1 *ReLo* Tableau Soundness & Completeness

This document describes in detail the proofs of Soundness and Completeness for the *ReLo* logic Grilo & Lopes (2020). The main definitions of a tableau for this logic are discussed, along with the aforementioned proofs. The definitions of tableau and branch are standard in the literature De Giacomo & Massacci (2000), Pratt (1980), Massacci (1994) and are as follows.

Definition 1 (Tableau in *ReLo*)**.** *A tableau $\mathcal{T}$ in ReLo is a rooted tree with nodes labeled with formulae, prefixed by states in an auxiliary finite sequence of states which guides the decomposition of formulae. A branch $\mathcal{B}$ is a path from the root to a leaf node. A segment $\mathcal{S}$ is a path from the root node to some intermediate node (i.e., a non-leaf node).*

Intuitively, a tableau in *ReLo* of a formula φ denotes a failed attempt to prove $\neg\varphi$, which in turn yields φ as valid. Branches of the tableau can be interpreted as a tentative of the construction of a model that holds for the initial formula $\neg\varphi$, and segments as the intermediate steps taken to construct such model. Definition 2 formalizes the rules that may be applied to formulas in *ReLo*'s Tableau.

Definition 2. *Tableau rules for ReLo*

- *Propositional rules*

$$(\textbf{\textit{And-T}})\quad \frac{w\!:\ \varphi \wedge \psi\ :\ T}{\begin{array}{c} w\!:\ \varphi\ :\ T \\ w\!:\ \psi\ :\ T \end{array}} \qquad (\textbf{\textit{And-F}})\quad \frac{w\!:\ \varphi \wedge \psi\ :\ F}{w\!:\ \varphi\ :\ F \qquad w\!:\ \psi\ :\ F}$$

$$(\textbf{\textit{Or-T}})\quad \frac{w : \varphi \vee \psi\ :\ T}{w\!:\ \varphi\ :\ T \qquad w\!:\ \psi\ :\ T} \qquad (\textbf{\textit{Or-F}})\quad \frac{w : \varphi \vee \psi\ :\ F}{\begin{array}{c} w\!:\ \varphi\ :\ F \\ w\!:\ \psi\ :\ F \end{array}}$$

$$(\textbf{\textit{Neg-T}})\quad \frac{w : \neg\varphi\ :\ T}{w : \varphi\ :\ F} \qquad (\textbf{\textit{Neg-F}})\quad \frac{w : \neg\varphi\ :\ F}{w : \varphi\ :\ T}$$

$$(\rightarrow -T)\quad \frac{w : \varphi \rightarrow \psi\ :\ T}{w : \varphi\ :\ F \qquad w : \psi : T} \qquad (\rightarrow -F)\quad \frac{w : \varphi \rightarrow \psi\ :\ F}{\begin{array}{c} w : \varphi\ :\ T \\ w : \psi\ :\ F \end{array}}$$

- *Modal rules*

$$(\langle t, \pi\rangle - T)\quad \frac{w\!:\ \langle t, \pi\rangle\varphi\ :\ T}{x\!:\ \varphi\ :\ T}$$

x a new state in the sequence of states with $wR_\pi x$.

$$(\langle t, \pi\rangle - F)\quad \frac{w\!:\ \langle t, \pi\rangle\varphi\ :\ F}{x\!:\ \varphi\ :\ F}$$

x any state (new or already existing in the sequence of states) with x accessed from w by π.

$([t,\pi]-T)$ $\dfrac{w\text{: } [t,\pi]\varphi : T}{x\text{: } \varphi : T}$

x any state (new or already eisting in the sequence of states) with x accessed from w by π.

$([t,\pi]-T)$ $\dfrac{w\text{: } [t,\pi]\varphi : F}{x\text{: } \varphi : F}$

x a new state in the sequence of states with $wR_\pi x$.

$(\langle t,\pi\rangle - \textbf{(R)} - T)$ $\dfrac{w\text{: } \langle t,\pi\rangle\varphi : T, \text{ iff } f(t,\pi)=\epsilon}{w\text{: } \varphi : T}$

w the same state. This rule captures the behavior of axiom $\mathcal{R}$. Its non validity is the normal case for diamond formulas.

- *Program operator rules*
  - *Iteration*

$(\langle t,\pi^\star\rangle)$-***T*** $\dfrac{w\text{: } \langle t,\pi^\star\rangle\varphi : T}{w\text{: } \mathcal{X}_{\langle\rangle} : T}$

$\mathcal{X}_{\langle\rangle} = \langle t,\pi^\star\rangle\varphi$

$(\langle t,\pi^\star\rangle)$-***F*** $\dfrac{w\text{: } \langle t,\pi^\star\rangle\varphi : F}{\begin{array}{c} w\text{: } \varphi : F \\ w\text{: } \langle t,\pi\rangle\langle t'_\pi,\pi^\star\rangle\varphi : F\end{array}}$

The data sequence t'_π is a sequence such that $t \subseteq \delta(w)$ and $t'_\pi \prec f(t_\pi,\pi)$.

$([t,\pi^\star])$-***T*** $\dfrac{w\text{: } [t,\pi^\star]\varphi : T}{\begin{array}{c} w\text{: } \varphi : T \\ w\text{: } [t,\pi][t'_\pi,\pi^\star]\varphi : T\end{array}}$

The data sequence t'_π is a sequence such that $t \subseteq \delta(w)$ and $t'_\pi \prec f(t_\pi,\pi)$.

$([t,\pi^\star])$-***F*** $\dfrac{w\text{: } [t,\pi^\star]\varphi : F}{w\text{: } \mathcal{X}_{[]} : F}$

$\mathcal{X}_{[]} = [t,\pi^\star]\varphi$

$(\mathcal{X}_{\langle\rangle})$ $\dfrac{w\text{: } \mathcal{X}_{\langle\rangle} : T}{w : \varphi : T \qquad \begin{array}{c} w : \varphi : F \\ w\text{: } \langle t,\pi\rangle\mathcal{X}_{\langle\rangle} : T\end{array}}$

$(\mathcal{X}_{[]})$ $\dfrac{w\text{: } \mathcal{X}_{[]} : F}{w\text{: } \varphi : F \qquad \begin{array}{c} w : \varphi : T \\ w\text{: } [t,\pi]\mathcal{X}_{[]} : F\end{array}}$

The iteration rules introduced by $\mathcal{X}$ rules can be intuitively detailed as follows: for formulas $\langle t,\pi^\star\rangle\varphi$, the reduction of the modality employing $\star$ is done by stating that φ is either valid in the reached state, or we may reduce it once again in another state where φ is valid. We call $\mathcal{X}$ an eventuality, $\mathcal{X}$ a new propositional symbol.

In what follows some useful definitions regarding notions of iteration in the tableau are discussed. We follow the methodology proposed by De Giacomo & Massacci (2000).

A set of prefixed formulae of a segment $\mathcal{S}$ is the set of all formulae φ in $\mathcal{S}$ that are prefixed by some w as

$$\mathcal{S}/w = \{\varphi \mid w : \varphi \in \mathcal{S}\}$$

A prefix w is said to be reduced in a branch $\mathcal{S}$ if the Modal rules are the only rules not applied yet to the set $\mathcal{S}/w$. It is fully reduced if all possible rules have been applied.

Intuitively, the reduction states that all iteration rules have been applied in formulae in $\mathcal{S}/w$ (if possible), while the fully reduction yields formulae with all transitional (i.e., rules that may change a state) rules applied.

A branch $\mathcal{B}$ is said to be π-completed if all prefixes are reduced, and for every w which is not fully reduced, there is a segment $\mathcal{S}$ and a prefix w' of $\mathcal{S}$ smaller than $\mathcal{B}$ and w which is fully reduced and w' is a copy of w in $\mathcal{B}$.

A prefix w' of a segment $\mathcal{S}'$ is said to be a copy of another prefix w in a segment $\mathcal{S}$ if $\mathcal{S}/w = S'/w'$ and they have the same transitions format given by the auxiliary tree of states.

Given two $\mathcal{X}$ formulae, $w : \mathcal{X}_i$ and $w' : \mathcal{X}_j$ that are considered equal, we define that $\mathcal{X}_j$ collapses into $\mathcal{X}_i$ as follows.

Definition 3. *A branch $\mathcal{B}$ is ignorable if it contains a $\mathcal{X}$ introduced by any of the $\mathcal{X} -$ rules which collapses to itself (either directly or transitively) and its corresponding shorter branch (obtained by $\mathcal{X}$ rule's application as the resulting left branch) is closed.*

Definition 4 (fulfilled $\mathcal{X}$)**.** *A $\mathcal{X}$ is said to be fulfilled if there is a $w : \varphi$ in the same segment with $w : \mathcal{X}_{\langle\rangle}$, yielding a contradiction, or $\mathcal{X}_{\langle\rangle}$ collapses to a $\mathcal{X}_{\langle\rangle_0}$ which has already appeared in the branch and is fulfilled.*

The idea Definition 3 presents is to denote that the looping introduced by $\mathcal{X}$ rules may not close, yielding a unsuccessful loop by always iterating and leaving the rightmost branch "open". If at some point the proof reaches a $\mathcal{X}_i$ with the same formulae as an already visited state, we can conclude that it will never be fulfilled and therefore ignore the remainder of the branch, as it has already been reduced earlier in the proof and it may introduce a potentially infinite pattern in the formula decomposition process. With Definition 3, we may define the closure of *ReLo* tableau as follows.

Definition 5 (Tableau closure)**.** *A tableau $\mathcal{T}$ is closed if all of its branches are either contradictory or ignorable, yielding that the formula syntactically holds in ReLo. Conversely, a tableau $\mathcal{T}$ is open if there is at least a branch with no contradictions or it is not ignorable.*

1.1 Tableau Usage Examples

In what follows we provide usage examples of the tableau developed for *ReLo*. We show its usage for the axioms defined in Section **??**, stating that they are indeed valid in the proposed Tableau:

- $\mathcal{K}$: $[t,(f,b)](\varphi \to \psi) \to ([t,(f,b)]\varphi \to [t,(f,b)]\psi)$

1.	w : [t, (f,b)]$(\varphi \to \psi) \to$ ([t, (f,b)]$\varphi \to$ [t, (f,b)]$\psi) : F$		
2.	w : [t, (f,b)]$(\varphi \to \psi) : T$		$1, \to$
3.	w : ([t, (f,b)]$\varphi \to$ [t, (f,b)]$\psi) : F$		$1, \to$
4.	w : [t, (f,b)]$\varphi : T$		$3, \to$
5.	w : [t, (f,b)]$\psi : F$		$3, \to$
6.	$x : \psi : F$		$5, [] - F$
7.	$x : \varphi \to \psi : T$		$2, [] - T$
8.	$x : \varphi : F$	$x : \psi : V$	$5, \to -T$
9.	$x : \varphi : T$	$\times$ 6,8	$4, [] - T$
	$\times$ 8,9		

- And-$\to$: $[t,(f,b)](\varphi \wedge \psi) \to ([t,(f,b)]\varphi \wedge [t,(f,b)]\psi)$

1.	w : [t, (f,b)]$(\varphi \wedge \psi) \to$ ([t, (f,b)]$\varphi \wedge$ [t, (f,b)]$\psi) : F$		
2.	w : [t, (f,b)]$(\varphi \wedge \psi) : T$		$1, \to -F$
3.	w : ([t, (f,b)]$\varphi \wedge$ [t, (f,b)]$\psi) : F$		$1, \to -F$
4.	w : [t, (f,b)]$\varphi : F$	w : [t, (f,b)]$\psi : F$	$3, \wedge - F$
5.	$x : \varphi : F$	$y : \psi : F$	$4, [] - F$
6.	$x : \varphi \wedge \psi : T$	$y : \varphi \wedge \psi : T$	$2, [] - T$
7.	$x : \psi : T$	$y : \varphi : T$	$6, \wedge - T$
8.	$x : \varphi : T$	$y : \psi : T$	$6, \wedge - T$
	$\times$ 5, 8	$\times$ 5, 8	

- $R \to$: $\langle t,(f,b)\rangle\varphi \to \varphi, iff(t,(f,b)) = \epsilon$

1.	$w : \langle$t, (f,b)$\rangle\varphi \to \varphi$, if f(t,(f,b) : F	
2.	$w : \langle$t, (f,b)$\rangle\varphi : T$	$1, \to -F$
3.	$w : \varphi : F$	$1, \to -F$
4.	$w : \varphi : T$	$2, <> -R - T$
	$\times$ 3,4	

- $R \leftarrow$: $\varphi \to \langle t,(f,b)\rangle\varphi, iff(t,(f,b)) = \epsilon$

1.	$w : \varphi \to \langle$t, (f,b)$\rangle\varphi$, if f(t,(f,b) : F	
2.	$w : \varphi : T$	$1, \to -F$
3.	$w : \langle$t, (f,b)$\rangle\varphi : F$	$1, \to -F$
4.	$w : \varphi : F$	$2, <> -R - T$
	$\times$ 2,4	

- It$- \to$: $\varphi \wedge [\text{t, (f,b)}][\text{t}_{f,b}\text{, (f,b)}^\star]\varphi \to [\text{t, (f,b)}^\star]\varphi$

1.	$w : \varphi \wedge [\text{t, (f,b)}][\text{t}_{f,b}\text{, (f,b)}^\star]\varphi \to [\text{t, (f,b)}^\star]\varphi : F$	
2.	$w : \varphi \wedge [\text{t, (f,b)}][\text{t}_{f,b}\text{, (f,b)}^\star]\varphi : T$	$1, \to -F$
3.	$w : [\text{t, (f,b)}^\star]\varphi : F$	$1, \to -F$
4.	$w : \varphi : T$	$2, \wedge - T$
5.	$w : [\text{t, (f,b)}][\text{t}_{f,b}\text{, (f,b)}^\star]\varphi : T$	$2, \wedge - T$
6.	$w : [\text{t, (f,b)}^\star]\varphi : T$	$5, [] - T$
	$\times$ 3,6	

- It$- \leftarrow$: $[\text{t, (f,b)}^\star]\varphi \to \varphi \wedge [\text{t, (f,b)}][\text{t}_{f,b}\text{, (f,b)}^\star]\varphi$

1.	$w : [\text{t, (f,b)}^\star]\varphi \to \varphi \wedge [\text{t, (f,b)}][\text{t}_{f,b}\text{, (f,b)}^\star]\varphi : F$		
2.	$w : [\text{t, (f,b)}^\star]\varphi : T$		$1, \to -F$
3.	$w : \varphi \wedge [\text{t, (f,b)}][\text{t}_{f,b}\text{, (f,b)}^\star]\varphi : F$		$1, \to -F$
4.	$w : \varphi : F$	$w : [\text{t, (f,b)}][\text{t}_{f,b}\text{, (f,b)}^\star]\varphi : F$	$2, \wedge - F$
5.	$w : \varphi : T$	$x : [\text{t, (f,b)}^\star]\varphi : F$	$2, []^\star - T$; $4, [] - T$
6.	$w : [\text{t}_{f,b}\text{, (f,b)}^\star]\varphi : T$	$\times$ 2,5	$2, []^\star - T$
	$\times$ 4,5		

- Ind: $\varphi \wedge [t, (f, b)^\star](\varphi \to [t_{(f,b)^\star}, (f, b)]\varphi) \to [t, (f, b)^\star]\varphi$

1.	$w : \varphi \wedge [\text{t, (f,b)}^\star](\varphi \to [\text{t', (f,b)}]\varphi) \to [\text{t, (f,b)}^\star]\varphi : F$				
2.	$w : \varphi \wedge [\text{t, (f,b)}^\star](\varphi \to [\text{t', (f,b)}]\varphi) : T$				$1, \to -F$
3.	$w : [\text{t, (f,b)}^\star]\varphi : F$				$1, \to -F$
4.	$w : \varphi : T$				$2, \wedge - T$
5.	$w : [\text{t, (f,b)}^\star](\varphi \to [\text{t', (f,b)}]\varphi : T$				$2, \wedge - T$
6.	$w : \varphi : F$	$w : \varphi : T$			$3, []^\star - F$
7.	$\times$ 4, 6	$w : [\text{t', (f,b)}]\mathcal{X}_{[]} : F$			$3, []^\star - F$
8.		$x : \mathcal{X}_{[]} : F$			$7, [] - T$
9.		$x : \varphi \to [\text{t', (f,b)}]\varphi : T$			$5, [] - T$
10.		$x : [\text{t, (f,b)}^\star](\varphi \to [\text{t', (f,b)}]\varphi : T$			$5, [] - T$
11.		$x : \varphi : F$	$x : [\text{t', (f,b)}]\varphi : T$		$9, \to -T$
12.		$\times$ 4, 11	$x : \varphi : F$	$x : \varphi : T$	$8, \mathcal{X}_{[]}$
13.			$\times$ 4, 12	$x : [\text{t',(f,b)}]\mathcal{X}_{[]} : F$	$8, \mathcal{X}_{[]}$
14.				$y : \mathcal{X}_{[]} : F$	$13, [] - F$
				$\times$ Ignorable, 14 = 8	

1.2 Soundness

The proof of *ReLo*'s tableau soundness follows standard techniques in literature Fitting (1983, 2012). We adapt them to *ReLo*'s reality in order to proceed. The soundness of the tableau proceeds by defining the notion of a satisfiable tableau, and then showing that satisfiability is a loop invariant property.

Let us define a map $m\colon \Phi \to \{T, F\}$ as a map that maps a formula (i.e., its labeling state and the formula itself) to T. Intuitively, m states that the tableau formula $w : \varphi$ is valid at the state denoted by its label.

Definition 6 (Tableau satisfiability)**.** *A ReLo tableau $\mathcal{T}$ is satisfiable if at least of one its branches $\mathcal{B}$ is satisfiable. A branch $\mathcal{B}$ is satisfiable if there is a map m which maps each of its formulae $(w : \varphi) \in \mathcal{B}$ to T.*

Lemma 1 (Satisfiability for *ReLo* Tableau is loop invariant)**.** *The application of any tableau expansion rule (as defined in Section* **??***) to a satisfiable tableau $\mathcal{T}$ will result in another satisfiable tableau $\mathcal{T}'$.*

Proof. Suppose $\mathcal{T}$ is a satisfiable tableau. Now, suppose a tableau expansion rule is applied to some formula ψ in a branch $\mathcal{B}$, resulting in a satisfiable tableau $\mathcal{T}'$. The proof proceeds by induction on the tableau expansion rules, showing that for each rule application, the resulting tableau is indeed satisfiable. From Definition 6, as $\mathcal{T}$ is satisfiable, it has a satisfiable branch. Suppose b is such a satisfiable branch of $\mathcal{T}$. We have to analyze the following cases:

1. case $b \neq \mathcal{B}$: as no rule was applied in b and b is a branch of $\mathcal{T}'$, b is still a satisfiable branch of a satisfiable tableau $\mathcal{T}'$.

2. case $b = \mathcal{B}$: branch $\mathcal{B}$ is satisfiable in $\mathcal{T}$. We must show that an tableau expansion rule application must yield a satisfiable tableau $\mathcal{T}'$. We need to consider each of the possible tableau expansion rules that could be applied to formula ψ. For these cases, we only consider rules $(* - T)$ as their dual may be obtained as follows:

   - $(And - F)$ may be obtained by the negation of $(Or - T)$
   - $(Or - F)$ may be obtained by the negation of $(And - T)$
   - $(\to -F)$ may be obtained by rewriting it to $\neg\varphi_1 \vee \varphi_2$, which reduces to the case of $(Or - F)$
   - $(\langle t, \pi\rangle{-}F)$ may be obtained by the negation of $([t, \pi^\star]{-}T)$
   - $([t, \pi]{-}F)$ may be obtained by the negation of $(\langle t, \pi\rangle{-}T)$
   - $(\langle t, \pi^\star\rangle{-}F)$ may be obtained by the negation of $([t, \pi^\star]{-}T)$
   - $([t, \pi^\star]{-}F)$ may be obtained by the negation of $(\langle t, \pi^\star\rangle{-}T)$

   Therefore, let the tableau expansion rule applied to a formula ψ be:

   - $(And - T)$: by induction, ψ is $w : \varphi_1 \wedge \varphi_2$ mapped to T by a map m in $\mathcal{T}$. Therefore, because $w : \varphi_1$ and $w : \varphi_2$ maps both to T by m in $\mathcal{B}'$ as a satisfiable branch, $\mathcal{T}'$ is a satisfiable tableau.

- $(or - T)$: by induction, ψ is $w : \varphi_1 \vee \varphi_2$ mapped to T by a map m in $\mathcal{T}$. The application of $(or - T)$ yields two different branches from $\mathcal{B}$, namely $\mathcal{B}_1$ with φ_1 and $\mathcal{B}_2$ with φ_2, and from ψ mapping to T by m, either φ_1 or φ_2 (or both) maps to T by m. If $m(\varphi_1) = T$, then $\mathcal{B}_1$ denotes a satisfiable branch in $\mathcal{T}'$, and if $m(\varphi_2) = T$, then $\mathcal{B}_2$ denotes a satisfiable branch in $\mathcal{T}'$. In both cases, applying this rule in a formula in the branch $\mathcal{B}$ results in a satisfiable branch $\mathcal{B}'$ of a satisfiable tableau $\mathcal{T}'$.
- $(\rightarrow -T)$: by induction, ψ is $w : \varphi_1 \rightarrow \varphi_2$, which can be rewritten as $\neg\varphi_1 \vee \varphi_2$, which reduces to the case of $(or - T)$.
- $(\langle t, \pi\rangle - T)$: by induction, ψ is $w : \langle t, \pi\rangle\varphi$ mapped to T by a map m in $\mathcal{T}$. This rule yields $m(x : \varphi) = T$ in $\mathcal{T}'$, where φ is labeled with a new state x in the sequence of states allocated to formulas in $\mathcal{T}$. Therefore, the resulting branch $\mathcal{B}'$ is a satisfiable branch in $\mathcal{T}'$.
- $([t, \pi] - T)$: by induction, ψ is $w : [t, \pi]\varphi$ mapped to T by a map m in $\mathcal{T}$. This rule yields $m(x : \varphi) = T$ in $\mathcal{T}'$, where φ is labeled with a new (or already existing) state in the sequence of states allocated to formulas in $\mathcal{T}$. Therefore, the resulting branch $\mathcal{B}'$ is a satisfiable branch in $\mathcal{T}'$.
- $([t, \pi^\star] - T)$: by induction, ψ is $w : [t, \pi^\star]\varphi$ mapped to T by a map m in $\mathcal{T}$. The branch $\mathcal{B}'$ that results from applying this rule to ψ is $\mathcal{B}$ with formulae $w : \varphi$ and $w : [t, \pi][t'_\pi, \pi^\star]\varphi$, where both formulae maps to t by m. Therefore, $\mathcal{B}'$ is a satisfiable branch, yielding $\mathcal{T}'$ as a satisfiable tableau.
- $(\langle t, \pi^\star\rangle - T)$: by induction, ψ is $w : \langle t, \pi^\star\rangle\varphi$ with $m(\langle t, \pi^\star\rangle\varphi) = T$ in $\mathcal{T}$. This rule yields a new propositional symbol $\mathcal{X}_{\langle\rangle}$, which in turn may be further reduced by the $\mathcal{X}_{\langle\rangle}$ rule. The application of $\mathcal{X}_{\langle\rangle}$ results in a tableau $\mathcal{T}'$ with two new branches as follows.
  - $\mathcal{B}_1$ which results from appending $w : \varphi$ to $\mathcal{B}$. From the rule, $m(w : \varphi)$ maps to T in $\mathcal{B}_1$, rendering it as a satisfiable branch of $\mathcal{T}'$, which consequently is a satisfiable tableau.
  - $\mathcal{B}_2$, which results from appending both $w : \varphi$ and $w : \langle t, \pi\rangle\mathcal{X}_{\langle\rangle}$. From the rule, in this branch $m(w : \varphi) = F$ and $m(w : \langle t, \pi\rangle\mathcal{X}_{\langle\rangle}) = T$. We need to ensure now that the outcome of multiple $\mathcal{X}_{\langle\rangle}$ are also satisfiable, which may result in $\mathcal{X}_{\langle\rangle}$ being ignorable branches can be discarded, still resulting on a satisfiable tableau $\mathcal{T}'$.

    Therefore, let us define a order relation $R_\prec$ as a relation over formulae $\mathcal{X}_{\langle\rangle}$ in a branch $\mathcal{B}$ as follows:
    - $(x_0 : \mathcal{X}_{\langle\rangle})\ R_\prec\ (x_1 : \mathcal{X}_{\langle\rangle})$ iff $x_0 : \mathcal{X}_{\langle\rangle}$ belongs to a segment X_0, $x_1 : \mathcal{X}_{\langle\rangle}$ appears in a segment X_1 and (x_0, X_0) is shorter than (x_1, X_1).
    - $(x_1 : \mathcal{X}_{\langle\rangle_1})\ R_\prec\ (x_0 : \mathcal{X}_{\langle\rangle_0})$ iff $\mathcal{X}_{\langle\rangle_1}$ collapses in $\mathcal{X}_{\langle\rangle_0}$ and x_0 is the fully reduced copy of x_1.

Now, let $R_{\prec-T}$ be $R_{\prec}$'s transitive closure. Therefore, by $R_{\prec}$, either $\mathcal{X}_{\langle\rangle}$ will be fulfilled by some $y : \varphi$ (as yielded by $\mathcal{X}_{\langle\rangle}$'s resulting rule application), or that at some point ω, $(x_0 : {\mathcal{X}_{\langle\rangle}}_0)$ $R_{\prec-T}$ $(\omega : {\mathcal{X}_{\langle\rangle}}_\omega)$ collapses to a already visited $x_i : {\mathcal{X}_{\langle\rangle}}_i$, which in turn satisfies the conditions to be considered a ignorable branch. Therefore, $\mathcal{B}_2$ is a ignorable branch obtained by the aforementioned steps, resulting in a consistent $\mathcal{T}'$.

□

With Lemma 1, we may proceed with the following lemmas, also standards in the literature De Giacomo & Massacci (2000), Fitting (2012).

Lemma 2. *If there is a closed tableau $\mathcal{T}$ for φ, then φ is not satisfiable*

Proof. The proof proceeds by contradiction. Suppose that there is a closed tableau for φ, but φ is satisfiable. The construction of the tableau for φ is the tableau $\langle w : \varphi\rangle$ which is basically φ labeled with a "starting" state w. This tableau is satisfiable (because we supposed φ is satisfiable). Then, from Lemma 1, each subsequent tableau generated by applying the tableau expansion rules to subformulas of φ results in satisfiable tableau, even for the closed tableau $\mathcal{T}$. But because $\mathcal{T}$ is closed, there is no satisfiable branch $\mathcal{B}$ in $\mathcal{T}$, which contradicts Lemma 1. □

Lemma 3 (Soundness of *ReLo* tableau)**.** *If φ has a tableau proof, then φ is a tautology in ReLo.*

Proof. A tableau proof for φ comes from a closed tableau for $\neg\varphi$. From Lemma 2, if $\neg\varphi$ has a closed tableau, then it is not satisfiable. Therefore, it follows that φ is indeed satisfiable. □

1.3 Completeness

For completeness, we will follow a standard argument in Literature Fitting (2012) in which considers the proof of the axiomatic system of *ReLo* in the proposed tableau. Because *ReLo*'s axiomatic system proposed in Definition **??** is both sound and complete, let us consider the following:

Lemma 4 (Completeness of *ReLo*'s Tableau)**.** *The tableau proposed in Section 1 is complete, i.e., if φ is valid in ReLo then it have a Tableau proof in ReLo.*

Proof. Section 1.1 provides syntactic proofs for each of the axiom rules introduced by Definition **??**. As *ReLo* is proved to be both sound and complete, a tableau proof for each of its axioms denote that every formula φ that is provable in *ReLo* contains a tableau proof. Therefore, *ReLo*'s Tableau is complete. □

References

De Giacomo, G. & Massacci, F. (2000), ‘Combining deduction and model checking into tableaux and algorithms for converse-pdl’, *Information and Computation* **162**(1-2), 117–137.

Fitting, M. (1983), *Proof methods for modal and intuitionistic logics*, Vol. 169, Springer Science & Business Media.

Fitting, M. (2012), *First-order logic and automated theorem proving*, Springer Science & Business Media.

Grilo, E. & Lopes, B. (2020), Relo: a dynamic logic to reason about reo circuits, *in* ‘Proceedings of the 15th International Workshop on Logical and Semantic Frameworks, with Applications (LSFA)’, p. 32.

Massacci, F. (1994), Strongly analytic tableaux for normal modal logics, *in* ‘International Conference on Automated Deduction’, Springer, pp. 723–737.

Pratt, V. R. (1980), ‘A near-optimal method for reasoning about action’, *Journal of Computer and System Sciences* **20**(2), 231–254.