A Tableaux System for $\mathcal{S}5_{DY}$ - Soundness, Completeness and Termination Argument

Luiz C. F. Fernandez
PESC/Coppe
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro - RJ
lcfernandez@cos.ufrj.br

Mario R. F. Benevides
Instituto de Computação
Universidade Federal Fluminense (UFF)
Niterói - RJ
mario@ic.uff.br

This work is an appendix to the WBL'23 submission *A Tableaux System for Dolev-Yao Multi-Agent Epistemic Logic*, a theorem prover for the Dolev-Yao Multi-Agent Epistemic Logic. Our prover is based on tableaux method and we prove it sound and complete. Finally, we provide a termination argument for our method.

1 Soundness

The soundness proof for our method is inspired by Costa [1]. First, we need some definitions:

Definition 1.1. *Let Γ be a set of formulae:*

1. *we denote $s \Vdash \Gamma$ to represent $s \Vdash \alpha$, for all $\alpha \in \Gamma$;*
2. *we say Γ is satisfiable if there exists a model $\mathcal{M}$ and some possible state $s \in S$ such that $s \Vdash \Gamma$;*
3. *a tableau branch is satisfiable if the set of all its formulae is satisfiable. A tableau is satisfiable if at least one branch is satisfiable.*

Lemma 1.2. *The rules of tableaux method preserve satisfiability. That is, if a tableau $\mathcal{T}$ is satisfiable then the tableau resulting from the application of a rule to $\mathcal{T}$ is satisfiable.*

Proof. Let $\mathcal{T}$ be a satisfiable tableau. By property 3 of Definition 1.1, $\mathcal{T}$ has at least one satisfiable branch, although it could have unsatisfiable ones. So, or the rule is applied to a satisfiable branch or to an unsatisfiable one.

First case: if the rule is applied to an unsatisfiable branch, each originally satisfiable branch remains unchanged. Therefore, the tableau resulting from the application of a rule is satisfiable.

Second case: if the rule is applied to a satisfiable branch θ, which consists of a set of formulae Γ and some specific formulae γ and δ which the rule is applied. As θ is satisfiable, by property 2 of Definition 1.1, there exists a model $\mathcal{M}$ and a possible state $s \in S$ such that $s \Vdash \Gamma$, in particular, $s \Vdash \gamma$ and $s \Vdash \delta$. Let's θ' be the new branch obtained by the application of an inference rule to θ. We have the following cases for each possible structure of γ and/or δ:

- for γ or δ of type $\neg\neg\alpha$, $\alpha \wedge \beta$, $\neg(\alpha \vee \beta)$, $\neg(\alpha \rightarrow \beta)$, $\alpha \vee \beta$, $\neg(\alpha \wedge \beta)$, $\alpha \rightarrow \beta$, $K_a\alpha$ or $\neg K_a\alpha$, the proof can be found in tableaux for modal logics literature [1, 2].
- R_{Dec}: for γ of type m and for δ of type k, where $m, k \in \Gamma$, since $s \Vdash m$ and $s \Vdash k$, that is, $s \Vdash m \wedge k$, by the soundness of axiom 6 of Section 3.1 of the original submission, we have $s \Vdash \{m\}_k$. Therefore, θ' is satisfiable.

Submitted to:
WBL'23

- $R^{\neg}_{Enc}$: for γ or δ of type $\neg\{m\}_k$ and $s \Vdash \neg\{m\}_k$. By the contrapositive of axiom 6 of Section 3.2 of the original submission and its soundness, we have $s \Vdash \neg(m \wedge k)$ and also $s \Vdash \neg m \vee \neg k$. Suppose $s \Vdash \neg m$, then θ' is satisfiable. Suppose $s \Vdash \neg k$, then θ' is also satisfiable. Therefore, θ' is satisfiable.
- the cases for rules R_{Pair} and $R^{\neg}_{Pair}$ are analogous to the cases for rules R_{Dec} and $R^{\neg}_{Enc}$, respectively, but using axiom 8 of Section 3.1 of the original submission.

□

The soundness of our tableaux method follows straightforward from the above lemma. If a formula $\neg\alpha$ has a closed tableaux, then it is unsatisfiable. Therefore α must be a valid formula.

2 Completeness

Smullyan [4] proved the completeness of tableaux method for classical logic based on the construction of a completed tableaux and showing that when we cannot build a closed tableau for a formula $\neg\alpha$, we have what is necessary to build a counter-model for α, therefore, α is not valid. Then, Fitting [2] extended this approach by adding the notion of *prefixed tableaux*, with the definition of a *completed tableau* and proving that if a formula α is valid, then every completed tableau for $\neg\alpha$ is closed. The completeness proof provided by Costa [1] is inspired by this approach and is also the base for the proof below. Let's begin with some definitions:

Definition 2.1. *Formulae of the form $X \wedge Y$, $\neg(X \vee Y)$, $\neg(X \to Y)$, $\neg\neg X$, (m,n) or occurrences of m and k are called type-α formulae, while every formulae of the form $X \vee Y$, $\neg(X \wedge Y)$, $X \to Y$, $\neg(m,n)$ or $\neg\{m\}_k$ are called type-β formulae. The components α_1 and α_2 from a type-α formula and the components β_1 and β_2 from a type-β formula are given in the tables bellow:*

α	α_1	α_2
$X \wedge Y$	X	Y
$\neg(X \vee Y)$	$\neg X$	$\neg Y$
$\neg(X \to Y)$	X	$\neg Y$
$\neg\neg X$	X	X
(m,n)	m	n
m	$\{m\}_k$	$\{m\}_k$
k		

Table 1: Components of a type-α formula

β	β_1	β_2
$X \vee Y$	X	Y
$\neg(X \wedge Y)$	$\neg X$	$\neg Y$
$X \to Y$	$\neg X$	Y
$\neg(m,n)$	$\neg m$	$\neg n$
$\neg\{m\}_k$	$\neg m$	$\neg k$

Table 2: Components of a type-β formula

Definition 2.2. *A branch θ of a tableau σ is called* complete *if it satisfies the following conditions (where Σ is a set of formulae of θ and γ a specific formula):*

1. *if $(\sigma,\alpha) \in \Sigma$, then $(\sigma,\alpha_1) \in \Sigma$ and $(\sigma,\alpha_2) \in \Sigma$;*
2. *if $(\sigma,\beta) \in \Sigma$, then $(\sigma,\beta_1) \in \Sigma$ or $(\sigma,\beta_2) \in \Sigma$;*
3. *if $(\sigma,K_a\gamma) \in \Sigma$, then $(\sigma',\gamma) \in \Sigma$ for every tableau σ' that occurs in Σ and is accessible from σ;*
4. *if $(\sigma,\neg K_a\gamma) \in \Sigma$, then $(\sigma',\gamma) \in \Sigma$ for some tableau σ' that is accessible from σ;*
5. *every branch of any tableau which is accessible from θ is complete or closed as well.*

Definition 2.3. *We say that a tableau $\mathcal{T}$ is* completed *if every branch of σ is complete or closed.*

So, if a branch θ of a tableau $\mathcal{T}$ is *complete* and *open*, then we have at least one open branch (that is also complete) per subordinated tableaux to θ.

Theorem 2.4. *Every complete and open branch of a tableau is satisfiable.*

Proof. Let θ be a complete and open branch of a tableau $\mathcal{T}$ and Σ be a set of formulae of θ and of the tableaux $\mathcal{T}_1, \mathcal{T}_2, \ldots$ (which are recursively subordinated to θ). We construct a model $\mathcal{M}$ where S is the set of tableaux $\{\mathcal{T}, \mathcal{T}_1, \mathcal{T}_2, \ldots\}$, $\sim_a$ is built from the pairs $(\mathcal{T}_1, \mathcal{T}_2)$, such that $\mathcal{T}_2$ is subordinated to $\mathcal{T}_1$ and satisfying the following conditions, where E is an expression and the prefixes $\sigma, \sigma_1, \sigma_2, \ldots$ are associated to $\{\mathcal{T}, \mathcal{T}_1, \mathcal{T}_2, \ldots\}$, respectively:

1. if $(\sigma, E) \in \Sigma$, then $V(\sigma, E) = T$;
2. if $(\sigma, \neg E) \in \Sigma$, then $V(\sigma, E) = F$;
3. if $(\sigma, E) \notin \Sigma$ and $(\sigma, \neg E) \notin \Sigma$, then $V(\sigma, E) = T$ can have any value. Let's choose F by default.

Now, for any $(\sigma, \gamma) \in \Sigma$, we have $s \Vdash \gamma$, where γ is a formula and s a possible state associated to σ. According to γ structure:

- for $(\sigma, p), (\sigma, \alpha), (\sigma, \beta), (\sigma, K_a\gamma)$ and $(\sigma, \neg K_a\gamma)$ the proof is found in [1]. We only show the case for the new rules presented in Section 4 of the original submission;
- The pair $(\sigma, \{m\}_k) \in \Sigma$, for some prefix σ. By condition 1 of Definition 2.2, we have $(\sigma, m) \in \Sigma$ and $(\sigma, k) \in \Sigma$ and by the induction hypothesis $s \Vdash m$ and $s \Vdash k$ and also $s \Vdash m \wedge k$, by the soundness of axiom axiom 6 of Section 3.1 of the original submission, we have $s \Vdash \{m\}_k$;
- The pair $(\sigma, \neg\{m\}_k) \in \Sigma$, for some prefix σ. By condition 2 of Definition 2.2, we have $(\sigma, \neg m) \in \Sigma$ or $(\sigma, \neg k) \in \Sigma$ and by the induction hypothesis $s \Vdash \neg m$ or $s \Vdash \neg k$ and also $s \Vdash \neg m \vee \neg k$ and $s \Vdash \neg(m \wedge k)$, by the soundness of the contrapositive of axiom 6 of Section 3.1 of the original submission, we have $s \Vdash \neg\{m\}_k$;
- the cases for rules $\mathrm{R_{Pair}}$ and $\mathrm{R^{\neg}_{Pair}}$ are analogous to the cases for rules $\mathrm{R_{Dec}}$ and $\mathrm{R^{\neg}_{Enc}}$, respectively, but using axiom 8 of Section 3.1 of the original submission.

Therefore, our model satisfies Σ. □

Theorem 2.5. *If a formula γ is valid, then γ has a proof by tableaux method.*

Proof. Let $\mathcal{T}$ be a completed tableau, started with $\neg\gamma$. If it is open, then $\neg\gamma$ is satisfiable by theorem 2.4. So, γ cannot be valid. Therefore, if γ is valid, then $\mathcal{T}$ is closed and γ has a proof by tableaux method. □

3 Termination property

For the tableaux rules presented in Section 2.3 of the original submission, Massacci [3] provides the termination argument below, adapted for our semantics.

3.1 Classical and modal rules

To guarantee the termination of the proof search it's used the "loop checking" approach, a combination of techniques to apply any rule only after check if it was not applied already to the same antecedent. First we need some definitions:

Definition 3.1. *In a branch θ of a tableau σ, a prefixed formula (σ,γ) is* reduced *for a rule in θ:*

- *if the rule generates (σ',γ') and (σ',γ') is in θ; or*
- *if the rule splits the tableau into (σ_1,γ_1) and (σ_2,γ_2) and at least one of those is in θ.*

The formula (σ,γ) is fully reduced *in θ if it is reduced for all applicable rules and σ is (fully) reduced if all prefixed formula (σ,γ) are (fully) reduced as well.*

So, for tableaux method for logic $\mathscr{K}$, the following technique is sufficient to terminate:

Technique 3.2. *Apply a rule to a prefixed formula (σ,γ) in θ only if the formula is not already reduced according to Definition 3.1, except for the knowledge operator.*

But for our case, the "loop checking" concept is required. Let's begin with the definition of a *copy* of a prefix:

Definition 3.3. *A prefix σ is a copy of a prefix σ_0 for branch θ if for every formula γ one has $(\sigma,\gamma)\in\theta$ iff $(\sigma_0,\gamma)\in\theta$.*

Now we define what is a π-*reduced* prefix:

Definition 3.4. *A prefix is π-reduced in θ if it is reduced for all rules except R_π. A branch θ is π-completed if:*

- *all prefixes are π-reduced in θ;*
- *for every σ that is not fully reduced there is a fully reduced copy σ_0 shorter than σ.*

So, the idea is to restrict the usage of R_π to formulae belonging to copies. The following technique together with Technique 3.2 prove that we will always have a π-completed branch:

Technique 3.5. *Select the prefixed formulae with the shortest prefix.*

As we have a π-completed branch, the next technique guarantees termination:

Technique 3.6. *Check if the prefix of a π-formula is not a copy of a shorter prefix before reducing it.*

3.2 Dolev-Yao Multi-Agent Epistemic Logic rules

As rules R_{Dec}, $R^{\neg}_{Enc}$, R_{Pair}, $R^{\neg}_{Pair}$ always yield a smaller conclusion than the premises, that is, they are considered *analytic* rules, the argument explained in Section 3.1 is not interfered.

References

[1] Marcos M. C. Costa (1992): *Introdução à Lógica Modal Aplicada à Computação*. UFRGS.

[2] Melvin Fitting (1983): *Proof Methods for Modal and Intuitionistic Logics*. Springer, Dordrecht, doi:10.1007/978-94-017-2794-5.

[3] Fabio Massacci (2000): *Single Step Tableaux for Modal Logics*. Journal of Automated Reasoning 24(3), pp. 319–364, doi:10.1023/A:1006155811656.

[4] Raymond M. Smullyan (1968): *First-Order Logic*. Ergebnisse der Math. und ihrer Grenzgebiete 43, Berlin, Germany: New York [Etc.]Springer-Verlag, doi:10.1007/978-3-642-86718-7.