A Tree Rewriting System for the Reflection Calculus

Sofía Santiago-Fernández [1] Joost J. Joosten [2]
David Fernández-Duque [3]

Universitat de Barcelona
Barcelona, Spain

Abstract

The *Reflection Calculus* (**RC**, c.f. [6], [8]) is the fragment of the polymodal logic **GLP** in the language $\mathcal{L}^+$ whose formulas are built up from $\top$ and propositional variables using conjunction and diamond modalities. **RC** is complete with respect to the arithmetical interpretation that associates modalities with reflection principles and has various applications in proof theory, specifically ordinal analysis.
We present TRC, a tree rewriting system (c.f. [2]) that is adequate and complete with respect to **RC**, designed to simulate **RC** derivations. TRC is based on a given correspondence between formulas of $\mathcal{L}^+$ and modal trees $\mathsf{Tree}^\diamond$. Modal trees are presented as an inductive type (c.f. [9], [16]) allowing precise positioning and transformations which give rise to the formal definition of rewriting rules and facilitates formalization in proof assistants. Furthermore, we provide a rewrite normalization theorem for systematic rule application. The normalization of the rewriting process enhances proof search efficiency and facilitates implementation (c.f. [20], [12], [15]).
By providing TRC as an efficient provability tool for **RC**, we aim to help on the study of various aspects of the logic such as the subformula property and rule admissibility.

Keywords: Strictly positive logics, Reflection Calculus, tree rewriting systems, abstract rewriting systems, polymodal logics, normalized rewriting, proof normalization, proof search, proof theory, type theory.

1 Introduction

Modal logics provide an attractive alternative to first or higher order logic for computational applications, largely due to the fact that they often enjoy a decidable consequence relation while remaining expressive enough to describe intricate processes. However, decidability alone does not suffice for practical implementation when complexity comes at a hefty price tag; even propositional

[1] sofia.santiago@ub.edu
[2] jjoosten@ub.edu
[3] fernandez-duque@ub.edu

logic is NP-complete, which quickly becomes intractable as formula size and especially the number of variables is large.

This is no longer an issue when working in *strictly positive* fragments (see e.g. [14]), which in contrast enjoy a polynomially decidable consequence relation. Strictly positive formulas do not contain negation and instead are built from atoms and $\top$ using conjunction and $\Diamond$ (or, more generally, a family of modalities $\langle i \rangle$ indexed by i in some set I). Strictly positive formulas tend to be contingent, so validity and satisfiability are no longer the most central problems, but the consequence relation is indeed useful for example for reasoning about ontologies and is the basis for some description logics [1].

One remarkable success story for strictly positive logics comes from the *reflection calculus* (**RC**) [8,7]. Beklemishev has shown how Japaridze's polymodal provability logic **GLP** [13] can be used to perform a proof-theoretic analysis of Peano aritmetic and its fragments [5]; however, the logic **GLP** is notoriously difficult to work with, especially as it is not Kripke-complete. In contrast, its strictly positive fragment is rather tame from both a theoretical and computational point of view, yet suffices for the intended proof-theoretic applications.

The current work is inspired by two distinct ideas that have arisen in the study of strictly positive logics. The first is the tree representation of formulas, which yield a way to decide strictly positive implications. This was developed by Kikot et al. [14] in a general setting and by Beklemishev [7] in the context of **RC**. In both cases, one assigns to each strictly positive formula φ a finite, tree-like Kripke model $T(\varphi)$ with the crucial property that $\varphi \to \psi$ is valid if and only if $T(\varphi) \models \psi$. Thus the study of strictly positive fragments can be reduced to the study of their tree-like Kripke models.

The second is the connection of strictly positive calculi to term rewrite systems. Strictly positive formulas and, particularly, those built exclusively from $\top$ and the modalities $\langle i \rangle$, may be regarded as *words* (or 'worms'). This has prompted Beklemishev [4] to view strictly positive fragments as term-rewriting systems [2], but connections between such systems and modal logic are not new and can be traced back to Foret [10].

Term rewriting is a discipline that integrates elements of logic, universal algebra, automated theorem proving, and functional programming. It has applications in algebra (e.g. Boolean algebra), recursion theory (computability of rewriting rules), software engineering and programming languages (especially functional and logic programming [19]), with the λ-calculus perhaps being the most familiar example [3]. Of particular interest to us, tree rewriting systems [11] are term rewriting systems such that terms are trees.

When terms represent formulas, rewrite rules are similar to deep inference rules, i.e. rules which may be applied to strict subformulas of the displayed formulas. This is the approach taken by Shamkanov [17] for developing a cut-free calculus for **GLP**. As is the case for other technical differences between **GLP** and the reflection calculus, our tree rewrite system makes up for the loss in expressive power with increased simplicity and transparent Kripke semantics.

Our approach is to recast **RC** as an abstract rewriting system in which terms are trees. In the parlance of rewrite systems, cut-elimination can be viewed as a normalization procedure for derivations. In our setting we do not have an analogue of the cut rule, but we do obtain a rewriting normalization theorem which states that the rewriting process can be consistently and efficiently executed by grouping rewriting rules by their kinds and applying them in a designated sequence. By doing so, it enhances our comprehension of the dynamics of the tree rewriting system, offering valuable insights into the nature of the rewriting process and the interplay among rules. Moreover, it furnishes an efficient framework for proof search methodologies. Thanks to the normalization theorem, the need for exhaustive exploration is minimized by focusing on normalized rewriting sequences, which mitigates the risk of redundancy in rewriting. Consequently, when searching for a proof of a certain result, we only need to consider the normalized derivations, thereby reducing the proof search space and improving computational efficiency [12]. Furthermore, it serves as a practical guide for implementing the rewriting process in proof assistants [15].

In our presentation, we make use of the inductive type of lists within the framework of type theory (cf. [16], [9]) to define the trees in our tree rewriting system. The use of lists allows to define inductive structures with an order, facilitating the specification of internal positions and transformations for rewriting systems, and its formalization in proof assistants.

Since lists play such a central role in our work, we conclude this introduction by establishing some notation. A list of elements of type $\mathcal{A}$ is either the empty list $\varnothing$ or $[x] \frown L$ for x an element of type $\mathcal{A}$, a list L of elements of type $\mathcal{A}$ and $\frown$ the operator of concatenation of lists. We write $x \frown L$ and $L \frown x$ to denote $[x] \frown L$ and $L \frown [x]$, respectively. The length of a list L of elements of type $\mathcal{A}$ is denoted by $|L|$.

2 From $\mathbf{K}^+$ to RC

In this section we present the basic sequent-style system $\mathbf{K}^+$ for the language of strictly positive formulae, concluding by an introduction to the Reflection Calculus (**RC**) as an extension of $\mathbf{K}^+$.

We consider the language of strictly positive formulae $\mathcal{L}^+$ composed from propositional variables p,q,..., in Prop, the constant $\top$, and connectives $\wedge$ for conjunction and $\langle\alpha\rangle$ for diamond modalities for each ordinal $\alpha \in \omega$. Formally, the strictly positive formulae φ of $\mathcal{L}^+$ are generated by the following grammar:

$$\varphi ::= \top \mid p \mid \langle\alpha\rangle\varphi \mid (\varphi \wedge \varphi), \quad \alpha \in \omega \text{ and } p \in \mathsf{Prop}.$$

The *modal depth* of φ, denoted by $\mathsf{md}(\varphi)$, is recursively defined as $\mathsf{md}(\top) := 0$, $\mathsf{md}(p) := 0$ for $p \in \mathsf{Prop}$, $\mathsf{md}(\langle\alpha\rangle\varphi) := \mathsf{md}(\varphi) + 1$ and $\mathsf{md}(\varphi \wedge \psi) := \max\{\mathsf{md}(\varphi), \mathsf{md}(\psi)\}$.

Sequents are expressions of the form $\varphi \vdash \psi$ for $\varphi, \psi \in \mathcal{L}^+$. If L is a logic, we write $\varphi \vdash_{\mathsf{L}} \psi$ for the statement that $\varphi \vdash \psi$ is provable in L. We write $\varphi \equiv_{\mathsf{L}} \psi$ to denote $\varphi \vdash_{\mathsf{L}} \psi$ and $\psi \vdash_{\mathsf{L}} \varphi$.

Polymodal **K** can be readily adapted to its strictly positive variant, where most notably the necessitation rule is replaced by distribution for each diamond modality.

Definition 2.1 ($\mathbf{K}^+$, [4]) The basic sequent-style system $\mathbf{K}^+$ is given by the following axioms and rules:

$\varphi \vdash_{\mathbf{K}^+} \varphi$; $\varphi \vdash_{\mathbf{K}^+} \top$;

if $\varphi \vdash_{\mathbf{K}^+} \psi$ and $\psi \vdash_{\mathbf{K}^+} \phi$ then $\varphi \vdash_{\mathbf{K}^+} \phi$ (cut);

$\varphi \wedge \psi \vdash_{\mathbf{K}^+} \varphi$ and $\varphi \wedge \psi \vdash_{\mathbf{K}^+} \psi$ (elimination of conjunction);

if $\varphi \vdash_{\mathbf{K}^+} \psi$ and $\varphi \vdash_{\mathbf{K}^+} \phi$ then $\varphi \vdash_{\mathbf{K}^+} \psi \wedge \phi$ (introduction to conjunction);

if $\varphi \vdash_{\mathbf{K}^+} \psi$ then $\langle\alpha\rangle\varphi \vdash_{\mathbf{K}^+} \langle\alpha\rangle\psi$ (distribution).

For Π a finite list of strictly positive formulae, $\bigwedge \Pi$ is defined by $\top$ for $\Pi = \varnothing$ and $\varphi \wedge \bigwedge \hat{\Pi}$ for $\Pi = \varphi \frown \hat{\Pi}$. Note that $\bigwedge(\Pi_1 \frown \Pi_2) \equiv_{\mathbf{K}^+} \bigwedge \Pi_1 \wedge \bigwedge \Pi_2$ for Π_1 and Π_2 finite lists of strictly positive formulae.

A diamond modality can be distributed over a conjunction of formulas for $\mathbf{K}^+$ as follows.

Lemma 2.2 $\langle\alpha\rangle(\varphi_1 \wedge ... \wedge \varphi_n) \vdash_{\mathbf{K}^+} \langle\alpha\rangle\varphi_1 \wedge ... \wedge \langle\alpha\rangle\varphi_n$.

Proof. By an easy induction on n. □

We aim to define a tree rewriting system adequate and complete w.r.t. the Reflection Calculus, the strictly positive fragment of Japaridze's polymodal logic formulated as an extension of $\mathbf{K}^+$.

Definition 2.3 (**RC**, [6], [8]) The *Reflection Calculus* (**RC**) is the strictly positive modal logic extending $\mathbf{K}^+$ by the following axioms:

$\langle\alpha\rangle\langle\alpha\rangle\varphi \vdash_{\mathbf{RC}} \langle\alpha\rangle\varphi$ (transitivity);

$\langle\alpha\rangle\varphi \vdash_{\mathbf{RC}} \langle\beta\rangle\varphi$, $\alpha > \beta$ (monotonicity);

$\langle\alpha\rangle\varphi \wedge \langle\beta\rangle\psi \vdash_{\mathbf{RC}} \langle\alpha\rangle(\varphi \wedge \langle\beta\rangle\psi)$, $\alpha > \beta$ (J).

3 Modal trees

In this section we present modal trees, a concrete set of inductively defined trees on which our rewriting system is based, and the corresponding framework for their manipulation. Modal trees are finite labeled trees with nodes labeled with lists of propositional variables and edges labeled with ordinals less than ω. Specifically, modal trees can be regarded as tree-like Kripke models of the form $(\mathcal{W}, \{R_\alpha\}_{\alpha\in\omega}, v)$ such that an ordinal α labels an edge if R_α relates the corresponding nodes, and a list of propositional variables labels a node if its elements are the only propositional variables being true under the valuation v in that node. However, for technical convenience, both in presenting the rewrite system and in formalizing our results in a proof assistant, it will be convenient to present modal trees as inductively defined structures. In particular, the children of a node of a modal tree are given by lists instead of sets, providing a default ordering on its children useful for unambiguously determining positions in the tree.

Definition 3.1 ($\mathsf{Tree}^\diamond$) The set of modal trees $\mathsf{Tree}^\diamond$ is defined recursively to be the set of pairs $\langle \Delta ; \Gamma \rangle$ where Δ is a finite list of propositional variables and Γ is a finite list of pairs $(\alpha, \mathtt{T})$, with $\alpha < \omega$ and $\mathtt{T} \in \mathsf{Tree}^\diamond$.

Elements of $\mathsf{Tree}^\diamond$ will be denoted by $\mathtt{T}$ and $\mathtt{S}$. Note that we employ distinct notations to enhance clarity on wether a pair is a modal tree: $\langle \cdot ; \cdot \rangle$ is used for a pair representing a modal tree, while $(\cdot, \cdot)$ denotes a pair comprising an ordinal and a modal tree. The *root* of a modal tree $\langle \Delta ; \Gamma \rangle$ is Δ and its *children* is the list $[\mathtt{S} | (\alpha, \mathtt{S}) \in \Gamma]$. Note that, in general we write $[f(\alpha, \mathtt{S}) | (\alpha, \mathtt{S}) \in \Gamma]$ to denote the list $[f(\alpha_1, \mathtt{S}_1), ..., f(\alpha_n, \mathtt{S}_n)]$ for $\Gamma = [(\alpha_1, \mathtt{S}_1), ..., (\alpha_n, \mathtt{S}_n)]$, $n \geq 0$ and f a function of domain $\mathsf{Ord}^{<\omega} \times \mathsf{Tree}^\diamond$. For the sake of readability we write $\gamma \in \Gamma$ and $\mathtt{T} \in \Gamma$ to denote $\gamma \in [\alpha | (\alpha, \mathtt{S}) \in \Gamma]$ and $\mathtt{T} \in [\mathtt{S} | (\alpha, \mathtt{S}) \in \Gamma]$ respectively, since the context permits a clear distinction. A modal tree is called a *leaf* if it has an empty list of children. The *height* of a modal tree $\mathtt{T}$, denoted by $\mathsf{h}(\mathtt{T})$, is inductively defined as $\mathsf{h}(\langle \Delta ; \varnothing \rangle) := 0$ and $\mathsf{h}(\langle \Delta ; \Gamma \rangle) := \mathsf{max}[\mathsf{h}(\mathtt{S}) | \mathtt{S} \in \Gamma] + 1$.

The sum of modal trees is the tree obtained by concatenating their roots and children.

Definition 3.2 The *sum* of modal trees $\mathtt{T}_1 = \langle \Delta_1 ; \Gamma_1 \rangle$ and $\mathtt{T}_2 = \langle \Delta_2 ; \Gamma_2 \rangle$ is defined as $\mathtt{T}_1 + \mathtt{T}_2 := \langle \Delta_1 \frown \Delta_2 ; \Gamma_1 \frown \Gamma_2 \rangle$.

More generally, for Λ a finite list of modal trees, $\sum \Lambda$ is defined as $\langle \varnothing ; \varnothing \rangle$ if $\Lambda = \varnothing$ and $\mathtt{T} + \sum \hat{\Lambda}$ if $\Lambda = \mathtt{T} \frown \hat{\Lambda}$. Note that $\mathsf{h}(\mathtt{T}_1 + \mathtt{T}_2) = \mathsf{max}\{\mathsf{h}(\mathtt{T}_1), \mathsf{h}(\mathtt{T}_2)\}$ for $\mathtt{T}_1, \mathtt{T}_2 \in \mathsf{Tree}^\diamond$.

A standard numbering of the nodes of the tree by strings of positive integers allows us to refer to positions in a tree. Specifically, the set of positions of a tree includes the root position, defined as the empty string, and the positions from its children which are obtained by appending the order of each child to its positions.

Definition 3.3 (Set of positions) The set of positions of a modal tree $\mathtt{T} = \langle \Delta ; \Gamma \rangle$, denoted by $\mathsf{Pos}(\mathtt{T}) \in \mathcal{P}(\mathbb{N}^*)$, is inductively defined as

- $\mathsf{Pos}(\langle \Delta ; \varnothing \rangle) := \{\epsilon\}$ for $\epsilon \in \mathbb{N}^*$ the empty string,
- $\mathsf{Pos}(\langle \Delta ; \Gamma \rangle) := \{\epsilon\} \cup \bigcup_{i=1}^{n} \{i\mathbf{k} | \mathbf{k} \in \mathsf{Pos}(\mathtt{S}_i)\}$ for $\Gamma = [(\alpha_1, \mathtt{S}_1), ..., (\alpha_n, \mathtt{S}_n)]$.

Using the precise position apparatus we can define derived notions like, for example, that of subtree.

Definition 3.4 (Subtree) The subtree of $\mathtt{T} \in \mathsf{Tree}^\diamond$ at a position $\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$, denoted by $\mathtt{T}|_{\mathbf{k}}$, is inductively defined over the length of $\mathbf{k}$ as

- $\mathtt{T}|_\epsilon := \mathtt{T}$,
- $\mathtt{T}|_{i\mathbf{r}} := \mathtt{S}_i|_{\mathbf{r}}$ for $1 \leq i \leq n$ such that $\mathtt{T} = \langle \Delta ; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_n, \mathtt{S}_n)] \rangle$.

We can now define subtree replacement based on the precise positioning provided.

Definition 3.5 (Replacement) Let $\mathtt{T}, \mathtt{S} \in \mathsf{Tree}^\diamond$ and $\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$. The *tree obtained from* $\mathtt{T}$ *by replacing the subtree at position* $\mathbf{k}$ *by* $\mathtt{S}$, denoted by $\mathtt{T}[\mathtt{S}]_{\mathbf{k}}$, is inductively defined over the length of $\mathbf{k}$ as

- $\mathtt{T}[\mathtt{S}]_\epsilon := \mathtt{S}$,
- $\mathtt{T}[\mathtt{S}]_{i\mathbf{r}} := \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}_i[\mathtt{S}]_\mathbf{r}), ..., (\alpha_n, \mathtt{S}_n)] \rangle$ for $1 \leq i \leq n$ and $\mathtt{T} = \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_n, \mathtt{S}_n)] \rangle$.

Here below we present useful results on positioning and replacement in a modal tree.

Lemma 3.6 *Let* $\mathtt{T}, \mathtt{S}, \hat{\mathtt{S}} \in \mathsf{Tree}^\diamond$ *be modal trees. Then, for* $\mathbf{k}$ *and* $\mathbf{r}$ *belonging to the adequate position sets we have*

(i) $(\mathtt{T}|_\mathbf{k})|_\mathbf{r} = \mathtt{T}|_{\mathbf{kr}}$;

(ii) $\mathtt{T}[\mathtt{T}|_\mathbf{k}]_\mathbf{k} = \mathtt{T}$;

(iii) $(\mathtt{T}[\mathtt{S}]_\mathbf{k})|_\mathbf{k} = \mathtt{S}$;

(iv) $(\mathtt{T}[\hat{\mathtt{S}}]_\mathbf{k})[\mathtt{S}]_\mathbf{k} = \mathtt{T}[\mathtt{S}]_\mathbf{k}$ *(transitivity of replacement);*

(v) $(\mathtt{T}[\hat{\mathtt{S}}]_\mathbf{k})[\mathtt{S}]_{\mathbf{kr}} = \mathtt{T}[\hat{\mathtt{S}}[\mathtt{S}]_\mathbf{r}]_\mathbf{k}$.

Proof. We proceed by induction on the tree structure of $\mathtt{T}$ for each statement. If $\mathtt{T}$ is a leaf, the results follow easily since $\mathbf{k} = \epsilon$. Otherwise, we continue by cases on the length of $\mathbf{k}$. For ϵ the statements trivially hold. Finally consider $i\hat{\mathbf{k}}$ for $1 \leq i \leq n$ and $\hat{\mathbf{k}} \in \mathsf{Pos}(\mathtt{S}_i)$ such that $\mathtt{T} = \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_n, \mathtt{S}_n)] \rangle$. Then, by definition and each statement's inductive hypothesis for $\mathtt{S}_i$, we conclude

1. $(\mathtt{T}|_{i\hat{\mathbf{k}}})|_\mathbf{r} = (\mathtt{S}_i|_{\hat{\mathbf{k}}})|_\mathbf{r} = \mathtt{S}_i|_{\hat{\mathbf{k}}\mathbf{r}} = \mathtt{T}|_{i\hat{\mathbf{k}}\mathbf{r}}$;
2. $\mathtt{T}[\mathtt{T}|_{i\hat{\mathbf{k}}}]_{i\hat{\mathbf{k}}} = \mathtt{T}[\mathtt{S}_i|_{\hat{\mathbf{k}}}]_{i\hat{\mathbf{k}}} = \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}_i[\mathtt{S}_i|_{\hat{\mathbf{k}}}]_{\hat{\mathbf{k}}}), ..., (\alpha_n, \mathtt{S}_n)] \rangle = \mathtt{T}$;
3. $(\mathtt{T}[\mathtt{S}]_{i\hat{\mathbf{k}}})|_{i\hat{\mathbf{k}}} = (\langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}_i[\mathtt{S}]_{\hat{\mathbf{k}}}), ..., (\alpha_n, \mathtt{S}_n)] \rangle)|_{i\hat{\mathbf{k}}} = (\mathtt{S}_i[\mathtt{S}]_{\hat{\mathbf{k}}})|_{\hat{\mathbf{k}}} = \mathtt{S}$;
4. $$\begin{aligned}(\mathtt{T}[\hat{\mathtt{S}}]_{i\hat{\mathbf{k}}})[\mathtt{S}]_{i\hat{\mathbf{k}}} &= (\langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}_i[\hat{\mathtt{S}}]_{\hat{\mathbf{k}}}), ..., (\alpha_n, \mathtt{S}_n)] \rangle)[\mathtt{S}]_{i\hat{\mathbf{k}}} \\ &= \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, (\mathtt{S}_i[\hat{\mathtt{S}}]_{\hat{\mathbf{k}}})[\mathtt{S}]_{\hat{\mathbf{k}}}), ..., (\alpha_n, \mathtt{S}_n)] \rangle \\ &= \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}_i[\mathtt{S}]_{\hat{\mathbf{k}}}), ..., (\alpha_n, \mathtt{S}_n)] \rangle = \mathtt{T}[\mathtt{S}]_{i\hat{\mathbf{k}}};\end{aligned}$$
5. $$\begin{aligned}(\mathtt{T}[\hat{\mathtt{S}}]_{i\hat{\mathbf{k}}})[\mathtt{S}]_{i\hat{\mathbf{k}}\mathbf{r}} &= (\langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}_i[\hat{\mathtt{S}}]_{\hat{\mathbf{k}}}), ..., (\alpha_n, \mathtt{S}_n)] \rangle)[\mathtt{S}]_{i\hat{\mathbf{k}}\mathbf{r}} \\ &= \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, (\mathtt{S}_i[\hat{\mathtt{S}}]_{\hat{\mathbf{k}}})[\mathtt{S}]_{\hat{\mathbf{k}}\mathbf{r}}), ..., (\alpha_n, \mathtt{S}_n)] \rangle \\ &= \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}_i[\hat{\mathtt{S}}[\mathtt{S}]_\mathbf{r}]_{\hat{\mathbf{k}}}), ..., (\alpha_n, \mathtt{S}_n)] \rangle = \mathtt{T}[\hat{\mathtt{S}}[\mathtt{S}]_\mathbf{r}]_{i\hat{\mathbf{k}}}.\end{aligned}$$

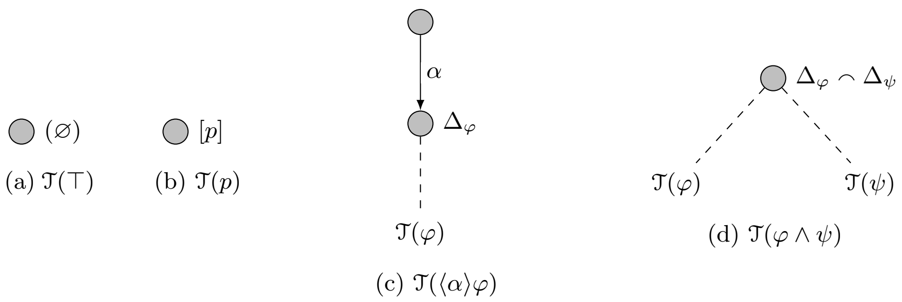
□

4 Relating formulas and modal trees

Our tree rewriting system is based on a correspondence between the language of $\mathcal{L}^+$ and $\mathsf{Tree}^\diamond$. Thereby we can ensure that transformations within the structure of modal trees accurately simulate derivations in a considered proof system. For this purpose, we introduce the tree embedding operator $\mathcal{T}$ inductively defined over the set of strictly positive formulas mapping them to modal trees. This definition is inspired by the canonical tree representation of strictly positive formulae presented by Beklemishev (see [7]) as a combinatiorial tool for proving the polytime decidability of **RC**. Additionally, we define $\mathcal{F}$ mapping modal trees to formulas. Ultimately, we prove that composition $\mathcal{T} \circ \mathcal{F}$ serves as the identity over $\mathsf{Tree}^\diamond$, while $\mathcal{F} \circ \mathcal{T}$ acts as the identity on $\mathcal{L}^+$ modulo equality for **K**+.

Definition 4.1 ($\mathcal{T}$) The modal tree embedding is the function $\mathcal{T} : \mathcal{L}^+ \longrightarrow \mathsf{Tree}^\diamond$ inductively defined over the structure of strictly positive formulae as

- $\mathcal{T}(\top) := \langle \varnothing; \varnothing \rangle$,
- $\mathcal{T}(p) := \langle [p]; \varnothing \rangle$ for $p \in \mathsf{Prop}$,
- $\mathcal{T}(\langle \alpha \rangle \varphi) := \langle \varnothing; [(\alpha, \mathcal{T}(\varphi))] \rangle$ for $\varphi \in \mathcal{L}^+$,
- $\mathcal{T}(\varphi \wedge \psi) := \mathcal{T}(\varphi) + \mathcal{T}(\psi)$ for $\varphi, \psi \in \mathcal{L}^+$.



(a) $\mathcal{T}(\top)$ (b) $\mathcal{T}(p)$ (c) $\mathcal{T}(\langle \alpha \rangle \varphi)$ (d) $\mathcal{T}(\varphi \wedge \psi)$

Fig. 1. Modal tree embedding

The modal depth of a formula coincides with the height of the modal tree which it is mapped to.

Lemma 4.2 $\mathsf{h}(\mathcal{T}(\varphi)) = \mathsf{md}(\varphi)$ *for* $\varphi \in \mathcal{L}^+$.

Proof. By an easy induction on the structure of φ. □

We also introduce a corresponding embedding in the opposite direction.

Definition 4.3 ($\mathcal{F}$) The strictly positive formulae embedding is the function $\mathcal{F} : \mathsf{Tree}^\diamond \longrightarrow \mathcal{L}^+$ defined as

$$\mathcal{F}(\langle \Delta; \Gamma \rangle) := \bigwedge \Delta \wedge \bigwedge [\langle \alpha \rangle \mathcal{F}(\mathtt{S}) | (\alpha, \mathtt{S}) \in \Gamma].$$

For the sake of readability, we will simply write $\diamond\Gamma$ to denote $[\langle \alpha \rangle \mathcal{F}(\mathtt{S}) | (\alpha, \mathtt{S}) \in \Gamma]$.

We conclude this section by providing a relation among strictly positive formulas and modal trees through the composition of the embeddings.

Proposition 4.4 (Embedding composition) $\mathcal{T} \circ \mathcal{F} = id_{\mathsf{Tree}^\diamond}$ *and* $\mathcal{F} \circ \mathcal{T} = id_{\mathcal{L}^+/\equiv_{\mathbf{K}^+}}$.

Proof. We firstly prove $\mathcal{T} \circ \mathcal{F}(\mathtt{T}) = \mathtt{T}$ for $\mathtt{T} \in \mathsf{Tree}^\diamond$ by induction on the modal tree structure. The leaf case follows by definition since $\mathcal{T}(\mathcal{F}(\langle \Delta; \varnothing \rangle)) = \mathcal{T}(\bigwedge \Delta \wedge \top) = \langle \Delta; \varnothing \rangle$. Otherwise, assuming $\mathcal{T}(\mathcal{F}(\mathtt{S})) = \mathtt{S}$ for every $\mathtt{S} \in \Gamma$, we conclude

$$\begin{aligned}
\mathcal{T}(\mathcal{F}(\langle \Delta; \Gamma \rangle)) &= \mathcal{T}(\bigwedge \Delta \wedge \bigwedge \diamond\Gamma) = \mathcal{T}(\bigwedge \Delta) + \mathcal{T}(\bigwedge \diamond\Gamma) \\
&= \langle \Delta; \varnothing \rangle + \sum [\mathcal{T}(\langle \alpha \rangle \mathcal{F}(\mathtt{S})) | (\alpha, \mathtt{S}) \in \Gamma] \\
&= \langle \Delta; \varnothing \rangle + \sum [\langle \varnothing; [(\alpha, \mathcal{T}(\mathcal{F}(\mathtt{S})))] \rangle | (\alpha, \mathtt{S}) \in \Gamma] \\
&= \langle \Delta; \varnothing \rangle + \sum [\langle \varnothing; [(\alpha, \mathtt{S})] \rangle | (\alpha, \mathtt{S}) \in \Gamma] = \langle \Delta; \Gamma \rangle.
\end{aligned}$$

Finally, we prove $\mathcal{F} \circ \mathcal{T}(\varphi) \equiv_{\mathbf{K}^+} \varphi$ for $\varphi \in \mathcal{L}^+$ by induction on the structure of φ.

- $\mathcal{F} \circ \mathcal{T}(\top) = \top \wedge \top \equiv_{\mathbf{K}^+} \top$; $\mathcal{F} \circ \mathcal{T}(p) = (p \wedge \top) \wedge \top \equiv_{\mathbf{K}^+} p$.
- Let us assume $\mathcal{F} \circ \mathcal{T}(\psi) \equiv_{\mathbf{K}^+} \psi$. Then,

$$\mathcal{F} \circ \mathcal{T}(\langle \alpha \rangle \psi) = \mathcal{F}(\langle \varnothing; [(\alpha, \mathcal{T}(\psi))] \rangle = \top \wedge \langle \alpha \rangle \mathcal{F} \circ \mathcal{T}(\psi) \wedge \top \equiv_{\mathbf{K}^+} \langle \alpha \rangle \psi.$$

- Let $\mathcal{T}(\psi) = \langle \Delta_\psi; \Gamma_\psi \rangle$ and $\mathcal{T}(\phi) = \langle \Delta_\phi; \Gamma_\phi \rangle$. Assuming $\mathcal{F} \circ \mathcal{T}(\psi) \equiv_{\mathbf{K}^+} \psi$ and $\mathcal{F} \circ \mathcal{T}(\phi) \equiv_{\mathbf{K}^+} \phi$,

$$\mathcal{F} \circ \mathcal{T}(\psi \wedge \phi) \equiv_{\mathbf{K}^+} \bigwedge \Delta_\psi \wedge \bigwedge \diamond\Gamma_\psi \wedge \bigwedge \Delta_\phi \wedge \bigwedge \diamond\Gamma_\phi \equiv_{\mathbf{K}^+} \psi \wedge \phi.$$

$\square$

5 The tree rewriting system for RC

We introduce the tree rewriting system for **RC**, an abstract rewriting system for $\mathsf{Tree}^\diamond$ which will be proven adequate and complete w.r.t. **RC** in the next section. Additionally, we present useful results for the rewriting, along with the Inside Rewriting Property which involves transforming a subtree while preserving the remaining parts invariant.

An *abstract rewriting system* is a pair $(A, \{\hookrightarrow^\mu\}_{\mu \in I})$ consisting of a set A and a sequence of binary relations $\hookrightarrow^\mu$ on A, also called rewriting rules (r.r.). Instead of $(a, b) \in \hookrightarrow^\mu$ we write $a \hookrightarrow^\mu b$ and call b *is obtained by applying* μ *to* a or b *is obtained by performing one step* μ *to* a. The composition of rewriting rules μ_1 and μ_2 is written $a \hookrightarrow^{\mu_1} \circ \hookrightarrow^{\mu_2} b$ and denotes that there is $\hat{a} \in A$ such that $a \hookrightarrow^{\mu_1} \hat{a} \hookrightarrow^{\mu_2} b$.

In particular, the rewriting rules of the tree rewriting system for **RC** transform a tree by replacing a subtree with a predetermined tree. The rules are classified into five kinds according to the performed transformation: atomic, structural, replicative, decreasing, and modal rewriting rules. Atomic rules duplicate or eliminate propositional variables in the lists labeling the nodes; the structural rule permutes the order of a node's children; the replicative rule duplicates a child of a node; decreasing rules either eliminate a child or remove a node and its children under certain conditions; and modal rules either decrease the label of an edge or apply a transformation simulating the J axiom of **RC**.

Atomic r.r.	$\mathtt{T} \hookrightarrow^{\rho^+} \mathtt{T}[\langle \Delta^{+i}; \Gamma \rangle]_{\mathbf{k}}$	ρ^+-rule
$0 < i \leq \lvert\Delta\rvert$	$\mathtt{T} \hookrightarrow^{\rho^-} \mathtt{T}[\langle \Delta^{-i}; \Gamma \rangle]_{\mathbf{k}}$	ρ^--rule
Structural r.r. $0 < i, j \leq \lvert\Gamma\rvert$, $i \neq j$	$\mathtt{T} \hookrightarrow^{\sigma} \mathtt{T}[\langle \Delta; \Gamma^{i \leftrightarrow j} \rangle]_{\mathbf{k}}$	σ-rule
Replicative r.r. $0 < i \leq \lvert\Gamma\rvert$	$\mathtt{T} \hookrightarrow^{\pi^+} \mathtt{T}[\langle \Delta; \Gamma^{+i} \rangle]_{\mathbf{k}}$	π^+-rule
Decreasing r.r.	$\mathtt{T} \hookrightarrow^{\pi^-} \mathtt{T}[\langle \Delta; \Gamma^{-i} \rangle]_{\mathbf{k}}$	π^--rule
$0 < i \leq \lvert\Gamma\rvert$ $0 < j \leq \lvert\tilde{\Gamma}\rvert$	$\mathtt{T} \hookrightarrow^{4} \mathtt{T}[\langle \Delta; \Gamma[(\beta, \mathtt{S})]_i \rangle]_{\mathbf{k}}$ for $\#_i\Gamma = (\beta, \langle \tilde{\Delta}; \tilde{\Gamma} \rangle)$ and $\#_j\tilde{\Gamma} = (\beta, \mathtt{S})$	4-rule
Modal r.r. $\alpha > \beta$	$\mathtt{T} \hookrightarrow^{\lambda} \mathtt{T}[\langle \Delta; \Gamma[(\beta, \mathtt{S})]_i \rangle]_{\mathbf{k}}$ for $\#_i\Gamma = (\alpha, \mathtt{S})$	λ-rule
$0 < i, j \leq \lvert\Gamma\rvert$ $i \neq j$	$\mathtt{T} \hookrightarrow^{\mathsf{J}} \mathtt{T}[\langle \Delta; (\Gamma[(\alpha, \langle \tilde{\Delta}; \tilde{\Gamma} \frown (\beta, \mathtt{S}) \rangle)]_i)^{-j} \rangle]_{\mathbf{k}}$ for $\#_i\Gamma = (\alpha, \langle \tilde{\Delta}; \tilde{\Gamma} \rangle)$ and $\#_j\Gamma = (\beta, \mathtt{S})$	J-rule

Table 1
Rewriting rules of $\mathcal{R}$ for $\mathtt{T} \in \mathsf{Tree}^{\diamond}$, $\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$ and $\mathtt{T}|_{\mathbf{k}} = \langle \Delta; \Gamma \rangle$.

To define the rewriting rules, we introduce the following notation. Let $\mathtt{T} = \langle \Delta; \Gamma \rangle$ be a modal tree, $0 < i, j \leq |\Gamma|$ and $n \leq |\Delta|$ such that $\Gamma = [(\alpha_1, \mathtt{S}_1), ..., (\alpha_m, \mathtt{S}_m)]$. The ith element of Γ is denoted by $\#_i\Gamma$. The list obtained by erasing the ith element of Γ, i.e. $[(\alpha_1, \mathtt{S}_1), ..., (\alpha_{i-1}, \mathtt{S}_{i-1}), (\alpha_{i+1}, \mathtt{S}_{i+1}), ..., (\alpha_m, \mathtt{S}_m)]$, is denoted by Γ^{-i}. The list obtained by duplicating the ith element of Γ and placing it at the beginning, i.e. $(\alpha_i, \mathtt{S}_i) \frown \Gamma$, is denoted by Γ^{+i}. Analogously, the list obtained by erasing the nth element of Δ and the list obtained by duplicating the nth element of Δ and placing it at the beginning are denoted by Δ^{-n} and Δ^{+n}, respectively. The *list obtained from* Γ *by replacing its ith element* by $(\alpha, \mathtt{S})$ is defined by $\Gamma[(\alpha, \mathtt{S})]_i := [(\alpha_1, \mathtt{S}_1), ..., (\alpha_{i-1}, \mathtt{S}_{i-1}), (\alpha, \mathtt{S}), (\alpha_{i+1}, \mathtt{S}_{i+1}), ..., (\alpha_m, \mathtt{S}_m)]$. Note that we use the same notation for replacement in a list of pairs and replacement in a modal tree since the context allows for a clear distinction. Finally, the list obtained by interchanging the ith element with the jth element, i.e. $(\Gamma[\#_i\Gamma]_j)[\#_j\Gamma]_i$, is denoted by $\Gamma^{i \leftrightarrow j}$.

We can now present the tree rewriting system for Reflection Calculus.

Definition 5.1 (TRC) The *tree rewriting system for* **RC**, denoted by TRC, is the abstract rewriting system $(\mathsf{Tree}^{\diamond}, \{\hookrightarrow^{\mu}\}_{\mu \in \mathcal{R}})$ for $\mathcal{R} = \{\rho^+, \rho^-, \sigma, \pi^+, \pi^-, 4, \lambda, \mathsf{J}\}$. The rewriting rules of $\mathcal{R}$ are defined in Table 1.

Due to the transformations they induce, the rules are named as follows:

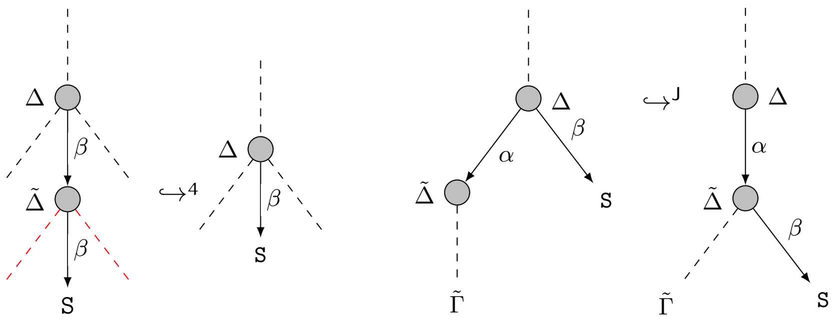


Fig. 2. 4-rule and J-rule.

the ρ^+-rule is called *atom duplication*, the ρ^--rule is called *atom elimination*, the σ-rule is called *child permutation*, the π^+-rule is called as *child duplication*, the π^--rule is called *child elimination*, the 4-rule is called *transitivity*, and the λ-rule is called *monotonicity*.

More generally, the tree rewriting relation is the union of the rewriting rules.

Definition 5.2 ($\hookrightarrow$) The *tree rewriting relation* $\hookrightarrow$ is defined as

$$\hookrightarrow := \hookrightarrow^{\rho^+} \cup \hookrightarrow^{\rho^-} \cup \hookrightarrow^{\sigma} \cup \hookrightarrow^{\pi^+} \cup \hookrightarrow^{\pi^-} \cup \hookrightarrow^{4} \cup \hookrightarrow^{\lambda} \cup \hookrightarrow^{\mathsf{J}} .$$

We say that *the step in* $\mathtt{T} \hookrightarrow \mathtt{T}'$ *has been performed at position* $\mathbf{k}$ if the applied rule replaces the subtree at position $\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$. We say that $\mathtt{T}$ *rewrites to* $\mathtt{T}'$, denoted by $\mathtt{T} \hookrightarrow^* \mathtt{T}'$, if $\mathtt{T}'$ is the result of applying zero, one or several rewriting rules of $\mathcal{R}$ to $\mathtt{T}$. In other words, $\hookrightarrow^*$ denotes the reflexive transitive closure of $\hookrightarrow$. The trees $\mathtt{T}$ *and* $\mathtt{T}'$ *are* TRC*-equivalent*, denoted by $\mathtt{T} \overset{*}{\hookleftarrow\!\!\!\hookrightarrow} \mathtt{T}'$, if $\mathtt{T} \hookrightarrow^* \mathtt{T}'$ and $\mathtt{T}' \hookrightarrow^* \mathtt{T}$. If $\mathtt{T}$ rewrites to $\mathtt{T}'$ by applying the rewriting rule μ zero, one or several times, we write $\mathtt{T} \hookrightarrow^{\mu *} \mathtt{T}'$. For Ω a list of rewriting rules, we define $\mathtt{T} \hookrightarrow^{\Omega} \mathtt{S}$ inductively as $\mathtt{T} \hookrightarrow^* \mathtt{T}$ by applying no rewriting rules if $\Omega = \varnothing$, and $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\hat{\Omega}} \mathtt{S}$ for $\Omega = \mu \frown \hat{\Omega}$. Likewise, for Ω_1 and Ω_2 lists of rewriting rules, $\mathtt{T} \hookrightarrow^{\Omega_1} \circ \hookrightarrow^{\Omega_2} \mathtt{S}$ denotes that there is $\hat{\mathtt{S}}$ such that $\mathtt{T} \hookrightarrow^{\Omega_1} \hat{\mathtt{S}} \hookrightarrow^{\Omega_2} \mathtt{S}$.

Modal trees with permuted lists labeling the nodes are TRC-equivalent.

Lemma 5.3 $\langle \Delta_1 \frown \Delta_2; \Gamma \rangle \overset{*}{\hookleftarrow\!\!\!\hookrightarrow} \langle \Delta_2 \frown \Delta_1; \Gamma \rangle$.

Proof. It suffices to show $\langle \Delta_1 \frown \Delta_2; \Gamma \rangle \hookrightarrow^* \langle \Delta_2 \frown \Delta_1; \Gamma \rangle$ by induction on the length of Δ_2. If Δ_2 is empty, the result trivially holds. Assuming $\langle \Delta \frown \Delta_2; \Gamma \rangle \hookrightarrow^* \langle \Delta_2 \frown \Delta; \Gamma \rangle$ for any list of propositional variables Δ, using atom duplication and atom elimination rewriting rules we conclude

$$\begin{aligned} \langle \Delta_1 \frown (p \frown \Delta_2); \Gamma \rangle = \langle (\Delta_1 \frown p) \frown \Delta_2; \Gamma \rangle &\hookrightarrow^* \langle \Delta_2 \frown (\Delta_1 \frown p); \Gamma \rangle \\ &\hookrightarrow^{\rho^+} \langle p \frown \Delta_2 \frown \Delta_1 \frown p; \Gamma \rangle \hookrightarrow^{\rho^-} \langle (p \frown \Delta_2) \frown \Delta_1; \Gamma \rangle. \end{aligned}$$

$\square$

Here are some useful results on rewriting a sum of modal trees.

Lemma 5.4 *Let* $\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3, \mathtt{S}_1, \mathtt{S}_2 \in \mathsf{Tree}^\diamond$. *Then,*

(i) $\mathtt{T}_1 \overset{*}{\leftrightarrow} \mathtt{T}_1 + \mathtt{T}_1$;

(ii) $\mathtt{T}_1 + \mathtt{T}_2 \overset{*}{\leftrightarrow} \mathtt{T}_2 + \mathtt{T}_1$;

(iii) $\mathtt{T}_1 + \mathtt{T}_2 \hookrightarrow^* \mathtt{T}_1$ *and* $\mathtt{T}_1 + \mathtt{T}_2 \hookrightarrow^* \mathtt{T}_2$;

(iv) *If* $\mathtt{T}_1 \hookrightarrow^* \mathtt{S}_1$, *then* $\mathtt{T}_1 + \mathtt{T}_2 \hookrightarrow^* \mathtt{S}_1 + \mathtt{T}_2$;

(v) *If* $\mathtt{T}_1 \hookrightarrow^* \mathtt{T}_2$ *and* $\mathtt{T}_1 \hookrightarrow^* \mathtt{T}_3$ *then* $\mathtt{T}_1 \hookrightarrow^* \mathtt{T}_2 + \mathtt{T}_3$;

(vi) *If* $\mathtt{S}_1 \hookrightarrow^* \mathtt{T}_1$ *and* $\mathtt{S}_2 \hookrightarrow^* \mathtt{T}_2$, *then* $\mathtt{S}_1 + \mathtt{S}_2 \hookrightarrow^* \mathtt{T}_1 + \mathtt{T}_2$.

Proof. The implication from left to right of the first statement holds by atom and child duplication, and the inverse implication by atom and child elimination. The second result holds by Lemma 5.3 and child permutation. The third follows by atom and child elimination. The fourth result holds by induction on the number of rewriting steps performed in $\mathtt{T}_1 \hookrightarrow^* \mathtt{S}_1$ and by cases on the rewriting rules. The fifth statement holds by the fourth result using the statements one and two:

$$\mathtt{T}_1 \hookrightarrow^* \mathtt{T}_1 + \mathtt{T}_1 \hookrightarrow^* \mathtt{T}_2 + \mathtt{T}_1 \hookrightarrow^* \mathtt{T}_1 + \mathtt{T}_2 \hookrightarrow^* \mathtt{T}_3 + \mathtt{T}_2 \hookrightarrow^* \mathtt{T}_2 + \mathtt{T}_3.$$

Finally, by the fifth statement it suffices to show $\mathtt{S}_1 + \mathtt{S}_2 \hookrightarrow^* \mathtt{T}_1$ and $\mathtt{S}_1 + \mathtt{S}_2 \hookrightarrow^* \mathtt{T}_2$ to prove the sixth result, which follow by the third statement and the hypotheses. □

The following results state that transformations in subtrees can be extended to the entire tree, allowing for systematic and consistent modifications throughout the tree. In consequence, we can effectively manipulate and modify complex tree structures while leaving the other parts untouched.

Proposition 5.5 (Inside Rewriting Property) *If* $\mathtt{S} \hookrightarrow^* \mathtt{S}'$, *then* $\mathtt{T}[\mathtt{S}]_{\mathbf{k}} \hookrightarrow^* \mathtt{T}[\mathtt{S}']_{\mathbf{k}}$.

Proof. By induction on the number of rewriting steps that are performed in $\mathtt{S} \hookrightarrow^* \mathtt{S}'$. If no rewriting step is performed, the result is trivially satisfied. Now assume $\mathtt{T}[\mathtt{S}]_{\mathbf{k}} \hookrightarrow^* \mathtt{T}[\hat{\mathtt{S}}]_{\mathbf{k}}$ for $\mathtt{S} \hookrightarrow^* \hat{\mathtt{S}}$ by performing n rewriting steps. Moreover, let $\mathtt{S} \hookrightarrow^* \hat{\mathtt{S}} \hookrightarrow^\mu \hat{\mathtt{S}}[\tilde{\mathtt{S}}]_{\mathbf{r}}$ for $\tilde{\mathtt{S}} \in \mathsf{Tree}^\diamond$ and $\mathbf{r} \in \mathsf{Pos}(\hat{\mathtt{S}})$ according to the rewriting rule μ. Since $\mathtt{T}[\mathtt{S}]_{\mathbf{k}} \hookrightarrow^* \mathtt{T}[\hat{\mathtt{S}}]_{\mathbf{k}}$ by the inductive hypothesis, it suffices to show $\mathtt{T}[\hat{\mathtt{S}}]_{\mathbf{k}} \hookrightarrow^\mu \mathtt{T}[\hat{\mathtt{S}}[\tilde{\mathtt{S}}]_{\mathbf{r}}]_{\mathbf{k}} = (\mathtt{T}[\hat{\mathtt{S}}]_{\mathbf{k}})[\tilde{\mathtt{S}}]_{\mathbf{kr}}$ by Lemma 3.6. We proceed by induction on the tree structure of $\mathtt{T}$. For $\mathtt{T}$ a leaf, by the hypothesis,

$$\mathtt{T}[\hat{\mathtt{S}}]_\epsilon = \hat{\mathtt{S}} \hookrightarrow^\mu \hat{\mathtt{S}}[\tilde{\mathtt{S}}]_{\mathbf{r}} = (\mathtt{T}[\hat{\mathtt{S}}]_\epsilon)[\tilde{\mathtt{S}}]_{\epsilon\mathbf{r}}.$$

Let $\mathtt{T} = \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_m, \mathtt{S}_m)]\rangle$ such that $\mathtt{S}_i[\hat{\mathtt{S}}]_{\mathbf{l}} \hookrightarrow^\mu (\mathtt{S}_i[\hat{\mathtt{S}}]_{\mathbf{l}})[\tilde{\mathtt{S}}]_{\mathbf{lr}}$ for $\mathbf{l} \in \mathsf{Pos}(\mathtt{S}_i)$ and $1 \le i \le m$. We continue by cases on the length of $\mathbf{k}$. For $\mathbf{k} = \epsilon$, the result is trivially satisfied. For $\mathbf{k} = i\hat{\mathbf{k}}$ such that $\hat{\mathbf{k}} \in \mathsf{Pos}(\mathtt{S}_i)$ and $1 \le i \le n$, we conclude by performing μ at position $\mathbf{kr}$ using the inductive hypothesis for $\mathtt{S}_i$ as follows,

$$\begin{aligned}\mathtt{T}[\hat{\mathtt{S}}]_{i\hat{\mathbf{k}}} &= \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}_i[\hat{\mathtt{S}}]_{\hat{\mathbf{k}}}), ..., (\alpha_m, \mathtt{S}_m)]\rangle \\ &\hookrightarrow^\mu \langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, (\mathtt{S}_i[\hat{\mathtt{S}}]_{\hat{\mathbf{k}}})[\tilde{\mathtt{S}}]_{\hat{\mathbf{k}}\mathbf{r}}), ..., (\alpha_m, \mathtt{S}_m)]\rangle = (\mathtt{T}[\hat{\mathtt{S}}]_{i\hat{\mathbf{k}}})[\tilde{\mathtt{S}}]_{i\hat{\mathbf{k}}\mathbf{r}}.\end{aligned}$$

□

Corollary 5.6 *If* $\mathtt{T}|_{\mathbf{k}} \hookrightarrow^* \mathtt{S}$*, then* $\mathtt{T} \hookrightarrow^* \mathtt{T}[\mathtt{S}]_{\mathbf{k}}$ *for* $\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$*.*

Proof. By Proposition 5.5 since $\mathtt{T} = \mathtt{T}[\mathtt{T}|_{\mathbf{k}}]_{\mathbf{k}}$ (Lemma 3.6). □

6 Adequacy and completeness

We aim to show how the tree rewriting system TRC faithfully simulates logical derivations in **RC** thanks to the embeddings defined in Section 4. Thereby, adequacy and completeness theorems are presented as key results that underscore the efficacy of tree rewriting systems in relating logical inference and structural transformation.

Firstly, we show adequacy by proving that if a rewriting step is performed, the sequent of formulas which the trees are mapped to is provable in **RC**.

Proposition 6.1 *If* $\mathtt{T} \hookrightarrow \mathtt{T}'$*, then* $\mathcal{F}(\mathtt{T}) \vdash_{\mathbf{RC}} \mathcal{F}(\mathtt{T}')$ *for* $\mathtt{T}, \mathtt{T}' \in \mathsf{Tree}^{\diamond}$*.*

Proof. By induction on the tree structure of $\mathtt{T}$. For $\mathtt{T}$ a leaf the result follows easily by cases on the performed rewriting rule. Now consider $\mathtt{T} = \langle \Delta; \Gamma \rangle$ for $\Gamma = [(\alpha_1, \mathtt{S}_1), ..., (\alpha_m, \mathtt{S}_m)]$ such that $\mathcal{F}(\mathtt{S}_i) \vdash_{\mathbf{RC}} \mathcal{F}(\mathtt{S}')$ if $\mathtt{S}_i \hookrightarrow \mathtt{S}'$ for $1 \leq i \leq m$. Assuming $\langle \Delta; \Gamma \rangle \hookrightarrow^{\mu} \mathtt{T}'$ for μ a rewriting rule, we show $\mathcal{F}(\mathtt{T}) \vdash_{\mathbf{RC}} \mathcal{F}(\mathtt{T}')$ by cases on the length of the position at which μ is performed.

First consider a rewriting at a position $i\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$. By definition, $\mathtt{T}'$ is of the form $\langle \Delta; [(\alpha_1, \mathtt{S}_1), ..., (\alpha_i, \mathtt{S}'), ..., (\alpha_m, \mathtt{S}_m)] \rangle$ for $\mathtt{S}_i \hookrightarrow^{\mu} \mathtt{S}'$ by rewriting at position $\mathbf{k}$. Hence, by the inductive hypothesis, $\mathcal{F}(\mathtt{S}_i) \vdash_{\mathbf{RC}} \mathcal{F}(\mathtt{S}')$. Thus,

$$\begin{aligned}
\mathcal{F}(\mathtt{T}) &\equiv_{\mathbf{K}^+} \bigwedge \Delta \wedge \langle \alpha_i \rangle \mathcal{F}(\mathtt{S}_i) \wedge \bigwedge \diamond(\Gamma^{-i}) \\
&\vdash_{\mathbf{RC}} \bigwedge \Delta \wedge \langle \alpha_i \rangle \mathcal{F}(\mathtt{S}') \wedge \bigwedge \diamond(\Gamma^{-i}) \equiv_{\mathbf{K}^+} \mathcal{F}(\mathtt{T}').
\end{aligned}$$

For rewriting performed at position ϵ, the proof concludes easily by cases on μ. Let see in some detail the cases of transitivity and J rewriting rules.

4-rule: Consider $\mathtt{T} \hookrightarrow^{4} \langle \Delta; \Gamma[(\beta, \mathtt{S})]_i \rangle$ such that $\#_i \Gamma = (\beta, \langle \tilde{\Delta}; \tilde{\Gamma} \rangle)$ for $0 < i \leq |\Gamma|$ and $\#_j \tilde{\Gamma} = (\beta, \mathtt{S})$ for $0 < j \leq |\tilde{\Gamma}|$. By Lemma 2.2 and transitivity rule for **RC** we conclude,

$$\begin{aligned}
\mathcal{F}(\mathtt{T}) &\equiv_{\mathbf{K}^+} \bigwedge \Delta \wedge \langle \beta \rangle \mathcal{F}(\langle \tilde{\Delta}; \tilde{\Gamma} \rangle) \wedge \bigwedge \diamond(\Gamma^{-i}) \\
&\equiv_{\mathbf{K}^+} \bigwedge \Delta \wedge \langle \beta \rangle (\bigwedge \tilde{\Delta} \wedge \langle \beta \rangle \mathcal{F}(\mathtt{S}) \wedge \bigwedge \diamond(\tilde{\Gamma}^{-j})) \wedge \bigwedge \diamond(\Gamma^{-i}) \\
&\vdash_{\mathbf{K}^+} \bigwedge \Delta \wedge \langle \beta \rangle \langle \beta \rangle \mathcal{F}(\mathtt{S}) \wedge \bigwedge \diamond(\Gamma^{-i}) \\
&\vdash_{\mathbf{RC}} \bigwedge \Delta \wedge \langle \beta \rangle \mathcal{F}(\mathtt{S}) \wedge \bigwedge \diamond(\Gamma^{-i}) \\
&\equiv_{\mathbf{K}^+} \mathcal{F}(\langle \Delta; \Gamma[(\beta, \mathtt{S})]_i \rangle).
\end{aligned}$$

J-rule: Consider $\mathtt{T} \hookrightarrow^{\mathsf{J}} \langle \Delta; (\Gamma[(\alpha, \langle \tilde{\Delta}; \tilde{\Gamma} \frown (\beta, \mathtt{S}) \rangle)]_i)^{-j} \rangle$ such that $\#_i \Gamma = (\alpha, \langle \tilde{\Delta}; \tilde{\Gamma} \rangle)$ and $\#_j \Gamma = (\beta, \mathtt{S})$ for $0 < i, j, \leq |\Gamma|$ satisfying $i \neq j$. Let $i < j$

without loss of generality. By the J rule for **RC** we conclude,

$$\begin{aligned}
\mathcal{F}(\mathtt{T}) &\equiv_{\mathbf{K}^+} \bigwedge \Delta \wedge \langle\alpha\rangle\mathcal{F}(\langle\tilde{\Delta};\tilde{\Gamma}\rangle) \wedge \langle\beta\rangle\mathcal{F}(\mathtt{S}) \wedge \bigwedge \Diamond((\Gamma^{-j})^{-i}) \\
&\vdash_{\mathbf{RC}} \bigwedge \Delta \wedge \langle\alpha\rangle(\mathcal{F}(\langle\tilde{\Delta};\tilde{\Gamma}\rangle) \wedge \langle\beta\rangle\mathcal{F}(\mathtt{S})) \wedge \bigwedge \Diamond((\Gamma^{-j})^{-i}) \\
&\equiv_{\mathbf{K}^+} \bigwedge \Delta \wedge \langle\alpha\rangle(\mathcal{F}(\langle\tilde{\Delta};\tilde{\Gamma} \frown (\beta,\mathtt{S})\rangle)) \wedge \bigwedge \Diamond((\Gamma^{-j})^{-i}) \\
&\equiv_{\mathbf{K}^+} \mathcal{F}(\langle\Delta;(\Gamma[(\alpha,\langle\tilde{\Delta};\tilde{\Gamma} \frown (\beta,\mathtt{S})\rangle)]_i)^{-j}\rangle).
\end{aligned}$$

□

Theorem 6.2 (Adequacy) *If* $\mathtt{T} \hookrightarrow^* \mathtt{T}'$*, then* $\mathcal{F}(\mathtt{T}) \vdash_{\mathbf{RC}} \mathcal{F}(\mathtt{T}')$ *for* $\mathtt{T},\mathtt{T}' \in \mathsf{Tree}^{\Diamond}$.

Proof. By an easy induction on the number of rewriting steps that are performed using Proposition 6.1. □

We conclude this section by showing that TRC is complete with respect to **RC**.

Theorem 6.3 (Completeness) *If* $\varphi \vdash_{\mathbf{RC}} \psi$*, then* $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\psi)$ *for* $\varphi,\psi \in \mathcal{L}^+$.

Proof. By induction on the length of the **RC** derivation.

If $\varphi \vdash_{\mathbf{RC}} \varphi$, then $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\varphi)$ by applying no rewriting rule; if $\varphi \vdash_{\mathbf{RC}} \top$, then $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\top)$ by atom and child elimination.

(Cut): If $\varphi \vdash_{\mathbf{RC}} \psi$ and $\psi \vdash_{\mathbf{RC}} \phi$ such that $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\psi)$ and $\mathcal{T}(\psi) \hookrightarrow^* \mathcal{T}(\phi)$, then $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\phi) \hookrightarrow^* \mathcal{T}(\phi)$.

(Elimination of conjunction): If $\varphi\wedge\psi \vdash_{\mathbf{RC}} \varphi$, by Lemma 5.4 follows $\mathcal{T}(\varphi\wedge\psi) = \mathcal{T}(\varphi) + \mathcal{T}(\psi) \hookrightarrow^* \mathcal{T}(\varphi)$. Analogously, if $\varphi \wedge \psi \vdash_{\mathbf{RC}} \psi$ then $\mathcal{T}(\varphi \wedge \psi) \hookrightarrow^* \mathcal{T}(\psi)$ by Lemma 5.4.

(Introduction to conjunction): If $\varphi \vdash_{\mathbf{RC}} \psi$ and $\varphi \vdash_{\mathbf{RC}} \phi$ such that $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\psi)$ and $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\phi)$, by Lemma 5.4 we conclude $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\psi)+\mathcal{T}(\phi) = \mathcal{T}(\psi \wedge \phi)$.

(Distribution): If $\varphi \vdash_{\mathbf{RC}} \psi$ such that $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\psi)$, then $\mathcal{T}(\langle\alpha\rangle\varphi) \hookrightarrow^* \mathcal{T}(\langle\alpha\rangle\psi)$ by the inside rewriting property (Corollary 5.6).

(Transitivity): If $\langle\alpha\rangle\langle\alpha\rangle\varphi \vdash_{\mathbf{RC}} \langle\alpha\rangle\varphi$, by the $\mathsf{4}$-rule,

$$\mathcal{T}(\langle\alpha\rangle\langle\alpha\rangle\varphi) = \langle\varnothing;[(\alpha,\langle\varnothing;[(\alpha,\mathcal{T}(\varphi))]\rangle)]\rangle \hookrightarrow^{\mathsf{4}} \langle\varnothing;[(\alpha,\mathcal{T}(\varphi))]\rangle = \mathcal{T}(\langle\alpha\rangle\varphi).$$

(Monotonicity): If $\langle\alpha\rangle\varphi \vdash_{\mathbf{RC}} \langle\beta\rangle\varphi$ for $\alpha > \beta$, by the λ-rule,

$$\mathcal{T}(\langle\alpha\rangle\varphi) = \langle\varnothing;[(\alpha,\mathcal{T}(\varphi))]\rangle \hookrightarrow^{\lambda} \langle\varnothing;[(\beta,\mathcal{T}(\varphi))]\rangle = \mathcal{T}(\langle\beta\rangle\varphi).$$

(J): Consider $\alpha > \beta$ and $\mathcal{T}(\varphi) = \langle\Delta;\Gamma\rangle$. If $\langle\alpha\rangle\varphi \wedge \langle\beta\rangle\psi \vdash_{\mathbf{RC}} \langle\alpha\rangle(\varphi \wedge \langle\beta\rangle\psi)$, by the J-rule,

$$\begin{aligned}
\mathcal{T}(\langle\alpha\rangle\varphi \wedge \langle\beta\rangle\psi) &= \langle\varnothing;[(\alpha,\langle\Delta;\Gamma\rangle),(\beta,\mathcal{T}(\psi))]\rangle \hookrightarrow^{\mathsf{J}} \langle\varnothing;[(\alpha,\langle\Delta;\Gamma \frown (\beta,\mathcal{T}(\psi))\rangle)]\rangle \\
&= \langle\varnothing;[(\alpha,\mathcal{T}(\varphi)+\mathcal{T}(\langle\beta\rangle\psi))]\rangle = \mathcal{T}(\langle\alpha\rangle(\varphi \wedge \langle\beta\rangle\psi)).
\end{aligned}$$

□

This concluding corollary states that a sequence can be shown in **RC** by identifying transformations within the trees in which the involved formulas are embedded into. Likewise, a rewriting can be proven by showing the corresponding **RC** sequence for the formulas embedding the corresponding trees.

Corollary 6.4 $\mathcal{T}(\varphi) \hookrightarrow^* \mathcal{T}(\psi) \implies \varphi \vdash_{\mathbf{RC}} \psi$ *and* $\mathcal{F}(\mathtt{T}) \vdash_{\mathbf{RC}} \mathcal{F}(\mathtt{T}') \implies \mathtt{T} \hookrightarrow^* \mathtt{T}'$.

Proof. By adequacy and completeness (Theorems 6.2 and 6.3, respectively) using the embedding composition property (Proposition 4.4). □

7 The Rewrite Normalization Theorem

In this section we present our rewriting normalization theorem, which allows us to perform rewriting in a designated sequence of rewriting rules according to their kinds. Specifically, we can always perform the rewrite process using normal rewriting sequences.

Definition 7.1 (Normal rewrite) A list of rewriting rules Ω is a *normal rewriting sequence* if it is of the form

$$\Omega_{\pi^+} \frown \Omega_{\diamond} \frown \Omega_{\delta} \frown \Omega_{\rho} \frown \Omega_{\sigma}$$

for $\Omega_{\pi^+}, \Omega_{\diamond}, \Omega_{\delta}, \Omega_{\rho}$ and Ω_{σ} lists of replicative, modal, decreasing, atomic and structural rewriting rules, respectively.

The order of the presented normal rewriting sequence adheres to the following principles. Firstly, performing any kind of rewriting rule before a replicative one cannot be equivalently reversed. Similarly, the order of modal and decreasing rewriting rules cannot be interchanged (for example, it may be necessary to decrease the label of an edge in order to apply 4-rule). Finally, atomic and structural rewriting rules, which pertain to node labels and child permutation, are placed last because nodes and children may be removed during the rewriting process.

In the following, we disclose the results needed to establish the normal rewriting theorem. Their proofs delve into technical intricacies, requiring meticulous attention to the rewritten positions and the arguments determining the application of the concerned rewriting rules. If two rewriting rules are applied in different positions of the tree such that their outcome transformation is disjoint, their permutation is proven straightforward. However, if their application intersects, additional transformations may be required. To illustrate these scenarios, we provide simplified examples alongside the proofs.

Lemma 7.2 *If* $\mathtt{T} \hookrightarrow^{\sigma} \circ \hookrightarrow^{\mu} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\sigma *} \mathtt{S}$ *for* $\mu \in \mathcal{R} \setminus \{\sigma\}$.

Proof. By cases on μ. Each case is proven by induction on the length of the position of σ and by cases on the position of μ. □

Corollary 7.3 *If* $\mathtt{T} \hookrightarrow^{\sigma *} \circ \hookrightarrow^{\Omega} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\Omega} \circ \hookrightarrow^{\sigma *} \mathtt{S}$ *for* Ω *a list of atomic, replicative, decreasing and modal rewriting rules.*

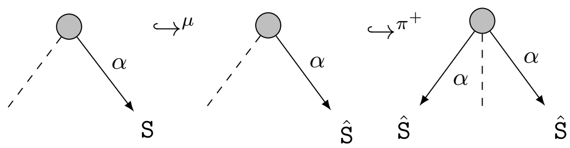
Proof. By induction on the length of Ω and by induction on the number of σ-steps performed using Lemma 7.2. $\square$

Lemma 7.4 *Let ρ be an atomic rule. If* $\mathtt{T} \hookrightarrow^{\rho} \circ \hookrightarrow^{\mu} \mathtt{S}$*, then* $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\rho *} \mathtt{S}$ *for* $\mu \in \{\pi^+, \pi^-, 4, \lambda, \mathsf{J}\}$.

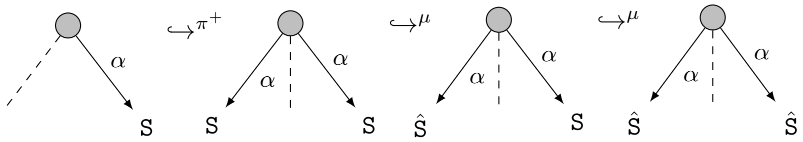
Proof. By cases on μ. Each case is shown by induction on the length of the position of ρ and by cases on the position of μ. If μ is π^+, we prove that either $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\rho} \mathtt{S}$ or $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\rho} \circ \hookrightarrow^{\rho} \mathtt{S}$ holds (see Figure 3). If μ is a decreasing rewriting rule, then either $\mathtt{T} \hookrightarrow^{\mu} \mathtt{S}$ or $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\rho} \mathtt{S}$. Otherwise we show $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\rho} \mathtt{S}$. $\square$

Corollary 7.5 *Let Ω_ρ be a list of atomic rewriting rules. If* $\mathtt{T} \hookrightarrow^{\Omega_\rho} \circ \hookrightarrow^{\mu} \mathtt{S}$*, then* $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\Omega'_\rho} \mathtt{S}$ *for* $\mu \in \{\pi^+, \pi^-, 4, \lambda, \mathsf{J}\}$ *and Ω'_ρ a list of atomic rewriting rules.*

Proof. By an easy induction on the length of Ω_ρ using Lemma 7.4. $\square$



(a) $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\pi^+} \mathtt{S}$ (for $\mathtt{S} \hookrightarrow^{\mu} \hat{\mathtt{S}}$).



(b) $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mu} \circ \hookrightarrow^{\mu} \mathtt{S}$.

Fig. 3. $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\pi^+} \mathtt{S} \Longrightarrow \mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mu} \circ \hookrightarrow^{\mu} \mathtt{S}, \mu \in \mathcal{R} \setminus \{\pi^+\}$.

Proposition 7.6 *Let δ be a decreasing rule. If* $\mathtt{T} \hookrightarrow^{\delta} \circ \hookrightarrow^{\mu} \mathtt{S}$*, then* $\mathtt{T} \hookrightarrow^{\mu *} \circ \hookrightarrow^{\delta *} \mathtt{S}$ *for* $\mu \in \{\pi^+, \lambda, \mathsf{J}\}$.

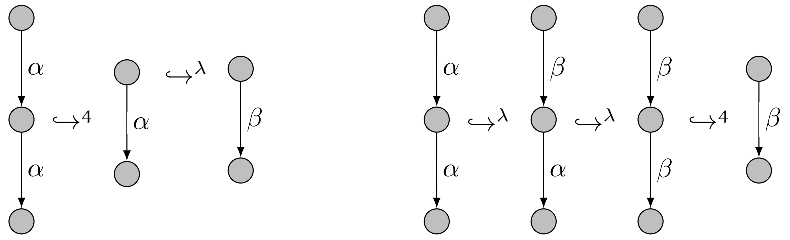
Proof. By induction on the length of the position of δ and by cases on the length of the position of μ.

Let δ be π^-. If μ is π^+, we show that either $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\pi^-} \mathtt{S}$ or $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\pi^-} \circ \hookrightarrow^{\pi^-} \mathtt{S}$ holds (see Figure 3). Otherwise we prove $\mathtt{T} \hookrightarrow^{\mu} \circ \hookrightarrow^{\pi^-} \mathtt{S}$.

Let δ be 4. If μ is π^+ we show that either $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{4} \mathtt{S}$ or $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{4} \circ \hookrightarrow^{4} \mathtt{S}$ (see Figure 3). For μ being λ we prove that either $\mathtt{T} \hookrightarrow^{\lambda} \circ \hookrightarrow^{4} \mathtt{S}$ or $\mathtt{T} \hookrightarrow^{\lambda} \circ \hookrightarrow^{\lambda} \circ \hookrightarrow^{4} \mathtt{S}$ (see Figure 4). Finally, for μ being J we show that either $\mathtt{T} \hookrightarrow^{J} \circ \hookrightarrow^{4} \mathtt{S}$ or $\mathtt{T} \hookrightarrow^{J} \circ \hookrightarrow^{J} \circ \hookrightarrow^{4} \mathtt{S}$ (see Figure 5). $\square$

Corollary 7.7 *Let δ and θ be decreasing and modal rewriting rules respectively, and Ω_δ be a list of decreasing rewriting rules.*

(i) *If* $\mathtt{T} \hookrightarrow^{\Omega_\delta} \circ \hookrightarrow^{\pi^+} \mathtt{S}$*, then* $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\Omega'_\delta} \mathtt{S}$ *for Ω'_δ a list of decreasing rules.*



(a) $\mathtt{T} \hookrightarrow^{4} \circ \hookrightarrow^{\lambda} \mathtt{S}$. (b) $\mathtt{T} \hookrightarrow^{\lambda} \circ \hookrightarrow^{\lambda} \circ \hookrightarrow^{4} \mathtt{S}$.

Fig. 4. $\mathtt{T} \hookrightarrow^{4} \circ \hookrightarrow^{\lambda} \mathtt{S} \Longrightarrow \mathtt{T} \hookrightarrow^{\lambda} \circ \hookrightarrow^{\lambda} \circ \hookrightarrow^{4} \mathtt{S}$.

(ii) *If* $\mathtt{T} \hookrightarrow^{\delta} \circ \hookrightarrow^{\theta *} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\theta *} \circ \hookrightarrow^{\delta} \mathtt{S}$.

(iii) *If* $\mathtt{T} \hookrightarrow^{\Omega_\delta} \circ \hookrightarrow^{\mu *} \mathtt{S}$ *for* $\mu \in \{\pi^+, \lambda, \mathsf{J}\}$, *then* $\mathtt{T} \hookrightarrow^{\mu *} \circ \hookrightarrow^{\Omega'_\delta} \mathtt{S}$ *for* Ω'_δ *a list of decreasing rewriting rules.*

Proof. The first statement follows by induction on the length of Ω_δ and Proposition 7.6. The second one is shown by induction on the number of modal rewriting rules applied and Proposition 7.6. The third result follows by cases on μ. For $\mu \in \{\lambda, \mathsf{J}\}$, we conclude by induction on the length of Ω_δ and the second statement. Otherwise, we conclude by induction on the number of μ-rules applied and the first statement. □

Proposition 7.8 *Let* θ *be a modal rewriting rule. If* $\mathtt{T} \hookrightarrow^{\theta} \circ \hookrightarrow^{\pi^+} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\pi^+ *} \circ \hookrightarrow^{\theta *} \circ \hookrightarrow^{\sigma *} \mathtt{S}$.

Proof. By induction on the length of the position of θ and by cases on the length of the position of π^+. If θ is λ, we show that either $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\lambda} \mathtt{S}$ or $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\lambda} \circ \hookrightarrow^{\lambda} \mathtt{S}$ holds (see Figure 3). For θ being J, we prove that either $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mathsf{J}} \mathtt{S}$ or $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \mathtt{S}$ (see Figure 3) or $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \mathtt{S}$ (see Figure 6) or $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\sigma} \mathtt{S}$ (see Figure 7). □

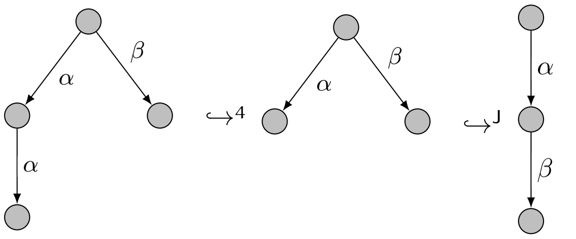
We now want to show permutation of the application of multiple π^+-rules after multiple modal rewriting rules. To show permutability of π^+-rules following a J-rule, we need to reorganize duplications applied at positions in increasing depth in the tree. Furthermore, we require flexibility to choose which branch to duplicate first.

For readability, we write $\mathtt{T} \hookrightarrow^{\pi^+(\mathbf{k},i)} \mathtt{S}$ to denote that $\mathtt{S}$ is obtained by applying the π^+-rule to duplicate the ith child at position $\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$. Similarly, $\mathtt{T} \hookrightarrow^{\sigma(\mathbf{k},i,j)} \mathtt{S}$ denotes the application of the σ-rule to permute the ith and jth children at position $\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$. Given this notation, the following result aids in reorganizing duplications at positions in increasing depth.

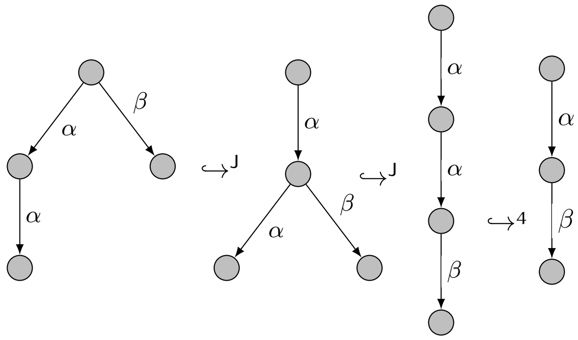
Lemma 7.9 (Depth-level permutability of π^+)

(i) *If* $\mathtt{T} \hookrightarrow^{\pi^+(l\mathbf{k},i)} \circ \hookrightarrow^{\pi^+(\epsilon,j)} \mathtt{S}$ *for* $j \neq l$, *then* $\mathtt{T} \hookrightarrow^{\pi^+(\epsilon,j)} \circ \hookrightarrow^{\pi^+((l+1)\mathbf{k},i)} \mathtt{S}$.

(ii) *If* $\mathtt{T} \hookrightarrow^{\pi^+(j\mathbf{k},i)} \circ \hookrightarrow^{\pi^+(\epsilon,j)} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\pi^+(\epsilon,j)} \circ \hookrightarrow^{\pi^+((j+1)\mathbf{k},i)} \circ \hookrightarrow^{\pi^+(1\mathbf{k},i)} \mathtt{S}$.



(a) $\mathtt{T} \hookrightarrow^{4} \circ \hookrightarrow^{\mathsf{J}} \mathtt{S}$.



(b) $\mathtt{T} \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{4} \mathtt{S}$.

Fig. 5. $\mathtt{T} \hookrightarrow^{4} \circ \hookrightarrow^{\mathsf{J}} \mathtt{S} \Longrightarrow \mathtt{T} \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{4} \mathtt{S}$.

(iii) *If* $\mathtt{T} \hookrightarrow^{\pi^+(l\mathbf{k}_l,i)} \circ \hookrightarrow^{\pi^+(n\mathbf{k}_n,j)} \mathtt{S}$ *for* $|n\mathbf{k}_n| < |l\mathbf{k}_l|$, *then* $\mathtt{T} \hookrightarrow^{\pi^+(n\mathbf{k}_n,j)} \circ \hookrightarrow^{\pi^+(l\mathbf{k}_l,i)} \mathtt{S}$.

Proof. The first two statements are trivially satisfied by definition. The third follows by induction on the tree structure of $\mathtt{T}$. $\square$

Likewise, the lemma below permits to choose which branch to duplicate first.

Remark 7.10 (Width-level permutability of π^+)

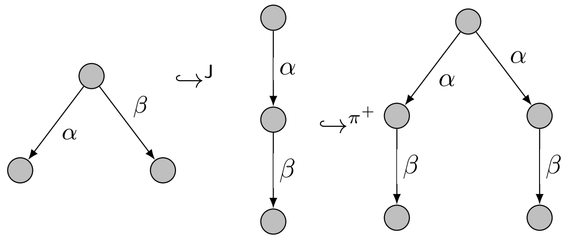
(i) If $\mathtt{T} \hookrightarrow^{\pi^+(\epsilon,i)} \circ \hookrightarrow^{\pi^+(\epsilon,j)} \mathtt{S}$ for $1 < j$ and $j \neq i+1$, then $\mathtt{T} \hookrightarrow^{\pi^+(\epsilon,j-1)} \circ \hookrightarrow^{\pi^+(\epsilon,i+1)} \hookrightarrow^{\sigma(\epsilon,1,2)} \mathtt{S}$.

(ii) If $\mathtt{T} \hookrightarrow^{\pi^+(n\mathbf{k}_n,i)} \circ \hookrightarrow^{\pi^+(l\mathbf{k}_l,j)} \mathtt{S}$ for $n \neq l$, then $\mathtt{T} \hookrightarrow^{\pi^+(l\mathbf{k}_l,j)} \circ \hookrightarrow^{\pi^+(n\mathbf{k}_n,i)} \mathtt{S}$.

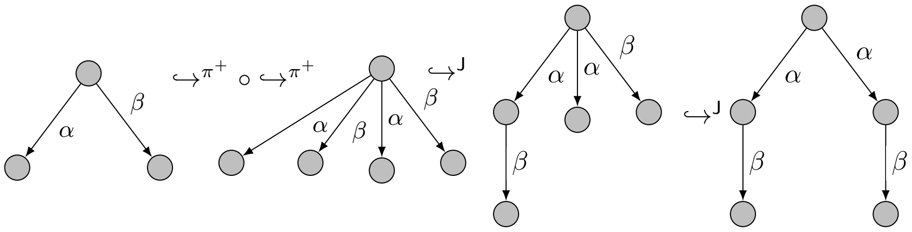
Therefore, we can show permutation of the application of arbitrary π^+-rules after a J-rule.

Proposition 7.11 *If* $\mathtt{T} \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\pi^+*} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\pi^+*} \circ \hookrightarrow^{\mathsf{J}*} \circ \hookrightarrow^{\sigma*} \mathtt{S}$.

Proof. If the J-rule and the π^+-rules are applied in different positions of the tree such that the performed transformations are disjoint, their permutation is proven straightforward as shown in Proposition 7.8. Therefore, we will use Lemma 7.9 and Remark 7.10 to reorganise the application of the π^+-rules to easily permute the disjoint cases. Otherwise, in cases where the subtree



(a) $\mathtt{T} \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\pi^+} \mathtt{S}$.



(b) $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \mathtt{S}$.

Fig. 6. $\mathtt{T} \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\pi^+} \mathtt{S} \Longrightarrow \mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \mathtt{S}$.

affected by the J-rule is duplicated, we use a unique labeling of the nodes to track the number of times those subtrees are duplicated. Specifically, we label the nodes with their positions. For $\mathtt{T} \hookrightarrow^{\mathsf{J}}\hookrightarrow^{*} \mathtt{S}$ such that J has been performed at position $n\mathbf{k} \in \mathsf{Pos}(\mathtt{T})$, we call the J-*label* to the label n. Moreover, if J has been performed at the empty position relating the ith and the jth children according to the notation of its definition, we call the *upper* J-*label* to the label i and the *lower* J-*label* to the label j.

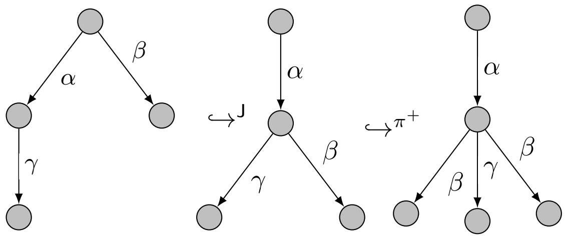
Therefore, the proof proceeds by induction on the length of the position in which the J-rule has been applied. For the J-rule applied at the empty position, we proceed by induction on the occurrences of the upper J-label. The inductive step is shown by Proposition 7.8 and Corollary 7.3, using Lemma 7.9 and Remark 7.10 to get a suitable reorganisation of the applications of the π^+-rules. The base case in which the upper J-label occurs once in $\mathtt{S}$ is shown by induction on the occurrences of the lower J-label using the same strategy.

Lastly, if the J-rule is applied at a non-empty position, we proceed by induction on the occurrences of the J-label. The base case follows by Proposition 7.8 using Lemma 7.9 and Remark 7.10 together with the inductive hypothesis on the position in which J has been performed. The inductive step is similarly proven by Proposition 7.8 and Corollary 7.3 using Lemma 7.9 and Remark 7.10. □

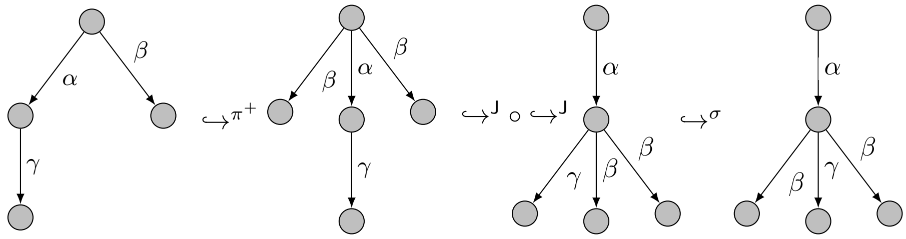
Corollary 7.12 *Let $\Omega_\diamond$ be a list of modal rewriting rules.*

(i) *If* $\mathtt{T} \hookrightarrow^{\lambda} \circ \hookrightarrow^{\pi^+ *} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\pi^+ *} \circ \hookrightarrow^{\lambda *} \mathtt{S}$ *by applying the same number of* π^+*-rules.*

(ii) *If* $\mathtt{T} \hookrightarrow^{\Omega_\diamond} \circ \hookrightarrow^{\pi^+ *} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\pi^+ *} \circ \hookrightarrow^{\Omega'_\diamond} \circ \hookrightarrow^{\sigma *} \mathtt{S}$ *for* $\Omega'_\diamond$ *a list of modal*



(a) $\mathtt{T} \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\pi^+} \mathtt{S}$.



(b) $\mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\sigma} \mathtt{S}$.

Fig. 7. $\mathtt{T} \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\pi^+} \mathtt{S} \Longrightarrow \mathtt{T} \hookrightarrow^{\pi^+} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\mathsf{J}} \circ \hookrightarrow^{\sigma} \mathtt{S}$.

rewriting rules.

Proof. The first statement follows by induction on the number of π^+-rules applied and Proposition 7.8. The second statement follows by induction on the length of $\Omega_\diamond$. The inductive step is shown by cases on the considered modal rewriting rule: the λ-rule case follows by the first statement and the J-rule case is shown by Proposition 7.11 and Corollary 7.2. □

Finally, we present the main theorem of the section.

Theorem 7.13 (Rewrite Normalization Theorem) *If* $\mathtt{T} \hookrightarrow^{*} \mathtt{S}$, *then* $\mathtt{T} \hookrightarrow^{\Omega} \mathtt{S}$ *for* Ω *a normal rewriting sequence.*

Proof. By induction on the number of rewriting steps that are performed using Corollaries 7.3, 7.5, 7.7 and 7.12. □

8 Conclusions and future work

We have provided a method for designing tree rewriting systems for different strictly positive logics and in particular re-cast the reflection calculus in this framework. Although not explicitly stated, it easily follows from the presented techniques that a rewriting system for $\mathbf{K}^+$ can be defined by only considering atomic, structural, extensional and π-rules, and we are confident that our framework will find applications in positive fragments for other (poly)modal logics. The use of abstract rewriting systems aids in the analysis of further properties like subformula property and admissibility of rules and provides a foundation for computationally efficient implementation.

Our work is based on a type-theoretic presentation of trees, which has the dual benefit of allowing for rules to be described in a precise and succinct manner and being particularly amenable to formalization in proof assistants. Many of our results have already been implemented in the proof assistant Coq and our goal is to fully formalize our work. Since rewriting normalization theorem and proof-theoretic investigations in general require checking multiple distinct cases in detail, the benefit of formalization is particularly clear in this type of proof, and indeed the community has been gravitating towards formalized proofs (see e.g. [18]). In the framework of **RC**, this has the added advantage that results often need not only to be *true,* but provable in suitable systems of arithmetic, and the latter can be made no more transparent than via formalization. Last but not least, proofs implemented in Coq can be automatically extracted into fully verified algorithms, paving the road to reasoners with the highest degree of reliability attainable by current technology.

References

[1] Baader, F., D. Calvanese, D. L. McGuinness, D. Nardi and P. F. Patel-Schneider, editors, "The Description Logic Handbook," Cambridge University Press, Cambridge, UK, 2007, 2 edition.

[2] Baader, F. and T. Nipkow, "Term rewriting and all that," Cambridge university press, 1998.

[3] Barendregt, H. P., "The lambda calculus - its syntax and semantics," Studies in logic and the foundations of mathematics **103**, North-Holland, 1985.

[4] Beklemishev, L., *A note on strictly positive logics and word rewriting systems*, in: S. Odintsov, editor, *Larisa Maksimova on Implication, Interpolation, and Definability*, Springer Verlag, 2018 .

[5] Beklemishev, L. D., *Provability algebras and proof-theoretic ordinals, I*, Annals of Pure and Applied Logic **128** (2004), pp. 103–124.

[6] Beklemishev, L. D., *Calibrating provability logic*, , **9**, College Publications, London, 2012 pp. 89–94.

[7] Beklemishev, L. D., *Positive provability logic for uniform reflection principles*, Annals of Pure and Applied Logic **165** (2014), pp. 82–105.

[8] Dashkov, E. V., *On the positive fragment of the polymodal provability logic GLP*, Mathematical Notes **91** (2012), pp. 318–333.

[9] Fitting, M., "Types, tableaus, and Gödel's god," Springer Science & Business Media, 2002.

[10] Foret, A., *Rewrite rule systems for modal propositional logic*, The Journal of Logic Programming **12** (1992), pp. 281–298.

[11] Genest, B., A. Muscholl, O. Serre and M. Zeitoun, *Tree pattern rewriting systems*, in: S. S. Cha, J.-Y. Choi, M. Kim, I. Lee and M. Viswanathan, editors, *Automated Technology for Verification and Analysis* (2008), pp. 332–346.

[12] Goubault-Larrecq, J. and I. Mackie, "Proof theory and automated deduction," Springer Science & Business Media, 2001.

[13] Japaridze, G., *The polymodal provability logic*, in: *Intensional logics and logical structure of theories: material from the Fourth Soviet-Finnish Symposium on Logic*, Metsniereba, Telavi, 1988 In Russian.

[14] Kikot, S., A. Kurucz, Y. Tanaka, F. Wolter and M. Zakharyaschev, *Kripke completeness of strictly positive modal logics over meet-semilattices with operators*, The Journal of Symbolic Logic **84** (2019), pp. 533–588.

[15] Newborn, M., "Automated theorem proving: theory and practice," Springer Science & Business Media, 2000.

[16] Pierce, B. C., "Types and programming languages," MIT press, 2002.
[17] Shamkanov, D., *Nested sequents for provability logic GLP*, Logic Journal of the IGPL **23** (2015), pp. 789–815.
[18] Shillito, I. and R. Goré, *Direct elimination of additive-cuts in GL4ip: verified and extracted*, in: D. Fernández-Duque, A. Palmigiano and S. Pinchinat, editors, *Advances in Modal Logic, AiML 2022, Rennes, France, August 22-25, 2022* (2022), pp. 429–449.
[19] Terese, "Term Rewriting Systems," Cambridge Tracts in Theoretical Computer Science **55**, Cambridge University Press, 2003.
[20] Wos, L., R. Overbeek, E. Lusk and J. Boyle, "Automated reasoning introduction and applications," McGraw-Hill, Inc., 1992.