\nu) = b^{\mathcal{B}}$. Therefore $e = (a^{\mathcal{A}}, b^{\mathcal{A}})$.

Vice versa, assume that $e = (a^{\mathcal{A}}, b^{\mathcal{A}})$. We have $(a^{\mathcal{B}}, b^{\mathcal{B}}) \in G^{\mathcal{B}}$. Let $\pi = \langle \nu, \mu \rangle$ where $\nu = a^{\mathcal{B}}$ and $\mu = b^{\mathcal{B}}$. Clearly, $\pi \in \Pi_2$ and $\mathit{first}(\nu, \pi, X_\pi) = \mu$. It follows that $e \in G^{\mathcal{A}}$, as desired.

Literals of the form $G = H \cup L$. We have

$$\begin{aligned} G^{\mathcal{A}} &= \left\{ ([(\pi,\nu)]_\sim, [(\pi,\mu)]_\sim) \mid \begin{pmatrix} (\nu,\pi(\nu)) \in G^{\mathcal{B}} \text{ and} \\ \mu = \mathit{first}(\nu,\pi,X_\pi) \end{pmatrix} \right\} \\ &= \left\{ ([(\pi,\nu)]_\sim, [(\pi,\mu)]_\sim) \mid \begin{pmatrix} (\nu,\pi(\nu)) \in H^{\mathcal{B}} \cup L^{\mathcal{B}} \text{ and} \\ \mu = \mathit{first}(\nu,\pi,X_\pi) \end{pmatrix} \right\} \\ &= \left\{ ([(\pi,\nu)]_\sim, [(\pi,\mu)]_\sim) \mid \begin{pmatrix} (\nu,\pi(\nu)) \in H^{\mathcal{B}} \text{ and} \\ \mu = \mathit{first}(\nu,\pi,X_\pi) \end{pmatrix} \right\} \cup \\ &\qquad \left\{ ([(\pi,\nu)]_\sim, [(\pi,\mu)]_\sim) \mid \begin{pmatrix} (\nu,\pi(\nu)) \in L^{\mathcal{B}} \text{ and} \\ \mu = \mathit{first}(\nu,\pi,X_\pi) \end{pmatrix} \right\} \\ &= H^{\mathcal{A}} \cup L^{\mathcal{A}}\,. \end{aligned}$$

Literals of the form $G = H \setminus L$. Assume that $e \in G^{\mathcal{A}}$. Then there exist $\pi \in \Pi$ and $\nu, \mu \in X_\pi$ such that:

$$\begin{gathered} e = ([(\pi,\nu)]_\sim, [(\pi,\mu)]_\sim) \\ (\nu, \pi(\nu)) \in G^{\mathcal{B}} \\ \mu = \mathit{first}(\nu, \pi, X_\pi) \end{gathered}$$

Then $(\nu, \pi(\nu)) \in H^{\mathcal{B}} \setminus L^{\mathcal{B}}$, which implies $e \in H^{\mathcal{A}}$. Next, suppose by contradiction that $e \in L^{\mathcal{A}}$. If $\{\nu, \mu\} \setminus V^{\mathcal{B}}_{\mathsf{node}} \neq \emptyset$, then we must have $(\nu, \pi(\nu)) \in L^{\mathcal{B}}$, a contradiction. Otherwise $\nu, \mu \in V^{\mathcal{B}}_{\mathsf{node}}$. But then $\pi = \pi_{a,b,G}$ with $a^{\mathcal{B}} = \nu$ and $b^{\mathcal{B}} = \mu$. By selection condition (b) above, it follows that $\pi(\nu) \in X_\pi$. Therefore $\mu = \mathit{first}(\nu, \pi, X_\pi) = \pi(\nu)$, which implies that $\pi = \langle \nu, \mu \rangle$. Thus, $(\nu, \mu) \in L^{\mathcal{B}}$, a contradiction.

Vice versa, assume that $e \in H^{\mathcal{A}} \setminus L^{\mathcal{A}}$. Then $e \in H^{\mathcal{A}}$ and $e \notin L^{\mathcal{A}}$. It follows that there exist $\pi \in \Pi$ and $\nu, \mu \in X_\pi$ such that:

$$\begin{gathered} e = ([(\pi,\nu)]_\sim, [(\pi,\mu)]_\sim) \\ (\nu, \pi(\nu)) \in H^{\mathcal{B}} \\ \mu = \mathit{first}(\nu, \pi, X_\pi) \end{gathered}$$

We distinguish two cases: either $(\nu, \pi(\nu)) \in L^{\mathcal{B}}$ or $(\nu, \pi(\nu)) \notin L^{\mathcal{B}}$. In the former case we have $e \in L^{\mathcal{A}}$, a contradiction. In the latter case, we have $(\nu, \pi(\nu)) \in G^{\mathcal{B}}$, which implies $e \in G^{\mathcal{A}}$.

Literals of the form $G^+(a, b)$. Without loss of generality, let $\pi = \pi_{a,b,G} \in \Pi_2$. By construction, we have $\langle \alpha_1, \dots, \alpha_n \rangle \subseteq G^{\mathcal{A}}$ where

- $\alpha_i = [(\pi, \nu_i)]_\sim$;
- $\nu_1 = a^{\mathcal{A}}$;
- $\nu_n = b^{\mathcal{A}}$;
- $\nu_{i+1} = \mathit{first}(\nu_i, \pi, X_\pi)$, for $1 \leq i < n$.

Thus, $[G^+(a, b)]^{\mathcal{A}} = \mathit{true}$.

Literals of the form $\neg G^+(a, b)$. Suppose, by contradiction, that there is a path $\langle \alpha_1, \dots, \alpha_n \rangle \subseteq G^{\mathcal{A}}$ such that $\alpha_1 = a^{\mathcal{A}}$ and $\alpha_n = b^{\mathcal{A}}$. By construction, there is *exactly one* path $\pi \in \Pi_2$ such that

- $\alpha_i = [(\pi, \nu_i)]_\sim$;
- $\nu_1 = a^{\mathcal{A}}$;
- $\nu_n = b^{\mathcal{A}}$;
- $\nu_{i+1} = \mathit{first}(\nu_i, \pi, X_\pi)$, for $1 \leq i < n$.

But then, we must have $(\nu_i, \pi(\nu_i)) \in G^{\mathcal{B}}$, for each $1 \leq i < n$. However, because of selection condition (c) above, there is a j such that $(\nu_j, \pi(\nu_j)) \notin G^{\mathcal{B}}$, a contradiction.

Literals of the form $\mathit{acyclic}(G)$. Suppose, by contradiction, that there is a cycle $\langle \alpha_1, \dots, \alpha_n \rangle \subseteq G^{\mathcal{A}}$, where $\alpha_1 = \alpha_n$. By construction, without loss of generality we can assume that $\alpha_1 \in V^{\mathcal{A}}_{\mathsf{node}}$ and that $\alpha_2, \dots, \alpha_{n-1} \notin V^{\mathcal{A}}_{\mathsf{node}}$. But then, $[G^+(a, a)]^{\mathcal{A}} = \mathit{true}$, and we can obtain a contradiction by following the same reasoning employed for the literals of the form $\neg G(a, b)$. ∎

The next two theorems show how Lemma 10 entails the decidability and *NP*-completeness of the language **DGRA**.

Theorem 11 (Decidability). *The problem of deciding the* **DGRA***-satisfiability of* **DGRA***-formulae is decidable.* □

Proof. A decision procedure for **DGRA** can be obtained as follows. Without loss of generality, let Γ be a conjunction of normalized **DGRA**-literals. Nondeterministically guess a **DGRA**-interpretation $\mathcal{A}$ over $\mathit{vars}(\Gamma)$, and check whether Γ is true in $\mathcal{A}$. By Lemma 10, the number of **DGRA**-interpretations that need to be guessed is finitely bounded. Moreover, the bound can be effectively computed. ∎

Theorem 12 (Complexity). *The problem of deciding the* **DGRA***-satisfiability of* **DGRA***-formulae is NP-complete.* □

$$\frac{}{a = a}\ (E1) \qquad\qquad \frac{\begin{array}{c}\ell(a)\\ a = b\end{array}}{\ell(b)}\ (E2)$$

Note: In rule (E1), a is a node-variable already occurring in the tableau. In rule (E2), a and b are node-variables, and ℓ is a **DGRA**-literal.

Figure 4: Equality rules.

Proof. *NP*-hardness follows by the fact that the propositional calculus is embedded in the language **DGRA**. To show membership in *NP*, it is sufficient to note that:

- In nondeterministic polynomial time in the size of a **DGRA**-formula φ, we can guess a conjunction Γ of normalized **DGRA**-literals such that Γ is **DGRA**-satisfiable if and only if φ is **DGRA**-satisfiable;
- In nondeterministic polynomial time in the size of Γ, we can guess a **DGRA**-interpretation $\mathcal{A}$ such that A_{node} satisfies the cardinality requirement in Lemma 10;
- In deterministic polynomial time in the size of Γ and $\mathcal{A}$, we can check whether Γ is true in $\mathcal{A}$. ■

4 A tableau calculus for DGRA

In this section we show how the small model property can be used in order to devise a tableau-based decision procedure for **DGRA**. Our tableau calculus is based on the insight that if a **DGRA**-formula φ is true in a k-small **DGRA**-interpretation, then it is enough to generate only k fresh node-variables in order to prove the **DGRA**-satisfiability of φ.

Without loss of generality, we assume that the input of our decision procedure is a conjunction of normalized **DGRA**-literals. Thus, let Γ be a conjunction of normalized **DGRA**-literals, and let $V_\tau = vars_\tau(\Gamma)$, for each sort τ. Intuitively, a **DGRA**-*tableau* for Γ is a tree whose nodes are labeled by normalized **DGRA**-literals.

Definition 13 (DGRA-tableaux). Let Γ be a conjunction of normalized **DGRA**-literals, and let $V_\tau = vars_\tau(\Gamma)$, for each sort τ. An initial **DGRA**-tableau for Γ is a tree consisting of only one branch B whose nodes are labeled by the literals in Γ.

A **DGRA**-tableau for Γ is either an initial **DGRA**-tableau for Γ, or is obtained by applying to a **DGRA**-tableau for Γ one of the rules in Figures 4–7. □

Definition 14. A branch B of a **DGRA**-tableau is closed if at least one of the following two conditions hold:

$$\frac{\begin{array}{c} x = y \cup z \\ a \in x \end{array}}{a \in y \;\big|\; a \in z}\ (S1) \qquad \frac{\begin{array}{c} x = y \cup z \\ a \in y \end{array}}{a \in x}\ (S2) \qquad \frac{\begin{array}{c} x = y \cup z \\ a \in z \end{array}}{a \in x}\ (S3)$$

$$\frac{\begin{array}{c} x = y \setminus z \\ a \in x \end{array}}{\begin{array}{c} a \in y \\ a \notin z \end{array}}\ (S4) \qquad \frac{\begin{array}{c} x = y \setminus z \\ a \in y \end{array}}{a \in z \;\Big|\; \begin{array}{c} a \notin z \\ a \in x \end{array}}\ (S5)$$

$$\frac{\begin{array}{c} x = \{a\} \\ b \in x \end{array}}{a = b}\ (S6) \qquad \frac{x = \{a\}}{a \in x}\ (S7)$$

Figure 5: Set rules.

$$\frac{\begin{array}{c} G = H \cup L \\ G(a,b) \end{array}}{H(a,b) \;\big|\; L(a,b)}\ (G1) \qquad \frac{\begin{array}{c} G = H \cup L \\ H(a,b) \end{array}}{G(a,b)}\ (G2) \qquad \frac{\begin{array}{c} G = H \cup L \\ L(a,b) \end{array}}{G(a,b)}\ (G3)$$

$$\frac{\begin{array}{c} G = H \setminus L \\ G(a,b) \end{array}}{\begin{array}{c} H(a,b) \\ \neg L(a,b) \end{array}}\ (G4) \qquad \frac{\begin{array}{c} G = H \setminus L \\ H(a,b) \end{array}}{L(a,b) \;\Big|\; \begin{array}{c} \neg L(a,b) \\ G(a,b) \end{array}}\ (G5)$$

$$\frac{\begin{array}{c} G = \{(a,b)\} \\ G(c,d) \end{array}}{\begin{array}{c} a = c \\ b = d \end{array}}\ (G6) \qquad \frac{G = \{(a,b)\}}{G(a,b)}\ (G7)$$

Figure 6: Graph rules.

(a) B contains two complementary literals $\ell, \neg\ell$;

(b) B contains literals of the form $acyclic(G)$ and $G^+(a,a)$.

A branch which is not closed is OPEN. A **DGRA**-tableau is CLOSED if all its branches are closed; otherwise it is OPEN. □

$$\frac{G(a,b)}{G^+(a,b)}\ (R1) \qquad \frac{\begin{array}{c}G^+(a,b)\\ G^+(b,c)\end{array}}{G^+(a,c)}\ (R2) \qquad \frac{}{G(a,b) \;\big|\; \neg G(a,b)}\ (R3)$$

$$\frac{G^+(a,b)}{\begin{array}{c}G(a,w)\\ G^+(w,b)\\ w \neq c_1\\ \vdots\\ w \neq c_m\end{array}}\ (R4) \qquad \frac{}{\neg G^+(a,b)}\ (R5)$$

Note: Let Γ be a conjunction of normalized **DGRA**-literals, and let $V_\tau = vars_\tau(\Gamma)$, for each sort τ. Also, let $m = |V_{\mathsf{node}}|$ and $g = |V_{\mathsf{graph}}|$. Finally, let B be a branch of a **DGRA**-tableau form Γ.

Rule (R3) can be applied to B provided that:

(a) $a, b \in vars_{\mathsf{node}}(\mathsf{B})$.

Rule (R4) can be applied to B provided that:

(b) B is saturated with respect to rule (R3);
(c) B does not contain literals of the form $G(a, d_1), G(d_1, d_2), \ldots, G(d_{k-1}, d_k), G(d_k, b)$;
(d) $vars_{\mathsf{node}}(\mathsf{B}) = \{c_1, \ldots, c_n\}$;
(e) $|vars_{\mathsf{node}}(\mathsf{B})| < m + m^2 \cdot g^2$.

Rule (R5) can be applied to B provided that:

(a) $a, b \in vars_{\mathsf{node}}(\mathsf{B})$;
(b) B is saturated with respect to rule (R3);
(c) B does not contain literals of the form $G(a, d_1), G(d_1, d_2), \ldots, G(d_{k-1}, d_k), G(d_k, b)$;
(f) $|vars_{\mathsf{node}}(\mathsf{B})| = m + m^2 \cdot g^2$.

Intuition behind rule (R4): Conditions (b) and (c) imply the existence of a w such that $G(a, w)$ and $G^+(w, b)$. Furthermore, w must be distinct from all the node-variables already occurring in B.

Intuition behind rule (R5): Conditions (b) and (c) imply the existence of a w such that $G(a, w)$ and $G^+(w, b)$. Furthermore, w must be distinct from all the node-variables already occurring in B. But since we are looking for "small" models, condition (f) tells us that we cannot add a fresh node-variables w to B. It must necessarily follow $\neg G^+(a, b)$.

Figure 7: Reachability rules.

$$
\begin{array}{c}
1.\ \mathit{acyclic}(G) \\
2.\ \mathit{acyclic}(L) \\
3.\ G = H \setminus L \\
4.\ \ H(a,a) \\
\begin{array}{cc}
\begin{array}{c} 5.\ \ L(a,a) \\ | \\ 8.\ L^{+}(a,a) \\ \bot \end{array}
&
\begin{array}{c} 6.\ \neg L(a,a) \\ 7.\ \ G(a,a) \\ | \\ 9.\ G^{+}(a,a) \\ \bot \end{array}
\end{array}
\end{array}
$$

Figure 8: A closed **DGRA**-tableau.

Given a **DGRA**-tableau T, we can associate to it a **DGRA**-formula $\phi(\mathsf{T})$ in disjunctive normal form as follows. For each branch B of T we let

$$\phi(\mathsf{B}) = \bigwedge_{\ell \in \mathsf{B}} \ell\,,$$

where ℓ denotes a **DGRA**-literal. Then, we let

$$\phi(\mathsf{T}) = \bigvee_{\mathsf{B} \in \mathsf{T}} \phi(\mathsf{B})\,.$$

Definition 15. A **DGRA**-tableau T is SATISFIABLE if there exists a **DGRA**-interpretation $\mathcal{A}$ such that $\phi(\mathsf{T})$ is true in $\mathcal{A}$. □

Definition 16. A branch B of a **DGRA**-tableau is SATURATED if no application of any rule in Figures 4–7 can add new literals to B. A **DGRA**-tableau is SATURATED if all its branches are saturated. □

5 An example

Figure 8 shows a closed **DGRA**-tableau for the following **DGRA**-unsatisfiable conjunction of normalized **DGRA**-literals:

$$\Gamma = \left\{ \begin{array}{l} \mathit{acyclic}(G)\,, \\ \mathit{acyclic}(L)\,, \\ G = H \setminus L\,, \\ H(a,a) \end{array} \right\}.$$

The inferences in the tableau can be justified as follows:

– Nodes 5 thru 7 are obtained by means of an application of rule ($G5$).
– Node 8 is obtained by means of an application of rule (R1). The resulting branch is closed because it contains the literals $\mathit{acyclic}(L)$ and $L^{+}(a,a)$.
– Node 9 is obtained by means of an application of rule (R1). The resulting branch is closed because it contains the literals $\mathit{acyclic}(G)$ and $G^{+}(a,a)$.

6 Correctness

In this section we prove that our tableau calculus for **DGRA** is terminating, sound, and complete, and therefore it yields a decision procedure for **DGRA**. We follow standard arguments in the proofs of termination and completeness. Nonetheless, the proof of soundness is somewhat tricky, and it is based on the small model property.

6.1 Termination

Lemma 17 (Termination). *The tableau rules in Figure 4–7 are terminating.* $\square$

Proof. Let Γ be a conjunction of **DGRA**-literals, and let T be a saturated **DGRA**-tableaux. We want to show that T is finite.

Note that all rules in Figure 4–7 deduce only flat **DGRA**-literals. Furthermore, by inspecting rule (R4), it follows that the number of fresh variables that can be generated is bounded by $m^2 \cdot g^2$, where $m = |vars_{\mathsf{node}}(\Gamma)|$ and $g = |vars_{\mathsf{graph}}(\Gamma)|$.

Thus, if B is any branch of T, then B contains only flat literals constructed using a finite number of variables. It follows that the number of literals occurring in B is finite. Since all branches in T are finite, T is also finite. $\blacksquare$

Note on complexity. Let Γ be a conjunction of normalized **DGRA**-literals, and let T be a saturated **DGRA**-tableau for Γ. By inspection of the proof of Lemma 17, it follows that the size of each branch in T is polynomially bounded by the size of Γ. This implies that our tableau-based decision procedure for **DGRA** is in *NP*, confirming the complexity result of Theorem 12.

6.2 Soundness

At first glance, it seems that our tableau calculus is not sound. "How can rule (R5) be sound?", may wonder the reader. Nonetheless, the following lemma shows that all the rules of our tableau calculus are sound in the sense that they preserve **DGRA**-satisfiability with respect to k-small **DGRA**-interpretations.

Lemma 18. *Let Γ be a conjunction of normalized* **DGRA***-literals, and let $V_\tau = vars_\tau(\Gamma)$, for each sort τ. Also, let $m = |V_{\mathsf{node}}|$ and $g = |V_{\mathsf{graph}}|$. Finally, let T be a* **DGRA***-tableau, and let T' be the result of applying to T one of the rules in Figures 4–7. Assume that there exists a* **DGRA***-interpretation $\mathcal{A}$ such that:*

(α_1) *$\phi(\mathsf{T})$ is true in $\mathcal{A}$;*
(α_2) *$\mathcal{A}$ is $(m^2 \cdot g^2)$-small with respect to V_{node};*
(α_3) *$c_i^{\mathcal{A}} \neq c_j^{\mathcal{A}}$, whenever c_i is a fresh node-variable not occurring in $vars_{\mathsf{node}}(\Gamma)$, and c_j is a node-variable distinct from c_i.*

Then there exists a **DGRA***-interpretation $\mathcal{B}$ such that:*

(β_1) *$\phi(\mathsf{T}')$ is true in $\mathcal{B}$;*

(β_2) *$\mathcal{B}$ is $(m^2 \cdot g^2)$-small with respect to V_{node};*

(β_3) *$c_i^{\mathcal{B}} \neq c_j^{\mathcal{B}}$, whenever c_i is a fresh* node*-variable not occurring in $vars_{\mathsf{node}}(\Gamma)$, and c_j is a* node*-variable distinct from c_i.* □

Proof. We concentrate only on rules (R4) and (R5), since the proof goes straightforwardly for the other rules.

Concerning rule (R4), assume that the literal $G^+(a,b)$ is in B, and that conditions (b), (c), (d), and (e) in Figure 7 hold. Also, let $\mathcal{A}$ be a **DGRA**-interpretation $\mathcal{A}$ satisfying conditions (α_1), (α_2), and (α_3). By condition (b) and (c) and the fact that $(a^{\mathcal{A}}, b^{\mathcal{A}}) \in (G^{\mathcal{A}})^+$, it follows that there exists a node $\nu \in A_{\mathsf{node}}$ such that $(a^{\mathcal{A}}, \nu) \in G^{\mathcal{A}}$, $(\nu, b^{\mathcal{A}}) \in (G^{\mathcal{A}})^+$, and $\nu \neq c_i^{\mathcal{A}}$, for each $c_i \in vars_{\mathsf{node}}(\mathcal{B})$. Clearly, $\phi(\mathsf{T}')$ is true in the **DGRA**-interpretation $\mathcal{B}$ obtained from $\mathcal{A}$ by letting $w^{\mathcal{B}} = \nu$. By condition (e), $\mathcal{B}$ is $(m^2 \cdot g^2)$-small with respect to V_{node}. Moreover, condition (α_3) implies condition (β_3).

Concerning rule (R5), assume that conditions (a), (b), (c), and (f) in Figure 7 hold. Also, let $\mathcal{A}$ be a **DGRA**-interpretation $\mathcal{A}$ satisfying conditions (α_1), (α_2), and (α_3). By condition (f), it follows that $A_{\mathsf{node}} = vars_{\mathsf{node}}(\mathsf{B})$. But then, by conditions (b) and (c), we have $(a^{\mathcal{A}}, b^{\mathcal{A}}) \notin (G^{\mathcal{A}})^+$, and soundness of rule (R5) follows by letting $\mathcal{B} = \mathcal{A}$. ■

Lemma 19 (Soundness). *Let Γ be a conjunction of* **DGRA***-literals. If there exists a closed* **DGRA***-tableau for Γ, then Γ is* **DGRA***-unsatisfiable.* □

Proof. Let T be a closed **DGRA**-tableau for Γ, and suppose by contradiction that Γ is **DGRA**-satisfiable. Let $m = |vars_{\mathsf{node}}(\Gamma)|$ and $g = |vars_{\mathsf{graph}}(\Gamma)|$. By Lemmas 10 and 18, there exists an **DGRA**-interpretation $\mathcal{A}$ that is $(m^2 \cdot g^2)$-small with respect to $vars_{\mathsf{node}}(\Gamma)$, and such that $\phi(\mathsf{T})$ is true in $\mathcal{A}$. It follows that T is satisfiable. But this is a contradiction because T is closed, and closed **DGRA**-tableaux cannot be satisfiable. ■

6.3 Completeness

Lemma 20. *Let Γ be a conjunction of normalized* **DGRA***-literals, and let B be a an open and saturated branch of a* **DGRA***-tableau for Γ. Then B is satisfiable.* □

Proof. Our goal is to define a **DGRA**-interpretation $\mathcal{A}$ satisfying B.

Let $V_\tau = vars_\tau(\Gamma)$, for each sort τ. Also, let W_{node} be the set of fresh node-variables introduced by applications of rule (R4), that is, $W_{\mathsf{node}} = vars_{\mathsf{node}}(\mathsf{B}) \setminus V_{\mathsf{node}}$. Finally, let $\sim$ be the equivalence relation over $V_{\mathsf{node}} \cup W_{\mathsf{node}}$ induced by the literals of the form $a = b$ occurring in B.

We let $\mathcal{A}$ be the unique **DGRA**-interpretation over $vars(\mathsf{B})$ defined by letting

$$A_{\mathsf{node}} = (V_{\mathsf{node}} \cup W_{\mathsf{node}})/\sim\,,$$

and

$$
\begin{aligned}
a^{\mathcal{A}} &= [a]_{\sim}\,, && \text{for each } a \in V_{\mathsf{node}} \cup W_{\mathsf{node}}\,,\\
x^{\mathcal{A}} &= \{[a]_{\sim} \mid \text{the literal } a \in x \text{ is in } \mathsf{B}\}\,, && \text{for each } x \in V_{\mathsf{set}}\,,\\
G^{\mathcal{A}} &= \{([a]_{\sim},[b]_{\sim}) \mid \text{the literal } G(a,b) \text{ is in } \mathsf{B}\}\,, && \text{for each } G \in V_{\mathsf{graph}}\,.
\end{aligned}
$$

We claim that all literals occurring in B are true in $\mathcal{A}$.

Literals of the form $a = b$, $a \neq b$, $a \in x$, and $G(a,b)$. Immediate.

Literals of the form $a \notin x$. Let the literal $a \notin x$ be in B, and assume by contradiction that $a^{\mathcal{A}} \in x^{\mathcal{A}}$. Then there exists a node-variable b such that $a \sim b$, and the literal $b \in x$ is in B. By saturation with respect to the equality rules, the literal $a \in x$ is also in B, which implies that B is closed, a contradiction.

Literals of the form $\neg G(a,b)$. This case is similar to the case of literals of the form $a \notin x$.

Literals of the form $x = y \cup z$. Let the literal $x = y \cup z$ be in B. We want to prove that $x^{\mathcal{A}} = y^{\mathcal{A}} \cup z^{\mathcal{A}}$.

Assume first that $\nu \in x^{\mathcal{A}}$. Then there exists a node-variable a such that $\nu = a^{\mathcal{A}}$ and the literal $a \in x$ is in B. By saturation with respect to rule (S1), either the literal $a \in y$ is in B or the literal $a \in z$ is in B. In the former case, $\nu \in y^{\mathcal{A}}$; in the latter, $\nu \in z^{\mathcal{A}}$.

Vice versa, assume that $\nu \in y^{\mathcal{A}} \cup z^{\mathcal{A}}$ and suppose, without loss of generality, that $\nu \in y^{\mathcal{A}}$. Then there exists a node-variable a such that $\nu = a^{\mathcal{A}}$ and the literal $a \in y$ is in B. By saturation with respect to rule (S2), the literal $a \in x$ is in B. Thus, $\nu \in x^{\mathcal{A}}$.

Literals of the form $x = y \setminus z$, and $x = \{a\}$. These cases are similar to the case of literals of the form $x = y \cup z$.

Literals of the form $G = H \cup L$, $G = H \setminus L$, and $G = \{(a,b)\}$. These cases are similar to the cases of literals of the form $x = y \cup z$, $x = y \setminus z$, and $x = \{a\}$.

Literals of the form $G^{+}(a,b)$. Let the literal $G^{+}(a,b)$ be in B. If B contains literals of the form $G(a,d_1), G(d_1,d_2), \ldots, G(d_{k-1},d_k), G(d_k,b)$ then we clearly have $(a^{\mathcal{A}}, b^{\mathcal{A}}) \in (G^{\mathcal{A}})^{+}$. Otherwise, conditions (a), (b), (c), and (f) in Figure 7 hold, which implies that the literal $\neg G^{+}(a,b)$ is in B. It follows that B is closed, a contradiction.

Literals of the form $\neg G^+(a, b)$. Let the literal $\neg G^+(a, b)$ be in B, and assume by contradiction that $[G^+(a, b)]^{\mathcal{A}} = \mathit{true}$. Then there exist node-variables $c_1, \ldots, c_n$, with $n \geq 0$, such that the literals $G(a, c_1)$, $G(c_1, c_2)$, $\ldots$, $G(c_{n-1}, c_n)$, and $G(c_n, b)$ are in B. By saturation with respect to rules (R1) and (R2), the literal $G^+(a, b)$ is in B, a contradiction.

Literals of the form *acyclic*(G). Let the literal $\mathit{acyclic}(G)$ be in B, and assume by contradiction that $[\mathit{acyclic}(G)]^{\mathcal{A}} = \mathit{false}$. Then there exist node-variables $a_1, \ldots, a_n$, with $n \geq 1$, such that the literals $G(a_1, a_2)$, $G(a_2, a_3)$, $\ldots$, $G(a_{n-1}, a_n)$, and $G(a_n, a_1)$ are in B. By saturation with respect to rules (R1) and (R2), the literal $G^+(a_1, a_1)$ is in B, a contradiction. ∎

Lemma 21 (Completeness). *Let Γ be a conjunction of normalized* **DGRA**-*literals. If Γ is* **DGRA**-*unsatisfiable then there exists a closed* **DGRA**-*tableau for Γ.* □

Proof. Assume, by contradiction, that Γ has no closed **DGRA**-tableau, and let T be a saturated **DGRA**-tableau for Γ. Since Γ has no closed **DGRA**-tableau, T must contain an open and saturated branch B. By Lemma 20, B is **DGRA**-satisfiable, which implies that Γ is also **DGRA**-satisfiable, a contradiction. ∎

7 Related work

7.1 Graph theory

To our knowledge, the decision problem for graph theory was first addressed by Moser [8], who presented a decision procedure for a quantifier-free fragment of directed graph theory involving the operators of singleton graph construction, graph union, and graph intersection.

This result was extended by Cantone and Cutello [5], who proved the decidability of a more expressive quantifier-free fragment of graph theory. Cantone and Cutello's language can deal with both directed and undirected graphs, and it allows one to express the operators singleton, union, intersection, and difference, as well as some notions which are characteristic of graphs such as transitivity, completeness, cliques, independent sets, and the set of all self-loops. Cantone and Cutello's language does not deal with reachability and acyclicity.

Cantone and Cincotti [4] studied the decision problem for the language **UGRA** (*undirected graphs with reachability and acyclicity*). Intuitively, **UGRA** is the same as **DGRA**, except that it deals with undirected graphs. Unfortunately, due to a flaw in [4], it is still an open problem whether the language **UGRA** is decidable. Nonetheless, the ideas presented in [4] are very promising. Our proof of the small model property for **DGRA** is inspired by these ideas.

7.2 Static analysis and verification

Graph-based logics are of particular interest in the fields of static analysis and verification, where researchers use various abstractions based on graphs in order to represent the states of the memory of a program. We mention here four of such logics.

Benedikt, Reps, and Sagiv [2] introduced a logic of reachability expressions L_r. In this logic, one can express that it is possible to go from a certain node a to another node b by following a path that is specified by a regular expression R. For instance, in L_r the expression $a\langle(R_1 \mid R_2)^*\rangle b$ asserts that it is possible to go from node a to node b by following 0 or more edges labeled by either R_1 or R_2.

Kuncak and Rinard [7] introduced the role logic RL, a logic that has the same expressivity of first-order logic with transitive closure. They also proved that a fragment RL^2 of role logic is decidable by reducing it to the two-variable logic with counting C^2.

Resink [10] introduced the local shape logic LSL. In this logic, it is possible to constrain the multiplicities of nodes and edges in a given graph. The logic LSL is equivalent to integer linear programming.

Ranise and Zarba [9] are currently designing together a logic for linked lists LLL, with the specific goal of verifying C programs manipulating linked lists. In this logic, graphs are specified by functional arrays, with the consequence that each node of a graph has at most one outgoing edge.

7.3 Description logics

Baader [1] introduced description logic languages with transitive closure on roles. These languages are related to **DGRA** because sets of nodes are akin to concepts, roles are akin to graphs, and transitive closure of roles is akin to reachability. Therefore, we envisage a bright future in which advances in description logics will lead to advances in graph theory, and vice versa, advances in graph theory will lead to advances in description logics.

8 Conclusion

We presented a tableau-based decision procedure for the language **DGRA**, a quantifier-free fragment of directed graph theory involving the notions of reachability and acyclicity. We showed that the decidability of **DGRA** is a consequence of its *small model property*: If a formula is satisfiable, then it has a model whose cardinality is polynomial in the size of the formula. The small model property is at the heart of our tableau calculus, which can be seen as a search strategy of "small" models of the input formula.

We plan to continue this research by using (extensions of) **DGRA** in order to formally verify programs manipulating pointers. Finally, we want to study the decision problem for the language **UGRA** (*undirected graphs with reachability* *and acyclicity*) originally introduced in [4]. Although we do not know whether the language **UGRA** is decidable, we conjecture that decidability holds, at least in the case in which the acyclicity predicate is removed from the language.

Acknowledgments

This report could not have existed without the exciting discussions with the following members of the research community: Aaron R. Bradley, Gianluca Cincotti, Jean-Christophe Filliâtre, Bernd Finkbeiner, Thomas In der Rieden, Jean Goubault-Larrecq, Deepak Kapur, Yevgeny Kazakov, Dirk Leinenbach, Claude Marché, David Nowak, Silvio Ranise, Sriram Sankaranarayanan, Viorica Sofronie-Stokkermans, Uwe Waldmann, and Thomas Wies.

We are also grateful to three anonymous peers for pointing out a mistake in an earlier version of this report, and for providing instructive references to the literature.

References

1. Franz Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In John Mylopoulos and Raymond Reiter, editors, *International Joint Conference on Artificial Intelligence*, pages 446–451, 1991.
2. Michael Benedikt, Thomas W. Reps, and Shmuel Sagiv. A decidable logic for describing linked data structures. In S. Doaitse Swierstra, editor, *European Symposium on Programming*, volume 1576 of *Lecture Notes in Computer Science*, pages 2–19. Springer, 1999.
3. Rodney M. Burstall. Some techniques for proving correctness of programs which alter data structures. *Machine Intelligence*, 7:23–50, 1972.
4. Domenico Cantone and Gianluca Cincotti. The decision problem in graph theory with reachability related constructs. In Peter Baumgartner and Hantao Zhang, editors, *First-Order Theorem Proving*, Technical Report 5/2000, pages 68–90. Universität Koblenz-Landau, 2000.
5. Domenico Cantone and Vincenzo Cutello. A decidable fragment of the elementary theory of relations and some applications. In *International Symposium on Symbolic and Algebraic Computation*, pages 24–29, 1990.
6. Jean H. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*. Harper & Row, 1986.
7. Viktor Kuncak and Martin C. Rinard. Generalized records and spatial conjunction in role logic. In Roberto Giacobazzi, editor, *Static Analysis*, volume 3148 of *lncs*, pages 361–376. Springer, 2004.
8. Louise E. Moser. A decision procedure for unquantified formulas of graph theory. In Ewing L. Lusk and Ross A. Overbeek, editors, *9th International Conference on Automated Deduction*, volume 310 of *Lecture Notes in Computer Science*, pages 344–357. Springer, 1988.
9. Silvio Ranise and Calogero G. Zarba. A decidable logic for pointer programs manipulating linked lists. Unpublished, 2005.

10. Arend Rensink. Canonical graph shapes. In David A. Schmidt, editor, *European Symposium on Programming*, volume 2986 of *Lecture Notes in Computer Science*, pages 401–415. Springer, 2004.