Relational Dual Tableaux
Foundations and Applications

Joanna Golińska-Pilarek

University of Warsaw, Poland

28th Italian Conference on Computational Logic
September 25-27, 2013
Catania, Italy
Dual Tableaux

- Based on Rasiowa-Sikorski diagrams [1960]
- The rules are of the form:
  \[
  \begin{array}{c}
  \Phi \\
  \Phi_1 | \ldots | \Phi_n
  \end{array}
  \]

- ’, ’ – disjunction, ’|’ – conjunction
- \(X\) is valid iff the meta-disjunction of formulas from \(X\) is valid
- The rules are semantically invertible, that is for every set \(X\) of formulas:
  \[X \cup \Phi\ \text{is valid if and only if all } X \cup \Phi_i \text{ are valid}\]

- Axioms: some valid sets of formulas
- Proof: a decomposition tree
- Provability of a formula: existence of a closed proof tree
Decomposition rules for the classical connectives:

\[(RS \lor) \quad \frac{\varphi \lor \psi}{\varphi, \psi} \quad (RS \neg \lor) \quad \frac{\neg(\varphi \lor \psi)}{\neg\varphi | \neg\psi}\]

\[(RS \neg) \quad \frac{\neg\neg\varphi}{\varphi}\]

Decomposition rules for quantifiers:

\[(RS \forall) \quad \frac{\forall x \varphi(x)}{\varphi(z)} \quad (RS \neg \forall) \quad \frac{\neg \forall x \varphi(x)}{\neg \varphi(z), \neg \forall x \varphi(x)}\]

*z* is a new variable

*z* is any variable
Specific rule for the identity:

\[(RS=) \quad \frac{\varphi(x)}{x = y, \varphi(x) \mid \varphi(y), \varphi(x)}\]

\(\varphi\) is an atomic formula, \(y\) is any variable

Axioms

- \(\varphi, \neg \varphi\)
- \(x = x\)
Example \( \forall x (\varphi \lor \psi(x)) \rightarrow (\varphi \lor \forall x \psi(x)) \)

\[
\begin{align*}
\neg \forall x (\varphi \lor \psi(x)) & \lor (\varphi \lor \forall x \psi(x)) \\
& \quad \text{(RS} \lor \text{) twice} \\
\neg \forall x (\varphi \lor \psi(x)), \varphi, \forall x \psi(x) & \\
& \quad \text{(RS} \forall \text{) with a new variable } z \\
\neg \forall x (\varphi \lor \psi(x)), \varphi, \psi(z) & \\
& \quad \text{(RS} \neg \forall \text{) with variable } z \\
\neg (\varphi \lor \psi(z)), \varphi, \psi(z), \ldots & \\
& \quad \text{(RS} \neg \lor \text{)} \\
\neg \varphi, \varphi, \ldots & \quad \text{closed} \\
\neg \psi(z), \psi(z), \ldots & \quad \text{closed}
\end{align*}
\]
Duality of RS and Smullyan Tableaux

Dual formulas and rules

- the root of a tree: \( \varphi \) in RS-system and \( \neg \varphi \) in T-system
- \( \psi \) is dual to \( \neg \psi \)
- RS-rule (\( \# \)) is dual to T-rule (\( \neg \# \))
- RS-rule (\( \neg \# \)) is dual to T-rule (\( \# \))
- double negation rule (\( \neg \)) is the same in both systems.
Transformation of proof trees

If $\mathcal{D}$ is a proof tree of RS (resp. T), then $\Gamma(\mathcal{D})$ is a proof tree of T (resp. RS) obtained by replacing:

- all the formulas of $\mathcal{D}$ with the dual formulas;
- each rule with dual rule + (double negation rule if needed).

Duality Theorem

$\Gamma(\Gamma(\mathcal{D})) = \mathcal{D}$.

That is RS and T proof trees are dual modulo double negation rule.
The classical and most useful relational logic is **THE LOGIC RL OF BINARY RELATIONS**.

**Formal features of RL**

- Formulas are intended to represent statements saying that two objects are related.
- Relations are specified in the form of relational terms.
- Terms are built from relational variables and relational constants by means of relational operations.
Relational logics – why?

Formal motivation

The relational logic RL is the logical representation of REPRESENTABLE RELATION ALGEBRAS introduced by Tarski in [Tarski 1941].

Representable Relation Algebras RRA:

- Relation algebras that are isomorphic to proper algebras of binary relations.
- Not all relation algebras are representable.
- RRA is not finitely axiomatizable.
- RRA is a discriminator variety with a recursively enumerable but undecidable equational theory.
Possible answer

- Broad applicability.
- Elements of relational structures can be interpreted as possible worlds, points (intervals) of time, states of a computer program, etc.
- We gain compositionality: the relational counterparts of the intensional connectives become compositional, that is the meaning of a compound formula is a function of meaning of its subformulas.
- It enables us to express an interaction between information about static facts and dynamic transitions between states.
Advantages of the relational logic

- A generic logic suitable for representing within a uniform formalism the three basic components of formal systems: syntax, semantics, and deduction apparatus.

- A general framework for representing, investigating, implementing, and comparing theories with incompatible languages and/or semantics.

- A great variety of logics can be represented within the relational logic, in particular modal, temporal, spatial, information, program, intuitionistic, and many-valued, among others.
Methodology of relational dual tableaux enables us to build proof systems for many non-classical logics in a systematic modular way:

- A dual tableau for the classical relational logic of binary relations is a core of most of the relational proof systems.

- For any particular logic some specific rules are designed and adjoined to the core set of rules.

- Relational dual tableau systems usually do more: they can be used for proving entailment, model checking, and satisfaction in finite models.
## Advantages of relational dual tableaux

- We need not implement each deduction system from the scratch.
- We only extend the core system with a module corresponding to a specific part of a logic under consideration.

## Implementations of relational dual tableaux exist

- A tool for the translation into relational language: [Formisano-Omodeo-Orłowska 06]
- A dual tableau for RL: [Formisano-Nicolosi Asmundo 06]
- Relational dual tableaux for non-classical logics: [Formisano-Nicolosi 06] and [Mora et al 11] (modal logics), [Golińska et al. 08] and [Burrieza et al.09] (order-of-magnitude reasoning logics).
The relational logic RL of binary relations

Language
- object variables: $x, y, z, \ldots$
- relational variables: $P_1, P_2, \ldots$
- relational constants: $1, 1'$
- relational operations: $-, \cup, \cap, ^{-1}, \text{and ;}$

Terms and formulas
- Atomic term: a relational variable or constant
- Compound terms: $-T, T \cup T', T \cap T', T^{-1}, T; T'$
- Formulas: $xTy$
The relational logic RL

Relational model: \( \mathcal{M} = (U, m) \)

- \( U \) – a non-empty set
- \( m(P_i) \) – any binary relation on \( U \)
- \( m(1) = U \times U, m(1') = \text{Id}_U \)
- \( m(-T) = (U \times U) \setminus m(T) \)
- \( m(T \cup T') = m(T) \cup m(T') \)
- \( m(T \cap T') = m(T) \cap m(T') \)
- \( m(T^{-1}) = m(T)^{-1} \)
- \( m(T ; T') = m(T) ; m(T') = \{(x, y) \in U \times U : \exists z \in U((x, z) \in m(T) \land (z, y) \in m(T'))\} \).

Joanna Golińska-Pilarek

Relational Dual Tableaux Foundations and Applications
The relational logic RL

Valuation

Any function \( v \) that assigns object variables to elements from \( U \).

Semantics

- Satisfaction \((\mathcal{M}, v \models xTy)\): \((v(x), v(y)) \in m(T)\)
- Truth \((\mathcal{M} \models xTy)\): satisfaction by all valuations in \( \mathcal{M} \)
- Validity: truth in all models.
Dual tableau for the relational logic RL

Decomposition rules:

(−) \[ \frac{x - Ty}{xTy} \]

(∪) \[ \frac{x(T \cup T')y}{xTy, xT'y} \]

(;) \[ \frac{x(T ; T')y}{xTz, x(T ; T')y | zT'y, x(T ; T')y} \]

z is any variable

(−1) \[ \frac{xT^{-1}y}{yTx} \]

(−∪) \[ \frac{x - (T \cup T')y}{x - Ty | x - T'y} \]

(−;) \[ \frac{x - (T ; T')y}{x - Tz, z - T'y} \]

z is a new variable
Dual tableau for the relational logic RL

Specific rules:

\[(1'1) \quad \frac{xRy}{xRz, xRy \mid y1'z, xRy} \quad (1'2) \quad \frac{xRy}{x1'z, xRy \mid zRy, xRy}\]

\(z\) is any object variable, \(R\) is an atomic term

Axioms:

- \(xTy, x \neg Ty\)
- \(x1y\)
- \(x1'x\)
Basic theorems

Soundness and Completeness

For every RL-formula $\varphi$:

$\varphi$ is RL-valid if and only if $\varphi$ is RL-provable.

The connection between RL and RRA

For every relational term $T$ and for all $x, y$:

$T = 1$ is RRA-valid if and only if $xTy$ is RL-provable.
Example – the proof of $1'$; $R \subseteq R$

\[
\frac{x(-(1'; R) \cup R)y}{x-(1'; R)y, xRy} \quad (\cup)
\]
\[
\frac{x-1'z, z-Ry, xRy}{x1'z, x-1'z, \ldots} \quad (1'2)
\]

\[
\frac{z-Ry, zRy, \ldots}{\text{closed}}
\]

\[
\frac{x1'z, x-1'z, \ldots}{\text{closed}}
\]
Entailment in RL

Fact [Tarski 1941]

\[ R_1 = 1, \ldots, R_n = 1 \text{ imply } R = 1 \]

if and only if

\[ (1 ; -(R_1 \cap \ldots \cap R_n) ; 1) \cup R = 1. \]

Entailment can be expressed in RL:

\[ xR_1 y, \ldots, xR_n y \text{ imply } xRy \]

if and only if

\[ x(1 ; -(R_1 \cap \ldots \cap R_n) ; 1) \cup R)y \text{ is RL-valid.} \]
Problem

Given: a finite $\mathcal{M}$, formula $\varphi$, a valuation $v$.

1. Model checking problem: $\mathcal{M} \models \varphi$?

2. Satisfaction problem: $\mathcal{M}, v \models \varphi$?

How to verify?

- Define the logic $\text{RL}_{\mathcal{M},\varphi}$ coding $\mathcal{M}$ and $\varphi$
- Construct dual tableau for $\text{RL}_{\mathcal{M},\varphi}$

For details see the book [Orłowska-Golińska-Pilarek 2011].
Alternative versions of the relational logic RL

Most of the non-classical logics can be translated either into a fragment or an extension of the relational logic RL.

**Possible fragments of RL**
- without the relational constants 1 and 1’
- some restriction on terms built with the composition operation

**Possible extensions of RL**
- with object constants and/or object operations
- more relational constants and/or relational operations
- additional $n$-ary relational symbols, for $n > 2$

Furthermore: any combination of the above without object/relational variables (only object/relational constants).
Development of a relational semantics for L (e.g., Kripke semantics).

Development of a relational logic $RL_L$ appropriate for a logic L.

Development of a validity preserving translation, $\tau$, from the language of logic L into the language of logic $RL_L$.

Construction of a dual tableau for $RL_L$ such that for every formula $\varphi$ of L:

$\varphi$ is valid in L iff its translation $\tau(\varphi)$ is provable in $RL_L$. 
Example of a relational representation – modal languages

<table>
<thead>
<tr>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>- propositional variables $p_i$ and/or constants $c_i$</td>
</tr>
<tr>
<td>- a set of relational constants</td>
</tr>
<tr>
<td>- a set of relational operations</td>
</tr>
<tr>
<td>- classical connectives: $\neg$, $\lor$, $\land$</td>
</tr>
<tr>
<td>- modal connectives, in particular: $[T]$, $\langle T \rangle$, $\langle\langle T \rangle\rangle$, where $T$ is a relational term.</td>
</tr>
</tbody>
</table>

The set of relational operations may be empty.

Then: relational terms = relational constants.

Formulas are defined as usual.
Examples of modal languages

- standard modal logics: a single relational constant and two modal connectives ($\Box$, $\Diamond$)

- standard temporal logics: two relational constants ($P$, $F$) and four modal connectives ($[P]$, $[F]$, $\langle P \rangle$, $\langle F \rangle$)

- a modal logic for FCA: two relational constants ($R$, $S$) and four modal connectives ($[R]$, $[S]$, $[[R]]$, $[[S]]$)

- dynamic logics of programs: infinitely many relational terms
Models are structures \((W, m)\) such that:

- \(W\) – a non-empty set of states
- \(m(p_i) \subseteq W\), for any propositional variable \(p_i\)
- \(m(c_i) \in W\), for any propositional constant \(c_i\)
- \(m(T) \subseteq W \times W\), for any relational term \(T\).

\(m(T)\) may satisfy some additional constraints (e.g., reflexivity, transitivity, symmetry, etc.)

Satisfaction defined by the induction on the complexity of formulas.

Truth: satisfaction in all states.

Validity: truth in all models.
## Language $RL_L$ for $L$

RL-language plus (if necessary):
- object constants
- relational constants
- relational operations.

The relational language for standard modal logics:

$$RL + \text{the relational constant } R + \text{relational constants } C_{c_i}$$

## Additional assumptions

Relational models of $RL_L$ must also satisfy all the conditions imposed on frames of a modal logic $L$. 

---

Joanna Golińska-Pilarek

Relational Dual Tableaux Foundations and Applications
\( \tau \) translates modal formulas into relational terms:

- \( \tau(p_i) = P_i \cup 1 \), for any propositional variable \( p_i \)
- \( \tau(c_i) = C_{c_i} \cup 1 \), for any propositional constant \( c_i \)
- \( \tau(\neg \varphi) = \neg \tau(\varphi) \)
- \( \tau(\varphi \lor \psi) = \tau(\varphi) \cup \tau(\psi) \)
- \( \tau(\varphi \land \psi) = \tau(\varphi) \cap \tau(\psi) \)

and for every relational constant \( T \) of \( L \):

- \( \tau(\langle T \rangle \varphi) = T \cup \tau(\varphi) \)
- \( \tau([T] \varphi) = - (T \cup - \tau(\varphi)) \)
- \( \tau(\langle\langle T \rangle \rangle \varphi) = -T \cup - \tau(\varphi) \)
- \( \tau(\llbracket T \rrbracket \varphi) = -(-T \cup \tau(\varphi)). \)
Some observations:

- Passing from modal formulas to relational terms we replace propositional variables by relational variables and propositional operations by relational operations.
- Accessibility relations are ‘taken out’ of the modal operation and become arguments of an appropriate relational operation.

The translation \( \tau \) preserves validity:

Translation Theorem

For every L-formula \( \varphi \) and for all object variables \( x \) and \( y \):

\[ \varphi \text{ is L-valid if and only if } x\tau(\varphi)y \text{ is } \text{RL}_L\text{-valid.} \]
A relational dual tableau for a standard modal logic $L$ consists of:

- All the rules and axioms of the classical relational logic $RL$.
- Rules/axioms corresponding to the specific conditions imposed on $L$-frames.

Next step: soundness and completeness proof for $RL_L$.

Then, by the translation and completeness theorems for $RL_L$:

**Relational Representation Theorem**

For every modal formula $\varphi$ and for all object variables $x$ and $y$:

$$\varphi$$ is $L$-valid if and only if $x_\tau(\varphi)y$ is $RL_L$-provable.
Examples of specific rules

The rules for reflexivity, symmetry, and transitivity

For all object symbols $x$ and $y$,

\[
\begin{align*}
\text{(ref } R) & \quad \frac{xRy}{x1'y, xRy} & \text{(sym } R) & \quad \frac{xRy}{yRx} \\
\text{(tran } R) & \quad \frac{xRy}{xRz, xRy | zRy, xRy} & \text{z is any object symbol}
\end{align*}
\]

$R\text{L}_L$-dual tableau extended with the above rules can serve as a sound and complete \textit{proof system} for the modal logics: $K$, $T$, $B$, $S4$, $S5$.

Important remark!

$R\text{L}_L$-dual tableau \textbf{IS NOT} a decision procedure for $L$. 
Relational representation of the logic PDL

Language

Language of RL expanded with * + relational constants $R_i$ corresponding to atomic programs of PDL.

Translation $\tau$

Defined as for standard modal logics with the following additional clauses:

- $\tau(R_i) = R_i$
- $\tau(T \cup S) = \tau(T) \cup \tau(S)$
- $\tau(T^*) = \tau(T)^*$
- $\tau(T ; S) = \tau(T) ; \tau(S)$
- $\tau(\varphi?) = 1' \cap \tau(\varphi)$. 
RL\textsubscript{PDL}-dual tableau can be used for verification of PDL-validity.

**RL\textsubscript{PDL}-dual tableau**

RL-dual tableau + the rules for the operation \(*\).

**The rules for the operation \(*\)**

\[
\begin{align*}
&\text{(*) } \frac{xT^*y}{xT^i y, xT^*y} \\
&\text{(-*) } \frac{x-T^*y}{x-T^0 y \mid \ldots \mid x-T^i y \mid \ldots }
\end{align*}
\]

for any \(i \geq 0\)

where \(T^0 = 1', T^{i+1} = T ; T^i\)

The rule \((-*)\) is infinitary.

**RL\textsubscript{PDL}-dual tableau IS NOT a decision procedure for PDL.**
Relational representation of the intuitionistic logic $\text{INT}$

**Logic INT**

- **INT-language:**
  
  the language of the classical propositional logic.

- **INT-models** are Kripke structures $(U, R, m)$ such that:
  
  - $R$ is a reflexive and transitive relation on $U$.
  - For all $s, s' \in U$:
    
    (her) If $(s, s') \in R$ and $s \in m(p)$, then $s' \in m(p)$.

**Satisfaction**

- $\mathcal{M}, s \models p$ iff $s \in m(p)$

- $\mathcal{M}, s \models \neg \varphi$ iff for every $s' \in U$, if $(s, s') \in R$, then
  
  $\mathcal{M}, s' \not\models \varphi$

- $\mathcal{M}, s \models (\varphi \rightarrow \psi)$ iff for every $s' \in U$, if $(s, s') \in R$ and
  
  $\mathcal{M}, s' \models \varphi$, then $\mathcal{M}, s' \models \psi$. 

Joanna Golińska-Pilarek

Relational Dual Tableaux Foundations and Applications
The relational logic $\text{RL}_{\text{INT}}$

- $\text{RL}_{\text{INT}}$-language = $\text{RL}$-language + a relational constant $R$.
- $\text{RL}_{\text{INT}}$-models are $\text{RL}$-models such that:
  - Relational variables are interpreted as right ideal relations.
  - $R$ is interpreted as a reflexive and transitive relation satisfying:
    \[(\text{her}'') \text{ If } (x, y) \in m(R) \text{ and } (x, z) \in m(P), \text{ then } (y, z) \in m(P).\]

Translation $\tau$

- $\tau(p_i) = P_i$
- $\tau(\neg \varphi) = -(R ; \tau(\varphi))$
- $\tau(\varphi \rightarrow \psi) = -(R ; (\tau(\varphi) \cap \neg \tau(\psi))).$
RL_{INT}-dual tableau can be used for verification of INT-validity.

RL_{INT}-dual tableau

RL-dual tableau + (ref $R$), (tran $R$), (rher’), (ideal).

The specific rules of RL_{INT}

(rher’)

\[
\frac{xP_iy}{zRx, xP_iy | zP_iy, xP_iy}
\]

$z$ is any object variable

(ideal)

\[
\frac{xP_iy}{xP_iZ, xP_iy}
\]

$z$ is any object variable

RL_{INT}-dual tableau IS NOT a decision procedure for INT.
In a similar way we can construct relational dual tableaux for the following logics:

- (almost) all multimodal (temporal, information, dynamic, epistemic, order-of-magnitude reasoning)
- many-valued
- relevant
- fuzzy

For details see the book [Orłowska-Golińska-Pilarek 2011].

**Disadvantage of the general relational methodology**

It does not guarantee that the constructed system will be a decision procedure.

In most cases it is NOT, while logics for which systems are constructed are decidable.
Towards decision procedures

Possible approaches

- Restricted relational language and/or applications of standard RL-rules, for instance:
  - The rule \((;\) cannot precede an application of the rule \((-;\)) and a chosen variable \(z\) must occur in a branch. (Used in systems for simple fragments of RL, see [OGP11].)
  - A relational language is restricted: only special forms of composition terms are allowed; some additional requirements on applications of standard RL-rules are assumed. (Used in systems for those fragments of RL that can be used to express modal and description logics. For details see papers of Cantone, Nicolosi-Asmundo, and Orłowska.)
Towards decision procedures

Possible approaches

- New rules instead of 'bad' rules.
  
  (Used in systems for modal and intuitionistic logics, see [Golińska-Huuskonen-Munoz-Velasco 13].)

- External techniques often used in tableaux: backtracking, backjumping, simplifications.

- Any combination of the above.
An example of relational decision procedures

Relational decision procedures for some modal and intuitionistic logics.

The main features of the approach

- Only restricted composition terms are allowed.
- New rules for the composition operation.
- New rules corresponding to specific properties of the accessibility relation.
- Extra conditions on applications of rules.
- Exactly one finite tree for each formula.
- Each of the systems is not only a base for an algorithm verifying validity of a formula, but is ITSELF a decision procedure, with all the necessary bookkeeping built into the rules.
Relational representation – come back!

Relational language appropriate for expressing modal and intuitionistic logics

- object variables: \{z_0, z_1, \ldots\}
- relational variables: \{P_1, P_2, \ldots\}
- the single relational constant: \(R\)
- relational operations: \{-, \cap, ;\}.

Relational terms

- Relational variables are terms.
- If \(S, T\) are terms, then so are \(-S, S \cap T, (R; T)\).

Relational formulas are of the form \(z_n Tz_0\), for \(n \geq 1\).
Terms and formulas are uniquely ordered.
Relational representation – come back!

Semantics
Relational models, satisfaction, truth, and validity are defined in a standard way.

Thus, models are structures \((U, m)\) and such that:
- Relational variables are interpreted as right ideal relations.
- \(m(R)\) satisfies all the conditions imposed on models of a logic in question.
- \(m\) satisfies the standard conditions of RL-models.

Translation
\(\tau(p_i) = P_i\), for any propositional variable \(p_i\).

Other cases defined as before.
All the systems contain the following rules:

- $(\neg)$, $(\cap)$, $(\neg \cap)$ – old rules in the new fashion
- $(R;)$ – the new rule for terms built with the composition operator

A dual tableau may also contain:

- (ref) – a new rule for reflexive relation $R$
- (tran) – a new rule for transitive relation $R$
- (her) – a new rule for heredity condition.
In the definition of a decomposition tree we additionally assume:

1. Whenever several rules are applicable to a node, the first possible schema from the following list is chosen: (−), (−∩), (∩), (ref), (her), (tran), and (R;). Within the schema, the instance with the minimal formula is applied.

2. The rule (R; ) can be applied to a node provided that it is not a subcopy of any of its predecessor nodes.

3. In a branch the rule (ref) can be applied to a given formula at most once.
The rule \((R;\cdot)\)

\[
\frac{X \cup \{z_k A_m z_0 \mid m \in M\} \cup \{z_k - (R; S_i) z_0 \mid i \in I\} \cup \{z_k (R; T_j) z_0 \mid j \in J\}}{X \cup \{z_k A_m z_0 \mid m \in M\} \cup \{z_k - S_i z_0 \mid i \in I\} \cup \{z_k T_j z_0 \mid i \in I, j \in J\}}
\]

1. \(k \geq 1\),
2. \(z_k Tz_0 \not\in X\),
3. \(M, I, J\) are sets of indices, \(I \neq \emptyset\),
4. \(A_m\) is a literal and \(S_i, T_j\) are terms,
5. \(N = \{k_i \mid i \in I\}\) is the set of consecutive natural numbers that do not occur in the premise.
Specific rules

(ref) \[ X \cup \{ z_k(R^s; T)z_0 \} \]
\[ X \cup \{ z_k(R^s; T)z_0 \} \cup \{ z_k(R^i; T)z_0 \mid i \in \{0, \ldots, s-1\} \} \]

\( T \) is a non-compositional term,
For all \( t > s \), it holds that \( z_k(R^t; T)z_0 \notin X \).

(tran) \[ X \cup \{ z_k(R; T)z_0 \} \]
\[ X \cup \{ z_k(R; T)z_0 \} \cup \{ z_k(R^2; T)z_0 \} \]

\( T \) is a non-compositional term.

(her) \[ X \cup \{ z_k-(R; T)z_0 \} \cup \{ z_k-P_iz_0 \mid i \in I \} \]
\[ X \cup \{ z_k-(R; T)z_0 \} \cup \{ z_k-P_i \mid i \in I \} \cup \{ z_k(R; -P_i)z_0 \mid i \in I \} \]

\( z_k-Pz_0 \notin X \) for any relational variable \( P \).
Termination and uniqueness

Every formula has exactly one finite tree.

Soundness and completeness

For every relational formula \( \varphi \):

\( \varphi \) is valid if and only if \( \varphi \) is provable.

Deterministic decision procedures

An RL\( _L \)-dual tableau is a sound and complete deterministic decision procedure for a logic L.
Current and future works

- Does the rule \((R;:)\) work for relational languages with more than one relational constant?
- Can be this approach extended to other semantic conditions?
- Can be this approach extended to other (more complex) fragments of RL?
- Is there any general and modular way to construct a relational decision procedure?
La ringrazio molto!


