Decidability of the Relational Syllogistic with Singular Terms

Hongkai Yin

Central European University

Abstract. We investigate the relational syllogistic logic which contains predicates of all finite arities and singular terms. We present a formal system which follows the syntax and semantics of the Quantified Argument Calculus (Quarc). We formulate a tableau calculus, and show that it has some desirable properties. With certain techniques for transforming tableaux, we prove that the satisfiability problem can be reduced to one involving only unary predicates, and hence we have a decision procedure for the logic.

1 Introduction

In the *classical syllogistic*, we have sentences such as (the formalization on the right side will be explained later):

All Greeks are mortal: $(\forall G)M$
Some Greeks are not philosophers: $(\exists G)\neg P$

It is natural to extend the language by introducing relational predicates and singular terms, while remaining faithful to the sentential structure shown above. For example,

Alice likes some philosophers: $(a, \exists P)L$
Every student likes some philosophers: $(\forall S, \exists P)L$
Some students do not like some philosophers: $(\exists S, \exists P)\neg L$
Every philosopher gives some books to some students: $(\forall P, \exists B, \exists S)G$

Similar logics and their decidability have received several treatments in the literature, sometimes under the name of the *relational syllogistic*. Some (extended) relational syllogistic logics were studied intensively in [6], but all those have only unary and binary predicates. The language in [3] contains predicates of all finite arities, but the formulas there do not allow existential quantifiers to occur within the scope of universal ones. The decidability of the relational syllogistic to be studied in this paper, where there are predicates of all finite arities and no restriction is set on the scope of quantifiers, has not been discussed in any publication so far.

The formalization we have seen in the examples above[1] follows the Quantified Argument Calculus (Quarc), which is a recently designed formal language originally introduced in [1]. Since then, several proof systems have been provided for it, including natural deduction [1], sequent calculus [5], and an axiomatic system [4].

In this paper we present a formal system for the relational syllogistic (which is essentially the fragment of Quarc where anaphora, reorder and sentential operators are excluded); we describe a tableau calculus which is sound and complete; and we sketch a decision procedure for the satisfiability of finite sets of formulas.

2 $\mathcal{RS}$: Syntax and Semantics

Definition 1. *Let* $\mathbf{R}$ *be a non-empty set of* predicates*, each of which has a fixed arity, and* $\mathbf{A}$ *be a set of* names *disjoint from* $\mathbf{R}$*. If* $\mathbf{P}$ *is the set of* unary *predicates in* $\mathbf{R}$*, and* $\mathbf{A} \cup \mathbf{P} \neq \emptyset$*, then* $L(\mathbf{R}, \mathbf{A})$ *is the* $\mathcal{RS}$ *language determined by* $\mathbf{A}$ *and* $\mathbf{R}$*.*

Definition 2. *The* arguments *of* $L(\mathbf{R}, \mathbf{A})$ *are defined as follows:*

- *Every name* $a \in \mathbf{A}$ *is a* singular *argument.*
- *If* $P \in \mathbf{R}$ *is a unary predicate, then* $\forall P$ *and* $\exists P$ *are (universally and particularly)* quantified *arguments.*

Definition 3. *Let* $\alpha_1, \ldots, \alpha_n$ *be (singular or quantified) arguments of* $L(\mathbf{R}, \mathbf{A})$*, and* R *an* n*-ary predicate in* $\mathbf{R}$ *(*$n \geq 1$*). Then* $\alpha_1 ... \alpha_n R$ *and* $\alpha_1 ... \alpha_n \neg R$ *are* formulas *of* $L(\mathbf{R}, \mathbf{A})$*.*

For convenience, we will often call $\alpha_1 ... \alpha_n R$ or $\alpha_1 ... \alpha_n \neg R$ a 'unary' formula if R is a unary predicate, and a 'multi-ary' formula if R is of arity $n > 1$. For example, if a is a name, P, Q, S are unary predicates, R is a binary predicate, and T is a ternary predicate, then aP, $\forall SP$ are unary formulas, and $a\exists PR$, $\forall P\exists SR$, $\exists S\exists P\neg R$, $\forall P\exists QaT$, $\forall P\exists Q\exists S\neg T$ are multi-ary formulas.

We write $\phi(\forall P)$ or $\phi(\exists P)$ to indicate that $\forall P$ or $\exists P$ is the leftmost occurrence of quantified arguments in the formula, and in such cases $\phi(a)$, $\phi(b)$, ... denote the results of replacing that occurrence by the singular arguments a, b, ... respectively. For example, if $\phi(\exists P)$ is $\exists PP$ then $\phi(a)$ is aP; if $\phi(\forall P)$ is $a\forall P\forall PT$ then $\phi(a)$ is $aa\forall PT$ and $\phi(b)$ is $ab\forall PT$.

Definition 4. *A* model *for the* $\mathcal{RS}$ *language* $L(\mathbf{R}, \mathbf{A})$ *is a pair,* $M = \langle D, I \rangle$*, where* D *is a non-empty set, called the* domain*, and* I *is a mapping, called the* interpretation*, such that:*

- *If* a *is a name,* $I(a) \in D$

[1] The parentheses and comma used there will be dropped in the formal language to be defined in this paper.

– *If P is a unary predicate, $I(P) \subseteq D$ and $I(P) \neq \emptyset$* [2]
– *If R is a predicate of arity $n > 1$, $I(R) \subseteq D^n$*

For any model M, we write D^* for the union of the interpretation of all unary predicates in the language: $D^* = \bigcup_{P \in \mathbf{P}} I(P)$, where $\mathbf{P}$ is the set of unary predicates in $\mathbf{R}$.

Definition 5. *The* satisfaction *of formulas by a model is defined as follows. We extend the language (and also the interpretation) so that every instance of every unary predicate has a name: for each $d \in D^*$, we add a new name k_d such that $I(k_d) = d$. Then:*

– $M \vDash a_1 \ldots a_n R$ *iff* $\langle I(a_1), \ldots, I(a_n)\rangle \in I(R)$
– $M \vDash a_1 \ldots a_n \neg R$ *iff* $M \nvDash a_1 \ldots a_n R$
– $M \vDash \phi(\forall P)$ *iff* $M \vDash \phi(k_d)$ *for all* $d \in I(P)$
– $M \vDash \phi(\exists P)$ *iff* $M \vDash \phi(k_d)$ *for some* $d \in I(P)$

Definition 6. *A set of formulas is* satisfiable *iff there is a model which satisfies all of its members.*

We write $-\phi$ for the formula resulting from switching each quantifier in ϕ and adding (removing) the negation symbol to (from) ϕ. For example, if ϕ is aP then $-\phi$ is $a\neg P$; if ϕ is $\forall P_1 \exists P_2 R$ then $-\phi$ is $\exists P_1 \forall P_2 \neg R$; and if ϕ is $\exists P_1 \forall P_2 \neg R$ then $-\phi$ is $\forall P_1 \exists P_2 R$.

Proposition 1. *Let ϕ be a formula. Then: for any model M, $M \vDash \phi$ iff $M \nvDash -\phi$.*

Proof. By induction on the number of the occurrences of quantifiers. □

We thus call $-\phi$ the *contradictory* of ϕ, or say that ϕ and $-\phi$ are contradictory. Note that although there is no sentential negation in the language, $-\phi$ functions semantically as the negation of ϕ. This fact is essential to the tableau calculus in the next section.

3 Tableaux

We describe a tableau calculus for the logic defined above. Given a finite set of formulas, we start with a sequence of all of its members without repetition, and then the sequence, which can be called the initial tableau, shall be expanded step-by-step according to the tableau expansion rules (see below).

Definition 7. *Let $\{\phi_1, \ldots, \phi_n\}$ be a finite set of $\mathcal{RS}$ formulas.*

[2] The non-emptiness here is a stipulation to preserve the entailment of subalterns from their superalterns in Aristotelian logic. It is also adopted by some works on two-valued Quarc, such as [1] and [7]. But please note that in some three-valued logics that also validate such entailment, this stipulation, as well as the non-emptiness of the whole domain, is removed. See [2] and [9].

1. *The sequence of formulas $\phi_1, \dots, \phi_n$ is an (initial) tableau for $\{\phi_1, \dots, \phi_n\}$. (If later the tableau is expanded, we also call $\phi_1, \dots, \phi_n$ the initial part of the expanded tableau.)*
2. *If $\mathbf{T}$ is a tableau for $\{\phi_1, \dots, \phi_n\}$ and $\mathbf{T}^*$ results from $\mathbf{T}$ by an application of a tableau expansion rule (see Definition 8), then $\mathbf{T}^*$ is a tableau for $\{\phi_1, \dots, \phi_n\}$.*

We say that a tableau is *completed* if no tableau expansion rule can be applied to formulas in it. We say that a tableau is *closed* if there is a formula ϕ such that both ϕ and $-\phi$ are in the tableau; otherwise we say it is *open*.

Definition 8. *Tableau Expansion Rules:*[3]

R1 *If $\phi(\exists P)$ is in a tableau* and has no successor (see below)*, add cP and $\phi(c)$ to the right end of the tableau for a new name c. (In case $\phi(\exists P)$ is $\exists PP$, add only one occurrence of cP.)*
R2 *If $\phi(\forall P)$ is in a tableau but aP is not for any name a, add cP and $\phi(c)$ to the right end of the tableau for a new name c. (In case $\phi(\forall P)$ is $\forall PP$, add only one occurrence of cP.)*
R3 *For any name a, if $\phi(\forall P)$ and aP are in a tableau but $\phi(a)$ is not, add $\phi(a)$ to the right end of the tableau.*

When R1 is applied, $\phi(\exists P)$ is the *predecessor* of cP and cP is a *successor* of $\phi(\exists P)$; also, $\phi(\exists P)$ is the predecessor of $\phi(c)$ and $\phi(c)$ is a successor of $\phi(\exists P)$. Similarly, when R2 is applied, $\phi(\forall P)$ is the predecessor of cP and cP is a successor of $\phi(\forall P)$; also, $\phi(\forall P)$ is the predecessor of $\phi(c)$ and $\phi(c)$ is a successor of $\phi(\forall P)$. When R3 is applied, $\phi(\forall P)$ is a predecessor of $\phi(a)$ and $\phi(a)$ is a successor of $\phi(\forall P)$; also, aP is a predecessor of $\phi(a)$ and $\phi(a)$ is a successor of aP.

Note that a formula cannot occur more than once in a tableau. This is because R1 and R2 add only formulas with a new name, and R3 always adds a formula that is not yet in a tableau.

Example 1. A completed and closed tableau for $\{\forall P_1P_2, \forall P_2P_3, \exists P_1\neg P_3\}$.

$$\forall P_1P_2, \forall P_2P_3, \exists P_1\neg P_3, aP_1, a\neg P_3, aP_2, aP_3$$

In this tableau, aP_1 and $a\neg P_3$ are added by applying R1 to $\exists P_1\neg P_3$; aP_2 is the result of applying R3 to $\forall P_1P_2$ and aP_1; aP_3 is the result of applying R3 to $\forall P_2P_3$ and aP_2. It is closed because $a\neg P_3$ and aP_3 are contradictory.

Example 2. A completed and closed tableau for $\{\exists P_1P_2, \forall P_2\forall P_3R, \forall P_1\forall P_3\neg R\}$.

$$\exists P_1P_2, \forall P_2\forall P_3R, \forall P_1\forall P_3\neg R, aP_1, aP_2, a\forall P_3R, bP_3, abR, a\forall P_3\neg R, ab\neg R$$

In this tableau, aP_1 and aP_2 are added by applying R1 to $\exists P_1P_2$; $a\forall P_3R$ is the result of applying R3 to $\forall P_2\forall P_3R$ and aP_2; bP_3 and abR are added by applying R2 to $a\forall P_3R$; $a\forall P_3\neg R$ is added by applying R3 to $\forall P_1\forall P_3\neg R$ and aP_1; $ab\neg R$ is added by applying R3 to $a\forall P_3\neg R$ and bP_3. It is closed by abR and $ab\neg R$.

[3] There is no ‘branching’ rule in our tableau calculus, which is why we define a tableau as a sequence of formulas, and also why we do not talk about ‘branches’.

Example 3. A completed and infinite tableau for $\forall P \exists PR$.

$$\forall P \exists PR, aP, a\exists PR, bP, abR, b\exists PR, cP, bcR, c\exists PR, \ldots$$

We first apply R2 to $\forall P \exists PR$, adding $aP, a\exists PR$. Then we apply R1 to $a\exists PR$, adding bP, abR. Then, as bP is in the tableau but $b\exists PR$ is not, we apply R3 to $\forall P \exists PR$ and bP, adding $b\exists PR$. Then we apply R1 to $b\exists PR$, adding cP, bcR. Then, again, as cP is in the tableau but $c\exists PR$ is not, we apply R3 to $\forall P \exists PR$ and cP, adding $c\exists PR$ ... As this process goes on to infinity, we indeed have a completed tableau. We also notice that the tableau is open.

So far the tableau expansion is nondeterministic, because in each step there may be more than one possible application of expansion rules. Also, it is not obvious that every tableau can be expanded into a completed one, given that completed ones can be infinite. Hence, some restrictions should be specified for the application of expansion rules. But since there are obvious ones that can be adapted in [8], we will save ourselves the work here.

We proceed to introduce the terms 'ancestor' and 'descendant'.

Definition 9. *Let ϕ, ψ, χ be formulas in a tableau. Then:*

1. *ϕ is an* ancestor *of itself;*
2. *If ϕ is a predecessor of ψ, then ϕ is an ancestor of ψ, $(*)$ provided that ϕ is a unary formula or ψ is a multi-ary formula;*
3. *If ϕ is an ancestor of ψ, and ψ is an ancestor of χ, then ϕ is an ancestor of χ.*

We say that ψ is a descendant *of ϕ if ϕ is an ancestor of ψ.*

In the following we state a few results that will be useful in the next section. Many details are omitted in the proofs due to space limitations.

Proposition 2. *A unary formula has only unary ancestors in a tableau.*

Proof. By the last definition, especially the proviso $(*)$. □

Proposition 3. *A multi-ary formula in a tableau has one and only one multi-ary ancestor in the initial tableau.*

Proof. By the extension rules, a multi-ary formula has at most one multi-ary predecessor. □

Proposition 4. *If a finite set of formulas has a closed tableau, then it has a tableau which is closed by two quantifier-free formulas.*

Proof. If a tableau is closed by two formulas containing quantifiers, it can always be expanded into one which is closed (also) by two respective quantifier-free descendants of them. □

Proposition 5. *Let Γ be a finite set of formulas. Γ is satisfiable iff there is no closed tableau for Γ.*

Proof. The 'only-if' direction is *soundness*. We first show that, given a satisfiable tableau (i.e. a tableau in which the set of formulas is satisfiable), the result of any application of any tableau expansion rule is also a satisfiable tableau. Hence, if Γ has a closed tableau, which is trivially unsatisfiable, then Γ is unsatisfiable.

The 'if' direction is *completeness*. We first prove that a completed open tableau induces a model which satisfies the set of formulas in it. (Note that the interpretation of unary predicates must be nonempty according to Definition 4.) Then, if Γ is unsatisfiable, a completed tableau for it must be closed. This, together with soundness, also shows that if Γ has a completed open tableau, then it cannot have any closed one. □

Proposition 6. *Let Γ be a finite set of formulas, and aP a formula where a is new to Γ. Then Γ has a closed tableau iff so does $\Gamma \cup \{aP\}$.*

Proof. Since a does not occur among formulas in Γ, we have that Γ is satisfiable iff so is $\Gamma \cup \{aP\}$, and then, by soundness and completeness, Γ has a closed tableau iff so does $\Gamma \cup \{aP\}$. □

Proposition 7. *The tableau calculus is terminating for finite sets of unary formulas (and is thus a decision procedure in that case).*

Proof. It is an easy observation that, in any procedure towards a completed tableau for a finite set of unary formulas, the number of new names introduced by R1 and R2 is bounded by the number of the occurrences of quantified arguments in the set. Hence, every completed tableau for it is finite, and thus can be achieved in a finite number of steps. □

In general, however, completed open tableaux may be infinite and the tableau calculus serves only as a semi-decision procedure.

4 Decidability

In this section we prove the decidability of the satisfiability of finite sets of $\mathcal{RS}$ formulas.

Proposition 8. *Let Γ be a finite set of formulas, and Δ be the set of all and only* unary *formulas in Γ. Then:*

1. *if Γ has a tableau closed by two unary formulas, Δ has a closed tableau;*
2. *if Γ has a tableau closed by two multi-ary formulas, there are two formulas $\alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R \in \Gamma$ $(n > 1)$ such that $\Delta \cup \{\alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R\}$ has a closed tableau.*

Proof. Suppose $\mathbf{T}$ is a closed tableau for Γ, and by Proposition 4 we can assume that it is closed by a formula $a_1 \ldots a_n R$ and its contradictory $a_1 \ldots a_n \neg R$ $(n \geq 1)$. We construct a sequence of formulas $\mathbf{S}$ by eliminating any formula in $\mathbf{T}$ that is not an ancestor of $a_1 \ldots a_n R$ or $a_1 \ldots a_n \neg R$, while preserving the relative order of the remaining ones. We write Γ' for the set of formulas from Γ that are left in **S**.

Note that **S** is not necessarily a tableau for Γ', for, by the proviso $(*)$ in Definition 9, there may be some unary formulas, say $c_1P_1, c_2P_2, \ldots, c_kP_k$, that are not in Γ and have no ancestors preceding each of them in **T**. However, if we move them to the left end of **S**, taking them as formulas in the initial tableau, we get a closed tableau for $\Gamma' \cup \{c_1P_1, c_2P_2, \ldots, c_kP_k\}$. Also, each such c_iP_i, as it has a multi-ary predecessor in **T**, was added by an application of either R1 or R2, so $c_1, c_2, \ldots, c_k$ are new to Γ, and hence new to Γ'. Then, by Proposition 6, we have that Γ' has a closed tableau.

In case $n = 1$, i.e. **T** is closed by a_1R and $a_1\neg R$, by Proposition 2, Γ' contains only unary formulas. Since Δ is the set of unary formulas in Γ, we have $\Gamma' \subseteq \Delta$, and hence Δ also has a closed tableau. In case $n > 1$, by Proposition 3, Γ' contains, apart from unary formulas, only two multi-ary formulas: $\alpha_1 \ldots \alpha_n R$ and $\beta_1 \ldots \beta_n \neg R$, which are the multi-ary ancestors of $a_1 \ldots a_n R$ and $a_1 \ldots a_n \neg R$ respectively. In this case, since $\Gamma' \subseteq \Delta \cup \{\alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R\}$, the latter also has a closed tableau. □

Moreover, when Γ has a tableau closed by two multi-ary formulas, and Δ does not have a closed tableau, some further conditions must be met by the arguments in those two multi-ary formulas. But first we need to introduce the following definition.

Definition 10. *Let α, β be arguments, and Δ a finite set of unary formulas that does not have a closed tableau. We say that α and β are* unifiable *on Δ, written $\alpha \sim_\Delta \beta$, if one of the following conditions is satisfied:*

- *α and β are the same singular argument*
- *α is a, β is $\forall P$, and $\Delta, a\neg P$ has a closed tableau*[4]
- *α is $\forall P$, β is a, and $\Delta, a\neg P$ has a closed tableau*
- *α is $\forall P_1$, β is $\forall P_2$, and $\Delta, \forall P_1 \neg P_2$ has a closed tableau*
- *α is $\exists P_1$, β is $\forall P_2$, and $\Delta, \exists P_1 \neg P_2$ has a closed tableau*
- *α is $\forall P_1$, β is $\exists P_2$, and $\Delta, \exists P_2 \neg P_1$ has a closed tableau*

There is obviously an algorithm which, given two arguments and a finite set of unary formulas that has no closed tableau, determines whether those two arguments are unifiable on the set in a finite amount of time. If both arguments are singular, we only need to check whether they are the same; if at least one of them is a universally quantified argument, we construct tableaux for a finite set of unary formulas (as specified in the definition), which, by Proposition 7, will terminate; and in all other cases, we conclude immediately that they are not unifiable.

Proposition 9. *Let Δ be a finite set of unary formulas that does not have a closed tableau. If $\alpha_i \sim_\Delta \beta_i$ for every $1 \leq i \leq n$, then $\Delta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$ has a closed tableau.*

[4] We write $\Delta, a\neg P$ to mean $\Delta \cup \{a\neg P\}$. Similarly below.

Proof. For each $1 \leq i \leq n$, since $\alpha_i \sim_\Delta \beta_i$, α_i and β_i must satisfy one of the conditions specified in the last definition. In each case we define a formula θ_i accordingly:

1. α_i and β_i are the same singular argument. We define θ_i to be a logical truth, e.g. $\forall PP$.
2. α_i is a, β_i is $\forall P$, and $\Delta, a\neg P$ has a closed tableau. We define θ_i to be aP.
3. α_i is $\forall P$, β_i is a, and $\Delta, a\neg P$ has a closed tableau. We define θ_i to be aP.
4. α_i is $\forall P_1$, β_i is $\forall P_2$, and $\Delta, \forall P_1 \neg P_2$ has a closed tableau. We define θ_i to be $\exists P_1 P_2$.
5. α_i is $\exists P_1$, β_i is $\forall P_2$, and $\Delta, \exists P_1 \neg P_2$ has a closed tableau. We define θ_i to be $\forall P_1 P_2$.
6. α_i is $\forall P_1$, β_i is $\exists P_2$, and $\Delta, \exists P_2 \neg P_1$ has a closed tableau. We define θ_i to be $\forall P_2 P_1$.

For each $1 \leq i \leq n$, no matter which case above holds for α_i and β_i, we have that $\Delta, -\theta_i$ has a closed tableau. Then, by Proposition 5, $\Delta, -\theta_i$ is unsatisfiable. Then, for every model, if it satisfies Δ then it also satisfies θ_i.

We define Θ to be $\{\theta_1, \ldots, \theta_n\}$. Then, for every model, if it satisfies Δ it must also satisfy Θ. Then $\Delta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$ is satisfiable only if so is $\Theta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$. By Proposition 5, if $\Theta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$ has a closed tableau then so does $\Delta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$.

Finally, it is an easy observation that $\Theta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$ has a closed tableau, so $\Delta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$ has a closed tableau. □

Proposition 10. *Let Δ be a finite set of unary formulas that does not have a closed tableau. If $\Delta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$ $(n > 1)$ has a closed tableau, then $\alpha_i \sim_\Delta \beta_i$ for every $1 \leq i \leq n$.*

Proof. Suppose $\mathbf{T}$ is a closed tableau for $\Delta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$. Then, since Δ has no closed tableau, and $\Delta, \alpha_1 \ldots \alpha_n R, \beta_1 \ldots \beta_n \neg R$ are the only two multi-ary formulas in the initial part of $\mathbf{T}$, by Propositions 3, 4, and 8, we can assume that it is closed by a formula $a_1 \ldots a_n R$ and its contradictory $a_1 \ldots a_n \neg R$, such that $\alpha_1 \ldots \alpha_n R$ is an ancestor of $a_1 \ldots a_n R$ and $\beta_1 \ldots \beta_n \neg R$ is an ancestor of $a_1 \ldots a_n \neg R$. For each $1 \leq i \leq n$, we consider the following cases:

Case 1. α_i is a_i and β_i is b_i. If a_i is the same as b_i, trivially $\alpha_i \sim_\Delta \beta_i$; if they are different, no multi-ary descendant of $\beta_1 \ldots \beta_n \neg R$ will be a contradictory of any multi-ary descendant of $\alpha_1 \ldots, \alpha_n R$, but this contradicts our assumption.

Case 2. α_i is a_i, β_i is $\exists P$. In this case, as R1 introduces a name different from a_i, no multi-ary descendant of $\beta_1 \ldots \beta_n \neg R$ will be a contradictory of any multi-ary descendant of $\alpha_1 \ldots, \alpha_n R$, but this contradicts our assumption.

Case 3. α_i is $\exists P_1$, β_i is $\exists P_2$. Similar to the last one, this case contradicts our assumption.

Case 4. α_i is a_i, β_i is $\forall P$. Then, since $\forall P$ is later replaced by a_i in an application of R3, $a_i P$ is in **T**. (Notice that a_i is not among the names introduced by R1 or R2.) If we add $a_i \neg P$ to the initial part of **T**, the result is a tableau for $\Delta, \alpha_1 \dots \alpha_n R, \beta_1 \dots \beta_n \neg R, a_i \neg P$. Also, as it is closed by $a_i P$ and $a_i \neg P$, by Proposition 8 we know that $\Delta, a_i \neg P$ has a closed tableau. Hence, $\alpha_i \sim_\Delta \beta_i$.

Case 5. α_i is $\forall P_1$, β_i is $\forall P_2$. Then, since both $\forall P_1$ and $\forall P_2$ are later replaced by a_i, we know that $a_i P_1$ and $a_i P_2$ are in **T**. If we add $\forall P_1 \neg P_2$ to the initial part of **T**, the result can be easily expanded into a closed tableau for $\Delta, \alpha_1 \dots \alpha_n R, \beta_1 \dots \beta_n \neg R, \forall P_1 \neg P_2$. (We get $a_i \neg P_2$ from $\forall P_1 \neg P_2$ and $a_i P_1$, and then it is closed by $a_i P_2$ and $a_i \neg P_2$.) Also, as it is closed by two contradictory unary formulas, by Proposition 8 we have that $\Delta, \forall P_1 \neg P_2$ has a closed tableau. Hence, $\alpha_i \sim_\Delta \beta_i$.

Case 6. α_i is $\exists P_1$, β_i is $\forall P_2$. Then $a_i P_1$ and $a_i P_2$ are in **T**. Also, since $\exists P_1$ is replaced by a_i in an application of R1, $a_i P$ is the leftmost formula containing a_i in **T**. We remove all formulas from **T** that are not ancestors of $a_i P_1$ or $a_i P_2$. We write **S** for the resulting sequence of formulas, and Δ' for the set of formulas from Δ that are left in **S**. (Notice that, by Proposition 2, Δ' does not contain multi-ary formulas.) As the predecessor of $a_i P_1$ in **T** is multi-ary, **S** is not a tableau for Δ'. We add $\exists P_1 \neg P_2$ to the left end of **S**, and add $a_i \neg P_2$ right next to $a_i P_1$, and write **S**$'$ for the result. Then, since $a_i P_1$ is the leftmost formula containing a_i in **S**$'$, $a_i P_1$ and $a_i \neg P_2$ can be seen as the result of applying R1 to $\exists P_1 \neg P_2$, hence **S**$'$ is a tableau for $\Delta', \exists P_1 \neg P_2$. Also, **S**$'$ is closed because it contains both $a_i P_2$ and $a_i \neg P_2$. Hence, $\Delta, \exists P_1 \neg P_2$ has a closed tableau, and $\alpha_i \sim_\Delta \beta_i$.

The other cases are similar. □

Proposition 11. *Let Γ be a finite set of formulas, and Δ the set of all and only unary formulas in Γ. Then Γ has a closed tableau if and only if Δ has a closed tableau,* or, *if Δ does not have a closed tableau, there are two formulas $\alpha_1 \dots \alpha_n R, \beta_1 \dots \beta_n \neg R \in \Gamma$ ($n > 1$) such that $\alpha_i \sim_\Delta \beta_i$ for every $1 \leq i \leq n$.*

Proof. The 'if' direction is straightforward by Proposition 9. The 'only-if' direction is a consequence of Propositions 8 and 10. □

Proposition 12. *The satisfiability problem for $\mathcal{RS}$ is decidable.*

Proof. Let Γ be a finite set of formulas, and Δ the set of all and only unary formulas in Γ. A decision procedure is sketched as follows. We first decide the satisfiability of Δ by tableau. If it is unsatisfiable, we conclude that Γ is unsatisfiable. If satisfiable, we move on to look for pairs of multi-ary formulas in Γ of the form $(\alpha_1 \dots \alpha_n R, \beta_1 \dots \beta_n \neg R)$. If there aren't such pairs, we conclude that Γ is satisfiable. If there are, then, for each such pair, we run the procedure indicated by Definition 10 to decide whether $a_i \sim_\Delta b_i$ for every $1 \leq i \leq n$. If we find a positive case then Γ is unsatisfiable; if not, Γ is satisfiable. □

5 Conclusion

In this paper we proved that the relational syllogistic defined in section 2 is a decidable logic. This was done mainly by showing that a closed tableau can always be transformed into one which is also closed and contains no predicates of arity $n > 1$.

References

1. Ben-Yami, H.: The Quantified Argument Calculus. The Review of Symbolic Logic **7**(1), 120–146 (2014). https://doi.org/10.1017/s1755020313000373
2. Lanzet, R.: A Three-Valued Quantified Argument Calculus: Domain-Free Model-Theory, Completeness, and Embedding of FOL. The Review of Symbolic Logic **10**(3), 549–582 (2017). https://doi.org/10.1017/s1755020317000053
3. Nishihara, N., Morita, K., Iwata, S.: An Extended Syllogistic System with Verbs and Proper Nouns, and Its Completeness Proof. Systems and Computers in Japan **21**(1), 96–111 (1990). https://doi.org/10.1002/scj.4690210110
4. Pascucci, M.: An Axiomatic Approach to the Quantified Argument Calculus. Erkenntnis (2022). https://doi.org/10.1007/s10670-022-00519-9
5. Pavlović, E., Gratzl, N.: Proof-Theoretic Analysis of the Quantified Argument Calculus. The Review of Symbolic Logic **12**(4), 607–636 (2019). https://doi.org/10.1017/s1755020318000114
6. Pratt-Hartmann, I., Moss, L.S.: Logics for the Relational Syllogistic. The Review of Symbolic Logic **2**(4), 647–683 (2009). https://doi.org/10.1017/s1755020309990086
7. Raab, J.: Aristotle, Logic, and QUARC. History and Philosophy of Logic **39**(4), 305–340 (2018). https://doi.org/10.1080/01445340.2018.1467198
8. Smullyan, R.M.: First-Order Logic. Springer Berlin Heidelberg (1968). https://doi.org/10.1007/978-3-642-86718-7
9. Yin, H., Ben-Yami, H.: The Quantified Argument Calculus with Two- and Three-valued Truth-valuational Semantics. Studia Logica **111**(2), 281–320 (2022). https://doi.org/10.1007/s11225-022-10022-5