Chapter 7
Dual Tableaux for Classical Modal Logics

7.1 Introduction

In a narrow sense, modal logic is a logic obtained from the classical logic by endowing it with unary propositional operations intuitively corresponding to 'it is necessary that' and 'it is possible that'. These operations are intensional, i.e., the truth of a formula built with the operation does not depend only on the truth of the subformula to which the operation is applied but also on a relevant state or a situation in which the truth is considered. A development of the semantics of modal logics in terms of a relational structure of states is due to Stig Kanger [Kan57] and Saul Kripke [Kri63]. Algebraic semantics of these standard modal logics is provided by Boolean algebras with normal and additive operations [JT52]. Since the origin of Kripke semantics, intensional logics have been introduced to computer science as an important tool for its formal methods.

In a broad sense modal logic is a field of studies of logics with intensional operations. The operations may have various intuitive interpretations and are relevant in a variety of fields. In logical theories intensional operations enable us to express qualitative degrees of truth, belief, knowledge, obligation, permission, etc. Elements of the relevant relational structures are then interpreted as possible worlds, situations, instants of time, etc. In computer science intensional operations serve as formal tools for expressing dynamic aspects of physical or cognitive processes. In these cases elements of the relational structures are interpreted as the states of a computer program, the tuples of a relational database, the objects of an information system with incomplete information, the agents of multiagents systems, etc.

The basic systems of modal logic in a modern form are due to Clarence Irving Lewis (see [Lew20, LL59, Zem73]). They evolved from his treatment of implication in search for elimination of the paradoxes of the classical implication of Frege–Russell. A development and broad range of research in modal logic and its applications can be traced through an extensive literature of the subject, see e.g., [Fey65, HC68, Seg71, Gal75, Gab76, Boo79, Che80, HC84, vB85, Boo93, Gol93, vB96, CZ97, HKT00, BdRV01, BvBW06].

E. Orłowska and J. Golińska-Pilarek, *Dual Tableaux: Foundations, Methodology, Case Studies*, Trends in Logic 33, DOI 10.1007/978-94-007-0005-5_7,

In this chapter we present a relational formalization of modal logics which originated in [Orł91]. Given a modal logic L, we show how one can construct a relational logic, $\mathsf{RL_L}$, and a dual tableau for $\mathsf{RL_L}$ so that it provides a validity checker for the logic L. We show that in fact the $\mathsf{RL_L}$-dual tableau does more: it can be used for proving entailment of an L-formula from a finite set of L-formulas, model checking of L-formulas in finite L-models, and verification of satisfaction of L-formulas in finite L-models. The relational formalization of modal logics presented in this chapter provides a paradigm for all the relational formalisms and dual tableaux considered in Parts III, IV, and V of the book.

7.2 Classical Propositional Logic

The vocabulary of the language of the classical propositional logic, PC, consists of the following pairwise disjoint sets:

- $\mathbb{V}$ – a countable set of propositional variables;
- $\{\neg, \vee, \wedge\}$ – the set of propositional operations of negation $\neg$, disjunction $\vee$, and conjunction $\wedge$.

The set of PC-formulas is the smallest set including $\mathbb{V}$ and closed with respect to the propositional operations. We admit the operations of implication, $\rightarrow$, and equivalence, $\leftrightarrow$, as the standard abbreviations, that is for all PC-formulas φ and ψ:

- $\varphi \rightarrow \psi \stackrel{\text{df}}{=} \neg\varphi \vee \psi$;
- $\varphi \leftrightarrow \psi \stackrel{\text{df}}{=} (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)$.

Let $\{0, 1\}$ be a two-element Boolean algebra whose elements represent truth-values 'false' and 'true', respectively. A PC-model is a structure $\mathcal{M} = (\{0, 1\}, v)$, where $v\colon \mathbb{V} \rightarrow \{0, 1\}$ is a valuation of propositional variables in $\{0, 1\}$. A PC-formula φ is said to be true in a PC-model $\mathcal{M} = (\{0, 1\}, v)$ whenever the following conditions hold:

- $\mathcal{M} \models p$ iff $v(p) = 1$, for every propositional variable p;
- $\mathcal{M} \models \neg\varphi$ iff $\mathcal{M} \not\models \varphi$;
- $\mathcal{M} \models \varphi \vee \psi$ iff $\mathcal{M} \models \varphi$ or $\mathcal{M} \models \psi$;
- $\mathcal{M} \models \varphi \wedge \psi$ iff $\mathcal{M} \models \varphi$ and $\mathcal{M} \models \psi$.

A PC-formula φ is PC-*valid* whenever it is true in all PC-models.

7.3 Propositional Modal Logics

The vocabulary of a *modal language* consists of the following pairwise disjoint sets:

- $\mathbb{V}$ – a countable set of propositional variables and/or constants;
- A set of relational constants;

- $\{\neg, \vee, \wedge\}$ – a set of classical propositional operations;
- A set of modal propositional operations is included in the set:

$$\{[R], \langle R\rangle, [[R]], \langle\langle R\rangle\rangle : R \text{ is a relational constant}\}.$$

The modal operations $\langle R\rangle$ and $\langle\langle R\rangle\rangle$ are definable in terms of $[R]$ and $[[R]]$, respectively, as follows:

$$\langle R\rangle\varphi \stackrel{\text{df}}{=} \neg[R]\neg\varphi, \quad \langle\langle R\rangle\rangle \stackrel{\text{df}}{=} \neg[[R]]\neg\varphi.$$

The propositional operations $[R]$, $\langle R\rangle$, $[[R]]$, and $\langle\langle R\rangle\rangle$ are referred to as *necessity, possibility, sufficiency*, and *dual sufficiency*, respectively.

The set of modal formulas is the smallest set including the set $\mathbb{V}$ and closed with respect to the propositional operations.

If the set of relational constants is a singleton set, say $\{R\}$, then the modal propositional operations $[R]$, $\langle R\rangle$, $[[R]]$, and $\langle\langle R\rangle\rangle$ are often written as $\Box$, $\Diamond$, $[[\,]]$, and $\langle\langle\rangle\rangle$, respectively. Some modal languages include propositional constants which are interpreted as singletons; they are referred to as *nominals*. If the cardinality of the set of relational constants is at least 2 or the language includes at least two different and not mutually definable modal operations, then the logic is referred to as *multimodal*.

Let a modal language be given. A *model* for the modal language is a structure $\mathcal{M} = (U, m)$ such that U is a non-empty set, whose elements are referred to as *states*, and m is a meaning function such that the following conditions are satisfied:

- $m(p) \subseteq U$, for every $p \in \mathbb{V}$;
- $m(R) \subseteq U \times U$, for every relational constant R.

The relations $m(R)$ are referred to as the *accessibility relations*.

A *frame* is a structure $\mathcal{F} = (U, m)$ such that U is a non-empty set and m is a map which assigns binary relations on U to relational constants. If there are finitely many relational constants in a modal language, then in the frames we often list explicitly all the corresponding relations and usually we denote them with the same symbols as the corresponding constants in the language. A model $\mathcal{M} = (U, m')$ is said to be *based on* a frame $\mathcal{F} = (U, m)$ whenever m is the restriction of m' to the set of relational constants.

A formula φ is said to be satisfied in a model $\mathcal{M}$ by a state $s \in U$, $\mathcal{M}, s \models \varphi$, whenever the following conditions are satisfied:

- $\mathcal{M}, s \models p$ iff $s \in m(p)$ for $p \in \mathbb{V}$;
- $\mathcal{M}, s \models \varphi \vee \psi$ iff $\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$;
- $\mathcal{M}, s \models \varphi \wedge \psi$ iff $\mathcal{M}, s \models \varphi$ and $\mathcal{M}, s \models \psi$;
- $\mathcal{M}, s \models \neg\varphi$ iff $\mathcal{M}, s \not\models \varphi$;
- $\mathcal{M}, s \models [R]\varphi$ iff for every $s' \in U$, if $(s, s') \in m(R)$, then $\mathcal{M}, s' \models \varphi$;
- $\mathcal{M}, s \models \langle R\rangle\varphi$ iff there is $s' \in U$ such that $(s, s') \in m(R)$ and $\mathcal{M}, s' \models \varphi$;

- $\mathcal{M}, s \models [[R]]\varphi$ iff for every $s' \in U$, if $\mathcal{M}, s' \models \varphi$, then $(s, s') \in m(R)$;
- $\mathcal{M}, s \models \langle\langle R\rangle\rangle\varphi$ iff there is $s' \in U$ such that $(s, s') \notin m(R)$ and $\mathcal{M}, s' \not\models \varphi$.

As usual, given a model $\mathcal{M}$ and a state s, '$\mathcal{M}, s \not\models \varphi$' is an abbreviation of 'not $\mathcal{M}, s \models \varphi$'. Throughout the book, by a modal logic we mean the pair L = (a modal language, a class of models of the language). Given a modal logic L, we write L-language and L-model for the relevant components of L. Formulas of the L-language are referred to as L-formulas.

An L-formula φ is said to be *true* in an L-model $\mathcal{M} = (U, m)$, $\mathcal{M} \models \varphi$, whenever for every $s \in U$, $\mathcal{M}, s \models \varphi$, and it is L-valid whenever it is true in all L-models. An L-formula φ is said to be *true* in the L-frame $\mathcal{F}$, $\mathcal{F} \models \varphi$ for short, whenever φ is true in all the L-models based on $\mathcal{F}$. Note that in every L-model the propositional operations $\neg$, $\vee$, and $\wedge$ receive their standard meaning as classical operations of negation, disjunction, and conjunction of PC, respectively.

Standard Modal Logics

The standard modal logics are K, T, B, $\mathsf{S4}$, and $\mathsf{S5}$. Their common language is a modal language with a single relational constant R and with the modal operations $[R]$ and $\langle R\rangle$.

The models of these logics are of the form $\mathcal{M} = (U, R, m)$ where:

- $\mathcal{M}$ is a K-model iff R is a binary relation on U;
- $\mathcal{M}$ is a T-model iff R is a reflexive relation on U (i.e., $1' \subseteq R$);
- $\mathcal{M}$ is a B-model iff it is a T-model such that R is a symmetric relation on U (i.e., $R^{-1} \subseteq R$);
- $\mathcal{M}$ is an $\mathsf{S4}$-model iff it is a T-model such that R is a transitive relation on U (i.e., $R\,;R \subseteq R$);
- $\mathcal{M}$ is a $\mathsf{S5}$-model iff it is a B-model and $\mathsf{S4}$-model (i.e., R is an equivalence relation on U).

All the standard modal logics are decidable. For details of the proof see e.g., [BvBW06].

7.4 Relational Formalization of Modal Logics

The logic $\mathsf{RL}(1, 1')$ serves as a basis for the relational formalisms for modal logics whose Kripke-style semantics is determined by frames with binary accessibility relations (see [Orł97b]). Let L be a modal logic. The relational logic RL_{L} appropriate for expressing L-formulas is obtained from $\mathsf{RL}(1, 1')$ by endowing its language with relational constants representing the accessibility relations from the models of L-language and with propositional constants of L (if there are any) which will be interpreted appropriately as relations.

The vocabulary of the relational logic $\mathsf{RL_L}$ consists of the symbols from the following pairwise disjoint sets:

- $\mathbb{OV}_{\mathsf{RL_L}}$ – a countable infinite set of object variables;
- $\mathbb{OC}_{\mathsf{RL_L}}$ – a countable (possibly empty) set of object constants;
- $\mathbb{RV}_{\mathsf{RL_L}}$ – a countable infinite set of relational variables;
- $\mathbb{RC}_{\mathsf{RL_L}} = \{1, 1'\} \cup \{R : R \text{ is a relational constant of } \mathsf{L}\} \cup \{C_c : c \text{ is a propositional constant of } \mathsf{L}\}$ – a set of relational constants;
- $\{-, \cup, \cap, ;, ^{-1}\}$ – the set of relational operations.

Object symbols, relational terms, and $\mathsf{RL_L}$-formulas are defined as in $\mathsf{RL}(1, 1')$-logic (see Sect. 2.3).

An $\mathsf{RL_L}$-structure is an $\mathsf{RL}(1, 1')$-model $\mathcal{M} = (U, m)$ (see Sect. 2.7) such that:

- $m(R) \subseteq U \times U$, for every relational constant R of L;
- $m(C_c) \subseteq U \times U$, for every propositional constant c of L.

An $\mathsf{RL_L}$-model is an $\mathsf{RL}(1, 1')$-model $\mathcal{M} = (U, m)$ such that:

- $m(C_c) = X \times U$, where $X \subseteq U$, for any propositional constant c of L; it follows that propositional constants of L are represented in $\mathsf{RL_L}$ as right ideal relations;
- The domains of relations $m(C_c)$ satisfy the constraints posed on propositional constants c in L-models; examples of such constants can be found in Sects. 11.3, 15.2, and 16.5;
- For all relational constants representing the accessibility relations of L, all the properties of these relations from L-models are assumed in $\mathsf{RL_L}$-models; many examples of such properties can be found in Sects. 7.5, 11.4, and 16.3.

If in a modal logic L there are finitely many accessibility relations, then in the $\mathsf{RL_L}$-models we list explicitly these relations and we denote them with the same symbols as the corresponding constants in the language.

As established in Sect. 2.7, the models of $\mathsf{RL_L}$ with $1'$ interpreted as the identity are referred to as standard $\mathsf{RL_L}$-models.

Translation

The translation of modal formulas into relational terms starts with a one-to-one assignment of relational variables to the propositional variables. Let τ' be such an assignment. Then the translation τ of formulas is defined inductively:

- $\tau(p) = \tau'(p) ; 1$, for any propositional variable $p \in \mathbb{V}$;
- $\tau(c) = C_c ; 1$, for any propositional constant $c \in \mathbb{V}$;
- $\tau(\neg\varphi) = -\tau(\varphi)$;
- $\tau(\varphi \vee \psi) = \tau(\varphi) \cup \tau(\psi)$;
- $\tau(\varphi \wedge \psi) = \tau(\varphi) \cap \tau(\psi)$;

and for every relational constant R of L:

- $\tau(\langle R\rangle\varphi) = R ; \tau(\varphi)$;
- $\tau([R]\varphi) = -(R ; -\tau(\varphi))$;

- $\tau(\langle\langle R \rangle\rangle \varphi) = -R \,;\, -\tau(\varphi)$;
- $\tau([[R]]\varphi) = -(-R \,;\, \tau(\varphi))$.

It follows that translation of defined propositional operations $\rightarrow$ and $\leftrightarrow$ is:

- $\tau(\varphi \rightarrow \psi) = -\tau(\varphi) \cup \tau(\psi)$;
- $\tau(\varphi \leftrightarrow \psi) = (-\tau(\varphi) \cup \tau(\psi)) \cap (-\tau(\psi) \cup \tau(\varphi)))$.

Hence, when passing from modal formulas to relational terms we replace propositional variables and constants by relational variables and relational constants, respectively, and propositional operations by relational operations. The crucial point here is that the accessibility relation is 'taken out' of the modal operation and it becomes an argument of an appropriate relational operation. In particular, possibility operation is replaced by the relational composition of two relations: the relation representing an accessibility relation and the relation resulting from the translation of the formula which is in the scope of the possibility operation. In this way to any formula φ of a modal logic there is associated a relational term $\tau(\varphi)$. The above translation assigns to any modal formula a right ideal relation i.e., a relation Q that satisfies $Q = Q \,;\, 1$. It follows from the following proposition:

Proposition 7.4.1. *For every set U, the following conditions are satisfied:*

1. *The family of right ideal relations on U is closed on $-$, $\cup$, and $\cap$;*
2. *For any relation R on U and any right ideal relation P on U, $R \,;\, P$ is a right ideal relation;*
3. *If P is a right ideal relation on U, then for all $s, s' \in U$: $(s, s') \in P$ iff for every $t \in U$, $(s, t) \in P$.*

Proof. 1. and 3. follow directly from the definition of right ideal relations. For 2., note that since $P = P \,;\, 1$, $R \,;\, P = R \,;\, (P \,;\, 1) = (R \,;\, P) \,;\, 1$. □

In some examples of dual tableaux proofs presented in the book a simpler translation of the formula to be proved is sufficient such that $\tau(p) = \tau'(p)$ for every propositional variable p appearing in the formula.

The translation τ is defined so that it preserves validity of formulas.

Proposition 7.4.2. *Let L be a modal logic and let φ be an L-formula. Then, for every L-model $\mathcal{M} = (U, m)$ there exists an RL_{L}-model $\mathcal{M}' = (U, m')$ with the same universe as that of $\mathcal{M}$ such that for all $s, s' \in U$, $\mathcal{M}, s \models \varphi$ iff $(s, s') \in m'(\tau(\varphi))$.*

Proof. Let φ be an L-formula and let $\mathcal{M} = (U, m)$ be an L-model. We define an RL_{L}-model $\mathcal{M}' = (U, m')$ as follows:

- $m'(1) = U \times U$;
- $m'(1')$ is the identity on U;
- $m'(\tau(p)) = m(p) \times U$, for every propositional variable p;
- $m'(\tau(c)) = m(c) \times U$, for any propositional constant $c \in \mathbb{V}$;

- $m'(R) = m(R)$;
- m' extends to all the compound terms as in $\mathsf{RL}(1, 1')$-models.

Now, we prove the proposition by induction on the complexity of formulas. Let $s, s' \in U$.

Let $\varphi = p$, $p \in \mathbb{V}$. Then $\mathcal{M}, s \models p$ iff $s \in m(p)$ iff $(s, s') \in m'(\tau(p))$, since $m'(\tau(p))$ is a right ideal relation.

Let $\varphi = \psi \vee \vartheta$. Then $\mathcal{M}, s \models \psi \vee \vartheta$ iff $\mathcal{M}, s \models \psi$ or $\mathcal{M}, s \models \vartheta$ iff, by the induction hypothesis, $(s, s') \in m'(\tau(\psi))$ or $(s, s') \in m'(\tau(\vartheta))$ iff $(s, s') \in m'(\tau(\psi)) \cup m'(\tau(\vartheta))$ iff $(s, s') \in m'(\tau(\psi \vee \vartheta))$.

Let $\varphi = \langle R \rangle \psi$. By the induction hypothesis, for all $t, s' \in U$, we have $\mathcal{M}, t \models \psi$ iff $(t, s') \in m'(\tau(\psi))$. Therefore, $\mathcal{M}, s \models \varphi$ iff there exists $t \in U$ such that $(s, t) \in m(R)$ and $\mathcal{M}, t \models \psi$ iff, by the induction hypothesis, there exists $t \in U$ such that $(s, t) \in m'(R)$ and $(t, s') \in m'(\tau(\psi))$ iff $(s, s') \in m'(R\,;\tau(\psi))$ iff $(s, s') \in m'(\tau(\varphi))$.

Let $\varphi = [[R]]\psi$. Then $\mathcal{M}, s \models \varphi$ iff for every $t \in U$, if $\mathcal{M}, t \models \psi$, then $(s, t) \in m(R)$ iff, by the induction hypothesis, for every $t \in U$, if $(t, s') \in m'(\tau(\psi))$, then $(s, t) \in m'(R)$ iff $(s, s') \notin m'(-R\,;\tau(\psi))$ iff $(s, s') \in m'(\tau(\varphi))$.

In the remaining cases the proofs are similar. □

Proposition 7.4.3. *Let* L *be a modal logic and let* φ *be an* L*-formula. Then, for every standard* RL_{L}*-model* $\mathcal{M}' = (U, m')$ *there exists an* L*-model* $\mathcal{M} = (U, m)$ *with the same universe as that of* $\mathcal{M}'$ *such that for all* $s, s' \in U$*, the condition of Proposition 7.4.2 holds.*

Proof. Let φ be an L-formula and let $\mathcal{M}' = (U, m')$ be a standard RL_{L}-model. We define an L-model $\mathcal{M} = (U, m)$ as follows:

- $m(p) = \{x \in U : \text{there exists } y \in U, (x, y) \in m'(\tau(p))\}$, for every propositional variable p;
- For every propositional constant c, $s \in m(c)$ iff there is $s' \in U$ such that $(s, s') \in m'(\tau(c))$;
- $m(R) = m'(R)$.

We can prove that $\mathcal{M}, s \models \varphi$ iff $(s, s') \in m'(\tau(\varphi))$ in a similar way as in Proposition 7.4.2. □

Proposition 7.4.4. *Let* L *be a modal logic and let* φ *be an* L*-formula. Then, for every* L*-model* $\mathcal{M}$ *there exists an* RL_{L}*-model* $\mathcal{M}'$ *such that for all object variables* x *and* y*,* $\mathcal{M} \models \varphi$ *iff* $\mathcal{M}' \models x\tau(\varphi)y$*.*

Proof. Let φ be an L-formula, let $\mathcal{M} = (U, m)$ be an L-model. We construct a standard RL_{L}-model $\mathcal{M}' = (U, m')$ as in the proof of Proposition 7.4.2. Let x and y be any object variables. Assume $\mathcal{M} \models \varphi$. Suppose there exists a valuation v in $\mathcal{M}'$ such that $\mathcal{M}', v \not\models x\tau(\varphi)y$. Then $(v(x), v(y)) \notin m'(\tau(\varphi))$. However, by Proposition 7.4.2, models $\mathcal{M}$ and $\mathcal{M}'$ satisfy $\mathcal{M}, v(x) \models \varphi$ iff $(v(x), v(y)) \in m'(\tau(\varphi))$. Therefore, $\mathcal{M}, v(x) \not\models \varphi$, and hence $\mathcal{M} \not\models \varphi$, a contradiction. Now, assume $\mathcal{M}' \models x\tau(\varphi)y$. Suppose there is $s \in U$ such that $\mathcal{M}, s \not\models \varphi$. Let s' be any element of U. By Proposition 7.4.2, $\mathcal{M}, s \models \varphi$ if and only if $(s, s') \in m'(\tau(\varphi))$. Let v be a valuation in $\mathcal{M}'$ such that $v(x) = s$ and $v(y) = s'$. Since $\mathcal{M}, s \not\models \varphi$, $(v(x), v(y)) \notin m'(\tau(\varphi))$, so $\mathcal{M}', v \not\models x\tau(\varphi)y$, and hence $\mathcal{M}' \not\models x\tau(\varphi)y$, a contradiction. □

Due to Proposition 7.4.3, the following can be proved:

Proposition 7.4.5. *Let* L *be a modal logic and let* φ *be an* L*-formula. Then, for every standard* RL_L*-model* $\mathcal{M}'$ *there exists an* L*-model* $\mathcal{M}$ *such that for all object variables* x *and* y*, the condition of Proposition 7.4.4 holds.*

From Theorem 2.7.2, Propositions 7.4.4, and 7.4.5, we get:

Theorem 7.4.1. *Let* L *be a modal logic. Then, for every* L*-formula* φ *and for all object variables* x *and* y*, the following conditions are equivalent:*

1. φ *is* L*-valid;*
2. $x\tau(\varphi)y$ *is* RL_L*-valid.*

Proof. (1. → 2.) Let φ be L-valid. Suppose $x\tau(\varphi)y$ is not RL_L-valid. Then, there exists a standard RL_L-model $\mathcal{M}$ such that $\mathcal{M} \not\models x\tau(\varphi)y$. By Proposition 7.4.5, there is an L-model $\mathcal{M}'$ such that $\mathcal{M}' \not\models \varphi$, which contradicts the assumption of L-validity of φ.

(2. → 1.) Let φ be an L-formula such that $x\tau(\varphi)y$ is RL_L-valid. Suppose φ is not L-valid. Then there exists an L-model $\mathcal{M}$ such that $\mathcal{M} \not\models \varphi$. By Proposition 7.4.4, there exists an RL_L-model $\mathcal{M}'$ such that $\mathcal{M}' \not\models x\tau(\varphi)y$, a contradiction. □

Relational dual tableaux for modal logics are extensions of $\mathsf{RL}(1, 1')$-dual tableau. We add to the $\mathsf{RL}(1,' 1)$-system the specific rules and/or axiomatic sets that reflect properties of the specific relational constants corresponding to accessibility relations and propositional constants (if there are any) of the logic in question. Given a logic L, a relational dual tableau for L, RL_L-dual tableau, enables us to prove facts about the relations from the models of L expressed in the language of L or in the language of RL_L.

Given a modal logic L, an RL_L-set is a finite set of RL_L-formulas such that the first-order disjunction of its members is true in all RL_L-models. Correctness of a rule is defined in a similar way as in the relational logics of classical algebras of binary relations (see Sect. 2.4).

Following the method of proving soundness and completeness of RL-dual tableau described in Sect. 2.6, we can prove soundness and completeness of the RL_L-dual tableau in a similar way. To prove soundness, it suffices to show that all the rules are RL_L-correct and all the axiomatic sets are RL_L-sets (see Proposition 7.5.1). In order to prove completeness we need to prove the closed branch property, the branch model property, and the satisfaction in branch model property. Then the completeness proof is the same as the completeness proof of $\mathsf{RL}(1, 1')$-dual tableau (see Sects. 2.5 and 2.7). In Table 7.1 we recall the main facts that have to be proved.

Table 7.1 The key steps in the completeness proof of relational dual tableaux

(1) **Closed Branch Property**: For any branch of an $\mathsf{RL_L}$-proof tree, if $xRy \in b$ and $x{-}Ry \in b$ for an atomic term R, then the branch can be closed;
(2) **Branch Model Property**: Define the branch model $\mathcal{M}^b$ determined by an open branch b of an $\mathsf{RL_L}$-proof tree and prove that it is an $\mathsf{RL_L}$-model;
(3) **Satisfaction in Branch Model Property**: For every branch b of an $\mathsf{RL_L}$-proof tree and for every $\mathsf{RL_L}$-formula φ, the branch model $\mathcal{M}^b$ and the identity valuation v^b in $\mathcal{M}^b$ satisfy: If $\mathcal{M}^b, v^b \models \varphi$, then $\varphi \notin b$.

7.5 Dual Tableaux for Standard Modal Logics

Let L be a standard modal logic as presented in Sect. 7.3 (see p. 146). The relational logic appropriate for expressing L-formulas, $\mathsf{RL_L}$, is obtained from logic $\mathsf{RL}(1, 1')$ by expanding its language with a relational constant R representing the accessibility relation from the models of L-language. If a relation R in the models of logic L is assumed to satisfy some conditions, e.g., reflexivity (logic T), symmetry (logic B), transitivity (logic $\mathsf{S4}$) etc., then in the corresponding logic $\mathsf{RL_L}$ we add the respective conditions as the axioms of its models. The translation of a modal formula of L into a relational term of $\mathsf{RL_L}$ is defined as in Sect. 7.4.

By Theorem 7.4.1 the following holds:

Theorem 7.5.1. *For every formula φ of a standard modal logic L and for all object variables x and y, the following conditions are equivalent:*

1. *φ is L-valid;*
2. *$xt(\varphi)y$ is $\mathsf{RL_L}$-valid.*

Dual tableaux for standard modal logics in their relational formalizations are constructed as follows. We add to the $\mathsf{RL}(1, 1')$-dual tableau the following rules:

- Logic T: (ref R),
- Logic B: (ref R) and (sym R),
- Logic $\mathsf{S4}$: (ref R) and (tran R),
- Logic $\mathsf{S5}$: (ref R), (sym R), and (tran R).

We recall that these rules are of the form (see Sect. 6.6):

For all object symbols x and y,

$$\text{(ref } R) \quad \frac{xRy}{x1'y,\ xRy} \qquad\qquad \text{(sym } R) \quad \frac{xRy}{yRx}$$

$$\text{(tran } R) \quad \frac{xRy}{xRz,\ xRy \mid zRy,\ xRy} \qquad z \text{ is any object symbol}$$

As defined in Sect. 7.4, given a standard modal logic L, an RL_L-structure is of the form $\mathcal{M} = (U, R, m)$, where (U, m) is an $\mathsf{RL}(1, 1')$-model and R is a binary relation on U. An RL_L-model is an RL_L-structure such that relation R satisfies the constraints posed in L-models.

Theorem 7.5.2 (Correspondence). *Let* L *be a standard modal logic and let* $\mathcal{K}$ *be a class of* RL_L*-structures. Relation* R *is reflexive (resp. symmetric, transitive) in all structures of* $\mathcal{K}$ *iff the rule (ref* R*), (resp. (sym* R*), (tran* R*)) is* $\mathcal{K}$*-correct.*

For the proof see Theorem 6.6.1. Theorem 7.5.2 leads to:

Proposition 7.5.1. *Let* L *be a standard modal logic. Then:*

1. *The* RL_L*-rules are* RL_L*-correct;*
2. *The* RL_L*-axiomatic sets are* RL_L*-sets.*

The notions of an RL_L-proof tree, a closed branch of such a tree, a closed RL_L-proof tree, and RL_L-provability are defined as in Sect. 2.4.

We recall that the completion conditions determined by the rules (ref R), (sym R), and (tran R) are:

For all object symbols x and y,

Cpl(ref R) If $xRy \in b$, then $x1'y \in b$;
Cpl(sym R) If $xRy \in b$, then $yRx \in b$;
Cpl(tran R) If $xRy \in b$, then for every object symbol z, either $xRz \in b$ or $zRy \in b$.

The notions of a complete branch of an RL_L-proof tree, a complete RL_L-proof tree, and an open branch of an RL_L-proof tree are defined as in RL-logic (see Sect. 2.5). In order to prove completeness, we need to define a branch model and to show the three theorems of Table 7.1.

The branch model is defined as in the completeness proof of $\mathsf{RL}(1, 1')$-dual tableau, that is $R^b = m^b(R) = \{(x, y) \in U^b \times U^b : xRy \notin b\}$. Using the completion conditions, it is easy to show that R^b satisfies the conditions assumed in the corresponding L-models (see the proof of Proposition 6.6.3). Hence, the branch model property is satisfied. In similar way as in $\mathsf{RL}(1, 1')$-dual tableau (see also RL_EQ-dual tableau in Sect. 6.6), we can prove the closed branch property and the satisfaction in branch model property:

Proposition 7.5.2 (Satisfaction in Branch Model Property). *For every open branch* b *of an* RL_L*-proof tree and for every* RL_L*-formula* φ*, if* $\mathcal{M}^b, v^b \models \varphi$*, then* $\varphi \notin b$*.*

Then, completeness can be proved as for $\mathsf{RL}(1, 1')$-dual tableau.

Theorem 7.5.3 (Soundness and Completeness of Relational Logics for Standard Modal Logics). *Let* L *be a standard modal logic and let* φ *be an* RL_L*-formula. Then, the following conditions are equivalent:*

1. *φ is* $\mathsf{RL_L}$*-valid;*
2. *φ is true in all standard* $\mathsf{RL_L}$*-models;*
3. *φ is* $\mathsf{RL_L}$*-provable.*

Finally, by Theorem 7.5.1 and Theorem 7.5.3, we obtain:

Theorem 7.5.4 (Relational Soundness and Completeness of Standard Modal Logics). *Let* L *be a standard modal logic and let φ be an* L*-formula. Then for all object variables x and y, the following conditions are equivalent:*

1. *φ is* L*-valid;*
2. *$x\tau(\varphi)y$ is* $\mathsf{RL_L}$*-provable.*

Example. We present a translation and a relational proof of a formula of logic K. Note that $\mathsf{RL_K}$-proof system is exactly the same as $\mathsf{RL}(1, 1')$-system, because in K-models the accessibility relation is an arbitrary binary relation. Consider the following K-formula:

$$\varphi = ([R]p \wedge [R]q) \rightarrow [R](p \wedge q).$$

For reasons of simplicity, let $\tau(p) = P$ and $\tau(q) = Q$. The translation $\tau(\varphi)$ of the formula φ into a relational term of logic $\mathsf{RL_K}$ is:

$$-[-(R\,;-P) \cap -(R\,;-Q)] \cup -(R\,;-(P \cap Q)).$$

We show that the formula φ is K-valid, by showing that $x\tau(\varphi)y$ is $\mathsf{RL_K}$-valid. Figure 7.1 presents its $\mathsf{RL_K}$-proof.

7.6 Entailment in Modal Logics

The logic $\mathsf{RL}(1, 1')$ can be used to verify entailment of formulas of non-classical logics, provided that they can be translated into binary relations. Let L be a modal logic. In order to verify the entailment we apply the method presented in Sect. 2.11. We translate L-formulas in question into relational terms of the logic $\mathsf{RL_L}$ and then we use the method of verification of entailment for $\mathsf{RL_L}$-logic as shown in Sect. 2.11.

For example, in every model $\mathcal{M}$ of K-logic the truth of the formula $p \rightarrow q$ in $\mathcal{M}$ implies the truth of $[R]p \rightarrow [R]q$ in $\mathcal{M}$. The translation of these formulas to $\mathsf{RL_K}$-terms is:

$$\tau(p \rightarrow q) = -P \cup Q,$$

$$\tau([R]p \rightarrow [R]q) = --(R\,;-P) \cup -(R\,;-Q),$$

where for simplicity $\tau(p) = P$ and $\tau(q) = Q$. To verify the entailment we need to show that $-P \cup Q = 1$ implies $--(R\,;-P) \cup -(R\,;-Q) = 1$. According to Proposition 2.2.1(7.), we need to show that the formula:

$$x[(1\,;-(-P \cup Q)\,;1) \cup --(R\,;-P) \cup -(R\,;-Q)]y$$

is $\mathsf{RL_K}$-provable. Figure 7.2 presents an $\mathsf{RL_K}$-proof of this formula.

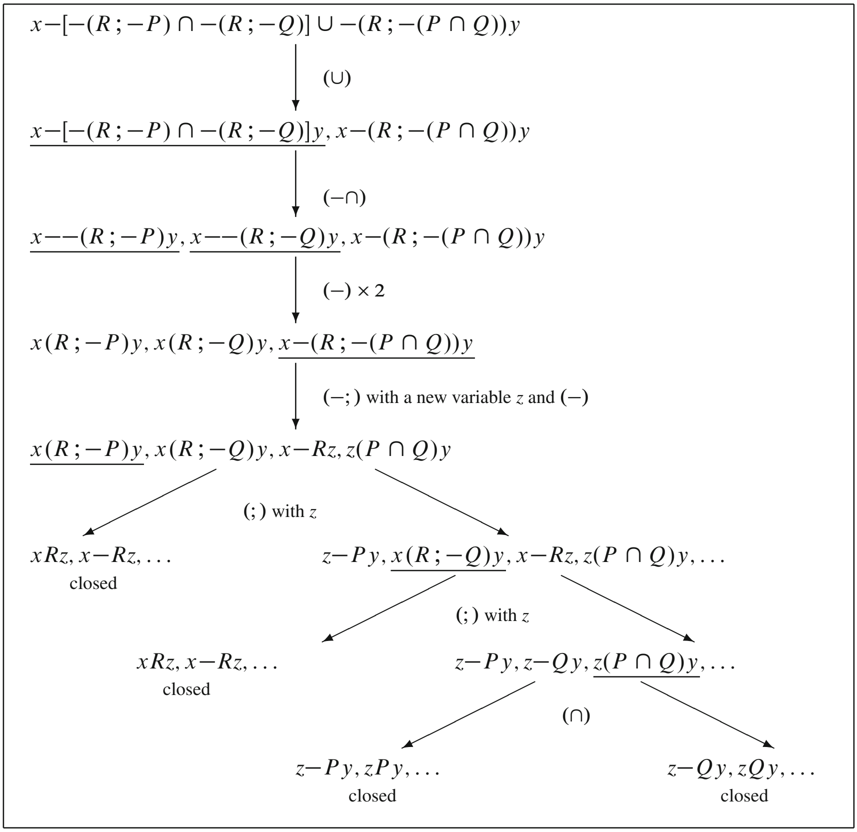

Fig. 7.1 An $\mathsf{RL_K}$-proof of K-formula $([R]p \wedge [R]q) \rightarrow [R](p \wedge q)$

Furthermore, observe that in every model $\mathcal{M}$ of K-logic the truth of $R\,;R \subseteq R$ in $\mathcal{M}$ implies the truth of $[R]p \rightarrow [R][R]p$ in $\mathcal{M}$. The translation of the latter formula to $\mathsf{RL_K}$-term is:

$$--(R\,;-P) \cup -(R\,;--(R\,;-P)),$$

where for simplicity $\tau(p) = P$. To verify the entailment we need to show that $-(R\,;R) \cup R = 1$ implies $--(R\,;-P) \cup -(R\,;--(R\,;-P)) = 1$. According to Proposition 2.2.1(7.), we need to show that the formula:

$$x[(1\,;-(-(R\,;R) \cup R)\,;1) \cup --(R\,;-P) \cup -(R\,;--(R\,;-P))]y$$

is $\mathsf{RL_K}$-provable. Figure 7.3 presents its $\mathsf{RL_K}$-proof.

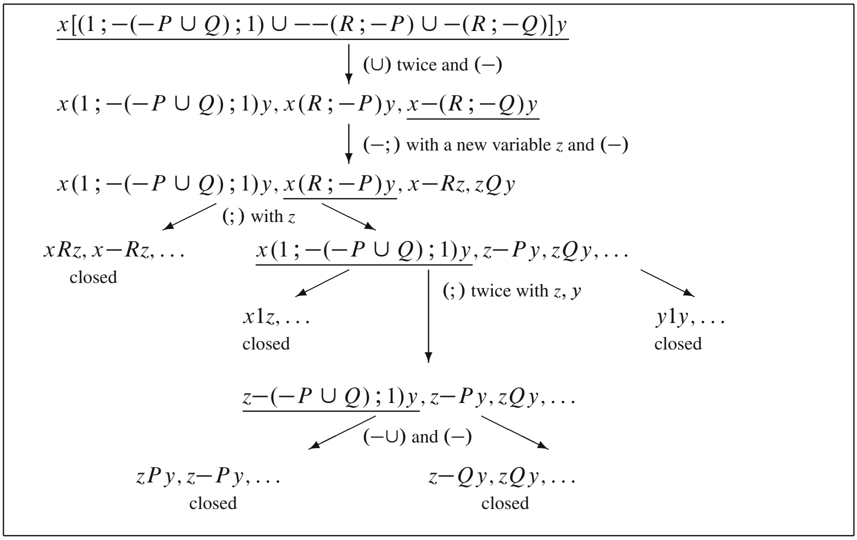

Fig. 7.2 An $\mathsf{RL_K}$-proof showing that $p \to q$ entails $[R]p \to [R]q$

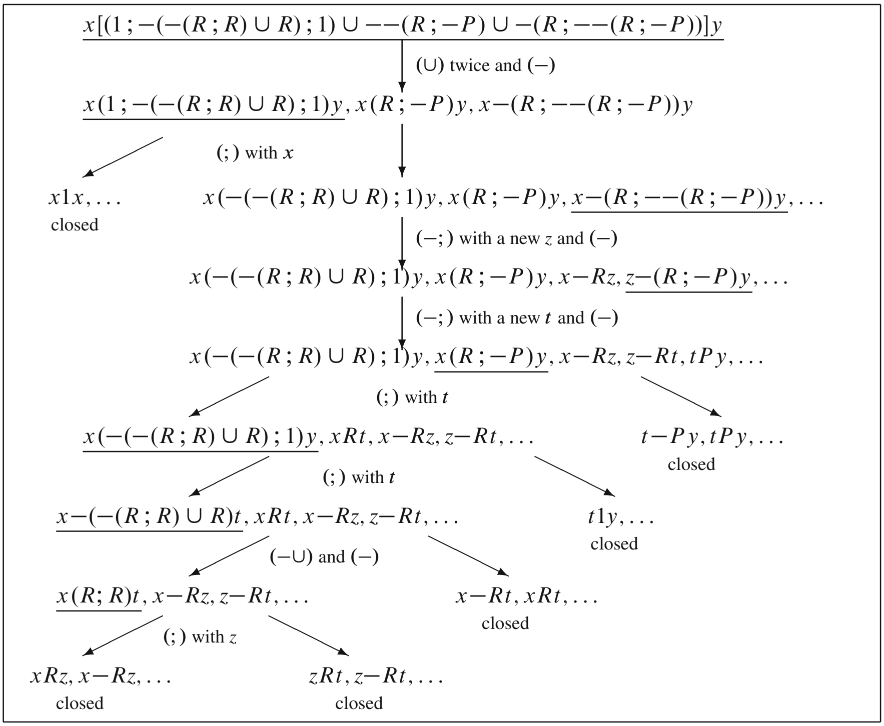

Fig. 7.3 An $\mathsf{RL_K}$-proof showing that $R ; R \subseteq R$ entails $[R]p \to [R][R]p$

7.7 Model Checking in Modal Logics

The method presented in Sect. 3.4 can be used for model checking in finite models of modal logics. The general idea is as follows. Let $\mathcal{M}$ be a finite model of a modal logic L and let φ be an L-formula. In order to verify whether φ is true in $\mathcal{M}$, we construct a relational logic $\mathsf{RL_L}$ and an $\mathsf{RL_L}$-model $\mathcal{M}'$ such that for all object variables x and y, the problem '$\mathcal{M} \models \varphi$?' is equivalent to the problem '$\mathcal{M}' \models x\tau(\varphi)y$?'. Then, we apply the method of model checking for the relational logic $\mathsf{RL_L}$, the model $\mathcal{M}'$, and the formula $x\tau(\varphi)y$ as presented in Sect. 3.4. For that purpose, we consider an instance $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$ of the logic $\mathsf{RL_L}$. Then, we obtain:

Theorem 7.7.1 (Relational Model Checking in Modal Logics). *For every L-formula φ, for every finite L-model $\mathcal{M}$, and for all object variables x and y, the following statements are equivalent:*

1. $\mathcal{M} \models \varphi$;
2. $x\tau(\varphi)y$ *is* $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$*-provable.*

By way of example, consider modal logic K. Let $\mathcal{M} = (U, R, m)$ be a K-model such that $U = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $m(p) = \{\mathtt{a}\}$, and the accessibility relation is $R = \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{a}, \mathtt{c})\}$. Let φ be the formula of the form $\neg\langle R\rangle\langle R\rangle p$. Let us consider the problem: 'is φ true in $\mathcal{M}$?'. The translation of the formula φ is:

$$\tau(\varphi) = -(R\,;(R\,;(P\,;1))),$$

where $\tau'(p) = P$. Using the construction from the proof of Proposition 7.4.2 it is easy to prove that there exists an $\mathsf{RL_K}$-model $\mathcal{M}'$ such that for all object variables x and y, $\mathcal{M} \models \varphi$ iff $\mathcal{M}' \models x\tau(\varphi)y$.

The model $\mathcal{M}' = (U', R', m')$ is an $\mathsf{RL_K}$-model such that $U' = U$, $R' = R$, and m' is the meaning function satisfying:

$$m'(P) = \{(\mathtt{a}, \mathtt{a}), (\mathtt{a}, \mathtt{b}), (\mathtt{a}, \mathtt{c})\}.$$

Let x and y be any object variables. The model checking problem 'is φ true in $\mathcal{M}$?' is equivalent to the problem 'is the formula $x\tau(\varphi)y$ true in $\mathcal{M}'$?'. For the latter we apply the method presented in Sect. 3.4. The vocabulary of the language adequate for testing whether $\mathcal{M}' \models x\tau(\varphi)y$ consists of the following pairwise disjoint sets of symbols:

- $\mathbb{OV}_{\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}}$ - a countable infinite set of object variables;
- $\mathbb{OC}_{\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}} = \{c_\mathtt{a}, c_\mathtt{b}, c_\mathtt{c}\}$ – the set of object constants;
- $\mathbb{RC}_{\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}} = \{R, P, 1, 1'\}$ – the set of relational constants;
- $\{-, \cup, \cap, ;, ^{-1}\}$ – the set of relational operations.

An $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$-model is the structure $\mathcal{N} = (W, R, n)$, where:

- $W = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$;
- $n(c_\mathtt{a}) = \mathtt{a}$, $n(c_\mathtt{b}) = \mathtt{b}$, and $n(c_\mathtt{c}) = \mathtt{c}$;

- $n(1) = W \times W$;
- $n(1') = \{(\mathtt{a}, \mathtt{a}), (\mathtt{b}, \mathtt{b}), (\mathtt{c}, \mathtt{c})\}$;
- $n(P) = \{(\mathtt{a}, \mathtt{a}), (\mathtt{a}, \mathtt{b}), (\mathtt{a}, \mathtt{c})\}$;
- $R = n(R) = \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{a}, \mathtt{c})\}$;
- n extends to all the compound terms as in RL-models.

The rules of $\mathsf{RL}_{\mathcal{M}', x\tau(\varphi)y}$-dual tableau which are specific for the model checking problem in question have the following forms:

$$(-R\mathtt{ij}) \quad \frac{x{-}Ry}{x1'c_{\mathtt{i}}, x{-}Ry \mid y1'c_{\mathtt{j}}, x{-}Ry}$$

for every $(\mathtt{i}, \mathtt{j}) \in \{(\mathtt{a}, \mathtt{a}), (\mathtt{b}, \mathtt{a}), (\mathtt{b}, \mathtt{b}), (\mathtt{c}, \mathtt{a}), (\mathtt{c}, \mathtt{b}), (\mathtt{c}, \mathtt{c})\}$;

$$(-P\mathtt{ij}) \quad \frac{x{-}Py}{x1'c_{\mathtt{i}}, x{-}Py \mid y1'c_{\mathtt{j}}, x{-}Py}$$

for every $(\mathtt{i}, \mathtt{j}) \in \{(\mathtt{b}, \mathtt{a}), (\mathtt{b}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{c}, \mathtt{a}), (\mathtt{c}, \mathtt{b}), (\mathtt{c}, \mathtt{c})\}$;

$$(1') \quad \frac{}{x{-}1'c_{\mathtt{a}} \mid x{-}1'c_{\mathtt{b}} \mid x{-}1'c_{\mathtt{c}}}$$

$$(\mathtt{i} \neq \mathtt{j}) \quad \frac{}{c_{\mathtt{i}}1'c_{\mathtt{j}}} \quad \text{for any } \mathtt{i}, \mathtt{j} \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \mathtt{i} \neq \mathtt{j}$$

The $\mathsf{RL}_{\mathcal{M}', x\tau(\varphi)y}$-axiomatic sets are:

- $\{c_{\mathtt{i}}Rc_{\mathtt{j}}\}$, for every $(\mathtt{i}, \mathtt{j}) \in \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{a}, \mathtt{c})\}$;
- $\{c_{\mathtt{i}}{-}Rc_{\mathtt{j}}\}$, for every$(\mathtt{i}, \mathtt{j}) \in \{(\mathtt{a}, \mathtt{a}), (\mathtt{b}, \mathtt{a}), (\mathtt{b}, \mathtt{b}), (\mathtt{c}, \mathtt{a}), (\mathtt{c}, \mathtt{b}), (\mathtt{c}, \mathtt{c})\}$;
- $\{c_{\mathtt{i}}Pc_{\mathtt{j}}\}$, for every $(\mathtt{i}, \mathtt{j}) \in \{(\mathtt{a}, \mathtt{a}), (\mathtt{a}, \mathtt{b}), (\mathtt{a}, \mathtt{c})\}$;
- $\{c_{\mathtt{i}}{-}Pc_{\mathtt{j}}\}$, for every $(\mathtt{i}, \mathtt{j}) \in \{(\mathtt{b}, \mathtt{a}), (\mathtt{b}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{c}, \mathtt{a}), (\mathtt{c}, \mathtt{b}), (\mathtt{c}, \mathtt{c})\}$.

Let x and y be object variables. The truth of φ in model $\mathcal{M}$ is equivalent to $\mathsf{RL}_{\mathcal{M}', x\tau(\varphi)y}$-provability of $x\tau(\varphi)y$. Figure 7.4 presents an $\mathsf{RL}_{\mathcal{M}', x\tau(\varphi)y}$-proof of $x\tau(\varphi)y$. The subtree Π_1 is presented in Fig. 7.5. The subtrees Π_2 and Π_3 are presented in Figs. 7.6 and 7.7, respectively. Observe that in a diagram of Fig. 7.7 the applications of the rules $(-P\mathtt{ca})$, $(-P\mathtt{cb})$, and $(-P\mathtt{cc})$ result in the nodes with formulas $v1'c_{\mathtt{a}}$, $v1'c_{\mathtt{b}}$, and $v1'c_{\mathtt{c}}$. Therefore, in the picture we identify all these nodes.

7.8 Verification of Satisfaction in Modal Logics

The method of verification of satisfaction in finite models presented in Sect. 3.5 can be also used in the case of standard modal logics. Let $\mathcal{M} = (U, R, m)$ be a finite model of a modal logic L, let φ be an L-formula, and let $\mathtt{a} \in U$ be a state. In order to verify whether φ is satisfied in $\mathcal{M}$ by the state $\mathtt{a}$, we construct a relational logic

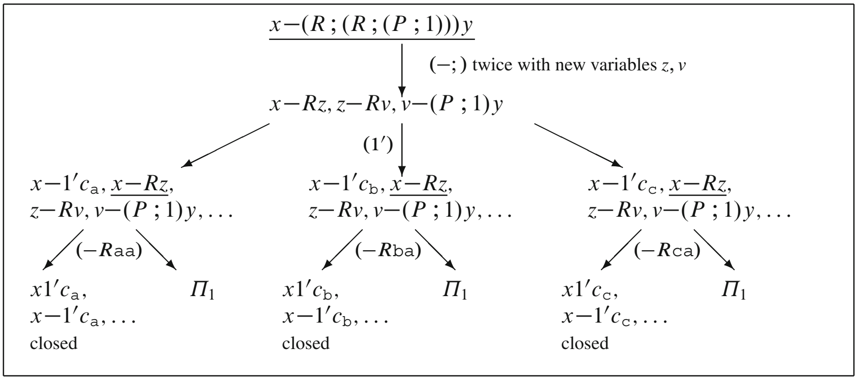

Fig. 7.4 An $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$-proof of the truth of K-formula $\langle R\rangle\langle R\rangle p \rightarrow \langle R\rangle p$ in the model $\mathcal{M}$

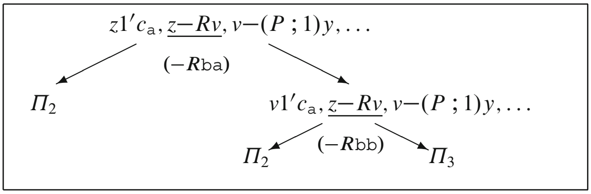

Fig. 7.5 The subtree Π_1

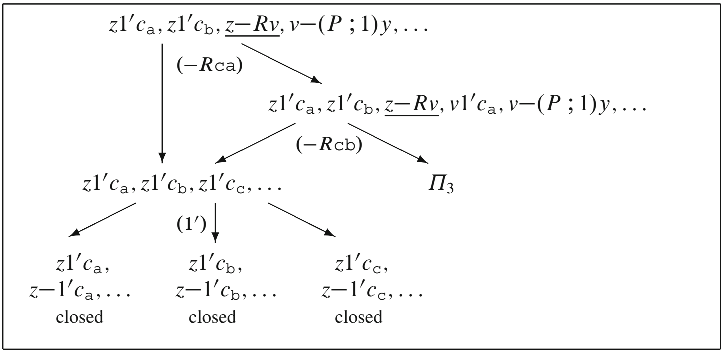

Fig. 7.6 The subtree Π_2

$\mathsf{RL_L}$, an $\mathsf{RL_L}$-model $\mathcal{M}' = (U', R', m')$, and a valuation v_a in $\mathcal{M}'$ such that for all object variables x and y, the problem '$\mathcal{M}, \mathtt{a} \models \varphi$?' is equivalent to the problem '$\mathcal{M}', v_{\mathtt{a}} \models x\tau(\varphi)y$?'. Then, we apply the method of verification of satisfaction as presented in Sect. 3.5 to the relational logic $\mathsf{RL_L}$, the model $\mathcal{M}'$, the formula $x\tau(\varphi)y$, and elements $v_{\mathtt{a}}(x)$ and $v_{\mathtt{a}}(y)$ of U'. We construct an instance $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$ of the logic $\mathsf{RL_L}$ and we obtain:

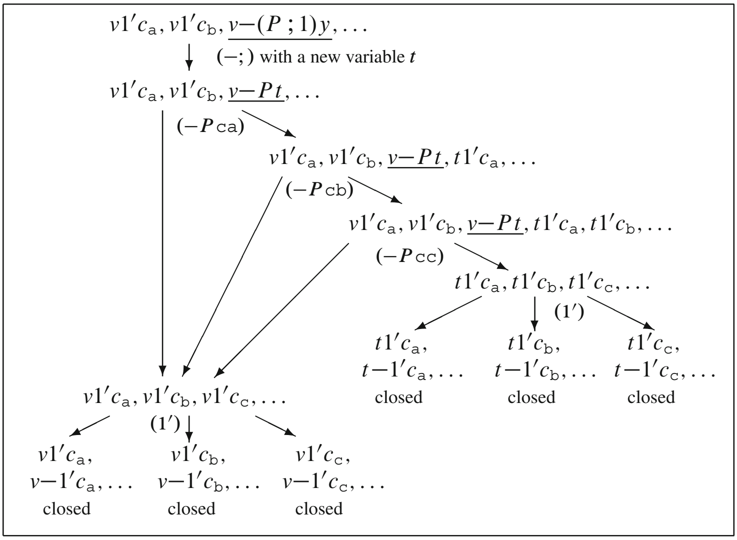

Fig. 7.7 The subtree Π_3

Theorem 7.8.1 (Relational Satisfaction in Standard Modal Logics). *For every* L*-formula* φ*, for every finite* L*-model* $\mathcal{M} = (U, R, m)$*, and for every state* $\mathtt{a} \in U$*, the following statements are equivalent:*

1. $\mathcal{M}, \mathtt{a} \models \varphi$*;*
2. $c_{v_{\mathtt{a}}(x)}\tau(\varphi)v_{\mathtt{a}}(y)$ *is* $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$*-provable.*

As an example of an application of the method, consider the modal logic K. Let $\mathcal{M} = (U, R, m)$ be a K-model such that $U = \{\mathtt{a}, \mathtt{b}\}$, $m(p) = \{\mathtt{a}\}$ and the accessibility relation is $R = \{(\mathtt{b}, \mathtt{a})\}$. Let φ be the formula $\langle R\rangle p$. Let us consider the problem: 'is φ satisfied in $\mathcal{M}$ by state $\mathtt{b}$?' The translation of the formula φ is $\tau(\varphi) = (R\,;(P\,;1))$, where $\tau'(p) = P$. By Proposition 7.4.2, there exist a standard $\mathsf{RL_K}$-model $\mathcal{M}'$ and a valuation $v_{\mathtt{b}}$ in $\mathcal{M}'$ such that for all object variables x and y, $\mathcal{M}, \mathtt{b} \models \varphi$ iff $\mathcal{M}', v_{\mathtt{b}} \models x\tau(\varphi)y$.

The $\mathsf{RL_K}$-model $\mathcal{M}' = (U', R', m')$ is such that $U' = U$, $R' = R$, and m' is the meaning function satisfying:

$$m'(P) = \{(\mathtt{a}, \mathtt{a}), (\mathtt{a}, \mathtt{b})\}.$$

Let $v_{\mathtt{b}}$ be a valuation such that $v_{\mathtt{b}}(x) = b$ and $v_{\mathtt{b}}(y) = a$. Then $\mathcal{M}'$ and $v_{\mathtt{b}}$ satisfy the condition: $\mathcal{M}, \mathtt{b} \models \varphi$ iff $\mathcal{M}', v_{\mathtt{b}} \models x\tau(\varphi)y$.

Therefore the satisfaction problem 'is φ satisfied in $\mathcal{M}$ by state $\mathtt{b}$?' is equivalent to the problem '$(\mathtt{b}, \mathtt{a}) \in m'(\tau(\varphi))$?'. By Proposition 3.5.1 this is equivalent to $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$-provability of $c_{\mathtt{b}}\tau(\varphi)c_{\mathtt{a}}$.

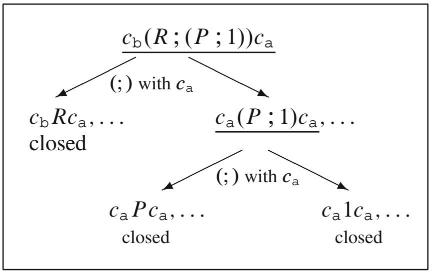

Fig. 7.8 An $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$-proof showing that $\langle R\rangle p$ is satisfied in $\mathcal{M}$ by state $\mathtt{b}$

$\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$-dual tableau contains the rules and axiomatic sets of $\mathsf{RL}(1,1')$-dual tableau adjusted to $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$-language and the rules and axiomatic sets specific for the satisfaction problem as presented in Sect. 3.5:

- The rules are: $(-R\mathtt{aa})$, $(-R\mathtt{ab})$, $(-R\mathtt{bb})$, $(-P\mathtt{bb})$, $(-P\mathtt{ba})$, $(1')$, and $(\mathtt{a} \neq \mathtt{b})$;
- The axiomatic sets are those that include either of the following subsets: $\{c_{\mathtt{b}}Rc_{\mathtt{a}}\}$, $\{c_{\mathtt{a}}Pc_{\mathtt{a}}\}$, $\{c_{\mathtt{a}}Pc_{\mathtt{b}}\}$, $\{c_{\mathtt{a}}{-R}c_{\mathtt{a}}\}$, $\{c_{\mathtt{b}}{-R}c_{\mathtt{b}}\}$, $\{c_{\mathtt{a}}{-R}c_{\mathtt{b}}\}$, $\{c_{\mathtt{b}}{-P}c_{\mathtt{b}}\}$, and $\{c_{\mathtt{b}}{-P}c_{\mathtt{a}}\}$.

It is easy to see that the formula $x\tau(\varphi)y$ is satisfied in $\mathcal{M}'$ by valuation $v_{\mathtt{b}}$, though it is not true in $\mathcal{M}'$. Therefore, the formula $\langle R\rangle p$ is satisfied in $\mathcal{M}$ by state $\mathtt{b}$, while it is not true in $\mathcal{M}$. Figure 7.8 presents an $\mathsf{RL}_{\mathcal{M}',x\tau(\varphi)y}$-proof of $c_{\mathtt{b}}\tau(\varphi)c_{\mathtt{a}}$, which shows that the formula $\langle R\rangle p$ is satisfied in $\mathcal{M}$ by state $\mathtt{b}$.