

Chapter 2

Dual Tableaux for Logics of Classical Algebras of Binary Relations

2.1 Introduction

The first steps in developing a mathematical theory of binary relations were taken by Augustus De Morgan in 1864 [Mor64]. His work was based on an earlier development by George Boole [Boo47] of a calculus of sets understood as an algebra of logic. Next, the theory was extensively developed by Charles Sanders Peirce [Pei83] and Ernst Schröder [Sch91]. This early theory studied binary relations between elements of a set and their properties. The research was mainly concerned with the arithmetic of binary relations and the relational notions that are expressible as equations in the calculus of relations. Alfred Tarski [Tar41] proposed a modern reformulation of the calculus of binary relations as a theory of abstract algebras called relation algebras. A relation algebra is a kind of a join of a Boolean algebra and an involuted monoid. The monoid operation of product is an abstract counterpart of the composition of two relations, the involution models the operation of forming the converse of a relation, and the neutral element of the product corresponds to the identity relation. An extensive study of the theory of relation algebras and the recent developments can be found in [TG87, HH02, Mad06], among others.

Many theories from a variety of fields, in particular from computer science and logic, can be interpreted as theories of algebras of relations. The fundamental relational structure consisting of a Boolean algebra together with a monoid constitutes a common core of a great variety of these theories. In Parts II, III, IV, and V of this book the relational interpretability of theories is a basis for the construction of dual tableau systems for them in a systematic modular way.

In this chapter we present a basic logic of binary relations which provides a framework for the relational interpretation of the theories considered in the book. The logic can be viewed as a generic logic for the development of dual tableaux for these theories. The rules of the dual tableau of the basic relational logic are included in all the proof systems presented in the chapters of the parts of the book listed above. For each particular theory we need, first, to expand the basic relational logic with specific relational constants and/or operations satisfying the appropriate axioms which enable us to prove relational interpretability of the given theory. Next, a dual tableau of the basic relational logic is extended with the rules characterizing

those specific constants and operations. The method of proving correctness and completeness of dual tableaux presented in this book follows the major steps of the corresponding proofs for the dual tableau of the basic relational logic. In this way, the developments of this chapter establish the tools for relational interpretability of theories and for construction of their relational proof systems.

Some other proof systems for relation algebras are presented in [Wad75, Hen80, Sch82, Mad83, Gor95, Gor97, Gor01].

2.2 Algebras of Binary Relations

The *full algebra of binary relations on a set U* is a structure of the form:

$$\mathfrak{R}(U) = (\mathcal{P}(U \times U), \cup, \cap, -, \emptyset, U \times U, ;, ^{-1}, Id),$$

where $(\mathcal{P}(U \times U), \cup, \cap, -, \emptyset, U \times U)$ is the Boolean algebra of all the subsets of $U \times U$, ‘;’ is the relative product of relations defined for all relations R, S on U as $R;S = \{(x, y) \in U \times U : \exists z \in U (xRz \wedge zSy)\}$, ‘ $^{-1}$ ’ is the converse of relations defined as $R^{-1} = \{(x, y) \in U \times U : yRx\}$, and Id is the identity on U , i.e., $Id = \{(x, x) : x \in U\}$.

A *proper algebra of binary relations* is an algebra $(W, \cup, \cap, -, \emptyset, 1, ;, ^{-1}, 1')$ whose elements are binary relations, $\cup, \cap, ;, ^{-1}, \emptyset$ are as above, 1 is an equivalence relation and the largest element of W , $1'$ is the identity on the field of 1 , and the complement operation ‘ $-$ ’ is defined with respect to 1 .

A generalization of algebras of binary relations is provided by the notion of relation algebra [Tar41].

A *relation algebra* is a structure of the form:

$$\mathfrak{A} = (W, +, \cdot, -, 0, 1, ;, \tilde{}, 1'), \text{ where :}$$

- $(W, +, \cdot, -, 0, 1)$ is a Boolean algebra,
- $;$ is an associative binary operation that distributes over $+$, i.e., $(x + y);z = x;z + y;z$ and $x;(y + z) = x;y + x;z$,
- $\tilde{}$ is a unary operation that distributes over $+$, i.e., $(x + y)^{\sim} = x^{\sim} + y^{\sim}$, and satisfies $(x^{\sim})^{\sim} = x$ and $(x;y)^{\sim} = y^{\sim};x^{\sim}$, for all $x, y \in W$,
- $1';x = x;1' = x$, for every $x \in W$,
- $x^{\sim};(-x;y) \leq -y$, for all $x, y, z \in W$.

The class of relation algebras is denoted by **RA**. Clearly, every algebra of binary relations is a relation algebra. The similarity type of relation algebras usually contains the three constants $0, 1$, and $1'$. However, 0 and 1 can be defined as $0 \stackrel{\text{df}}{=} 1' \cdot (-1')$ and $1 \stackrel{\text{df}}{=} 1' + (-1')$. We may also define $0' \stackrel{\text{df}}{=} -1'$. The relation $0'$ is called the *diversity* relation.

A relation algebra is said to be *representable* whenever it is isomorphic to a proper algebra of binary relations. The class of all representable relation algebras is

denoted by **RRA**. There exist relation algebras \mathfrak{A} (both finite and infinite) such that $\mathfrak{A} \in \mathbf{RA}$ and $\mathfrak{A} \notin \mathbf{RRA}$. Such algebras can be found in [Lyn50]. Some representation theorems, where each algebra from the class **RA** is embeddable into an algebra which is not necessarily an algebra of binary relations, can be found in [Dun01a]. In that paper each element of a relation algebra is mapped into a set of binary relations, not necessarily to a single relation. It is known that **RRA** is not finitely axiomatizable. Moreover, an infinite axiomatization of **RRA** requires infinitely many relation variables [Mon64]. **RRA** is a variety with a recursively enumerable but undecidable equational theory.

A relation R on U is said to be a *right* (resp. *left*) *ideal* relation whenever $R; 1 = R$ (resp. $1; R = R$). Right ideal relations play an important role in relational formalization of logics discussed in Parts III, IV, and V of the book. Various properties of right ideal relations are presented in the relevant chapters, see e.g., Propositions 7.4.1 and 16.4.1.

Proposition 2.2.1. *For all relations $R, S \in \mathfrak{R}(U)$, the following conditions are satisfied:*

1. $R \subseteq S$ iff $\neg R \cup S = 1$,
2. $R = S$ iff $(\neg R \cup S) \cap (\neg S \cup R) = 1$,
3. $R \neq \emptyset$ iff $1; R; 1 = 1$,
4. $R \neq 1$ iff $1; (\neg R); 1 = 1$,
5. $R = 1$ and $S = 1$ iff $R \cap S = 1$,
6. $R = 1$ or $S = 1$ iff $1; \neg(1; (\neg R); 1); 1 \cup S = 1$,
7. $R = 1$ implies $S = 1$ iff $(1; (\neg R); 1) \cup S = 1$.

Some arithmetic laws of relation algebras are collected in the following theorem:

Theorem 2.2.1. *The following are true in all relation algebras:*

1. $r; 0 = 0; r = 0$,
2. $1^\sim = 1$,
3. $1; 1 = 1$,
4. If $r \leq 1'$, then $r^\sim = r$,
5. If $r, s \leq 1'$, then $r; s = r \cdot s$,
6. If $r \leq 1'$, then $(r; 1) \cdot s = r; s$ and $(1; r) \cdot s = s; r$,
7. $(\neg r)^\sim = \neg(r^\sim)$,
8. $(r \cdot s); t \leq (r; t) \cdot (s; t)$ and $r; (s \cdot t) \leq (r; s) \cdot (r; t)$,
9. If $t \leq 1'$, then $(t; r) \cdot s = t; (r \cdot s)$ and $(r; t) \cdot s = (r \cdot s); t$,
10. If s is right ideal, then $r \cdot s = (s \cdot 1')$; r ,
11. If s is left ideal, then $r \cdot s = r; (s \cdot 1')$.

Example. We present two examples of relation algebras quoted from [Mad91]. The full algebra of binary relations on a three element set $\{0, 1, 2\}$, $\mathfrak{R}(\{0, 1, 2\})$, has 512 elements and 9 atoms. Its atoms are $\{(0, 0)\}$, $\{(0, 1)\}$, $\{(0, 2)\}$, $\{(1, 0)\}$, $\{(1, 1)\}$, $\{(1, 2)\}$, $\{(2, 0)\}$, $\{(2, 1)\}$, and $\{(2, 2)\}$.

Table 2.1 Compositions of atoms of the pentagonal algebra

	$1'$	a	b
$1'$	$1'$	a	b
a	a	$1' + b$	$a + b$
b	b	$a + b$	$1' + a$

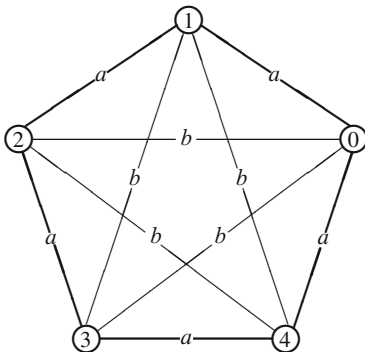


Fig. 2.1 Representation of the pentagonal algebra

The pentagonal relation algebra \mathfrak{B} has three atoms $1'$, a , and b . The elements of \mathfrak{B} are $0, 1', a, b, 1' + b = -a, 1' + a = -b, a + b = 0'$, and $1 = 1' + a + b$. The compositions of atoms are given in Table 2.1. This table determines all the compositions of elements of \mathfrak{B} , since every element is a join of atoms and the composition ; distributees over +.

The algebra \mathfrak{B} is representable, that is it can be embedded in some $\mathfrak{R}(U)$. It turns out that this is possible only if $card(U) = 5$. A copy of \mathfrak{B} inside $\mathfrak{R}(\{0, \dots, 4\})$ can be defined as follows:

$$\begin{aligned}
 1' &= \{(0, 0), (1, 1), (2, 2), (3, 3), (4, 4)\}; \\
 a &= \{(0, 1), (1, 0), (1, 2), (2, 1), (2, 3), (3, 2), (3, 4), (4, 3), (4, 0), (0, 4)\}; \\
 b &= \{(0, 2), (2, 0), (1, 3), (3, 1), (2, 4), (4, 2), (3, 0), (0, 3), (4, 1), (1, 4)\}.
 \end{aligned}$$

A picture of this representation in Fig. 2.1 shows a as a pentagon and b as a pentagram.

2.3 Logics of Binary Relations

In this chapter we present a general form of logics of binary relations considered in the book. There are two kinds of expressions of relational languages: terms and formulas. Terms represent relations and formulas express the facts that a pair of objects belongs to a relation.

Most of the relational logics of binary relations considered in the book are defined according to the following scheme. The vocabulary of the language of a relational logic L consists of the symbols from the following pairwise disjoint sets:

- $\mathbb{O}\mathbb{V}_L$ – a countable infinite set of object variables;
- $\mathbb{O}\mathbb{C}_L$ – a countable (possibly empty) set of object constants;
- $\mathbb{R}\mathbb{V}_L$ – a countable infinite set of binary relational variables;
- $\mathbb{R}\mathbb{C}_L$ – a countable (possibly empty) set of binary relational constants;
- A set of relational operations including the standard operations $-, \cup, \cap, ;,$ and $^{-1}$.

Object constants are needed for relational representation of theories in Chaps. 3, 15, and Sect. 16.5. Relational representations of all the theories of Parts III, IV, and V require specific relational constants.

The set $\mathbb{R}\mathbb{A}_L = \mathbb{R}\mathbb{V}_L \cup \mathbb{R}\mathbb{C}_L$ is called the *set of atomic relational terms*. The set $\mathbb{O}\mathbb{S}_L = \mathbb{O}\mathbb{V}_L \cup \mathbb{O}\mathbb{C}_L$ is called the *set of object symbols*. The set $\mathbb{R}\mathbb{T}_L$ of *relational terms* is the smallest (with respect to inclusion) set of expressions that includes all the atomic relational terms and is closed with respect to all the relational operations. *L-formulas* are of the form xRy , where $x, y \in \mathbb{O}\mathbb{S}_L$ and $R \in \mathbb{R}\mathbb{T}_L$. An *L-formula* xRy is said to be *atomic* whenever $R \in \mathbb{R}\mathbb{A}_L$.

With an *L-language* a class of *L-structures* is associated. An *L-structure* has the form $\mathcal{M} = (U, m)$, where U is a non-empty set and m is a meaning function which assigns:

- (m1) Elements of U to object constants, that is $m(c) \in U$, for every $c \in \mathbb{O}\mathbb{C}_L$;
- (m2) Binary relations on U to atomic relational terms, that is $m(R) \subseteq U \times U$, for every $R \in \mathbb{R}\mathbb{A}_L$.

m extends to all the compound relational terms, in particular:

- (m3) $m(-R) = (U \times U) \setminus m(R)$;
- (m4) $m(R \cup S) = m(R) \cup m(S)$;
- (m5) $m(R \cap S) = m(R) \cap m(S)$;
- (m6) $m(R^{-1}) = (m(R))^{-1} = \{(x, y) \in U \times U : (y, x) \in m(R)\}$;
- (m7) $m(R; S) = m(R); m(S) = \{(x, y) \in U \times U : \exists z \in U, (x, z) \in m(R) \text{ and } (z, y) \in m(S)\}$.

So, for *L-structures* we only require that object constants are interpreted as elements of their universes, relational constants are interpreted as binary relations, and standard relational operations receive their usual meaning.

L-models are *L-structures*, where meaning function m satisfies the conditions specific for the given logic and/or interpretation of specific relational and/or object constants of the logic. Relational logics considered in the book are always defined in terms of their languages and the classes of models.

In order to avoid a confusion between syntactic objects in the languages of theories and semantic constraints on relations in the models of the theories, we write

xRy for the former, where x, y are object symbols and R is a relational term of a language, and $(x, y) \in R$ for the latter, where x and y are objects from the universe of a model and R is a relation in the model.

Let $\mathcal{M} = (U, m)$ be an \mathbf{L} -structure. A *valuation* in \mathcal{M} is any function $v: \mathbb{O}\mathbb{S}_{\mathbf{L}} \rightarrow U$ such that $v(c) = m(c)$, for every $c \in \mathbb{O}\mathbb{C}_{\mathbf{L}}$. *Satisfaction* of an \mathbf{L} -formula xRy by a valuation v in an \mathbf{L} -structure \mathcal{M} is defined as:

- $\mathcal{M}, v \models xRy$ iff $(v(x), v(y)) \in m(R)$.

An \mathbf{L} -formula xRy is *true* in \mathcal{M} whenever it is satisfied by all the valuations in \mathcal{M} . If \mathcal{K} is a class of \mathbf{L} -structures, then an \mathbf{L} -formula xRy is said to be \mathcal{K} -*valid* whenever it is true in every structure of \mathcal{K} , and it is \mathbf{L} -*valid* whenever it is true in all \mathbf{L} -models.

Fact 2.3.1. *Let \mathbf{L} and \mathbf{L}' be relational logics such that every \mathbf{L} -structure is an \mathbf{L}' -structure. Then for any relational formula xRy , if xRy is \mathbf{L}' -valid, then it is \mathbf{L} -valid.*

2.4 Relational Dual Tableaux

Relational dual tableaux are founded on the Rasiowa–Sikorski system for the first-order logic presented in Chap. 1. They are powerful tools for performing the four major reasoning tasks: verification of validity, verification of entailment, model checking, and verification of satisfaction. Relational proof systems are determined by their axiomatic sets of formulas and rules which most often apply to finite sets of relational formulas. Some relational proof systems with infinitary rules are known in the literature (see Sect. 19.2), but in the present chapter we confine ourselves to finitary rules only. The axiomatic sets take the place of axioms. The rules are intended to reflect properties of relational operations and constants. There are two groups of rules: decomposition rules and specific rules. Given a formula, the decomposition rules of the system enable us to transform it into simpler formulas, or the specific rules enable us to replace a formula by some other formulas. The rules have the following general form:

$$\text{(rule)} \quad \frac{\Phi(\bar{x})}{\Phi_1(\bar{x}_1, \bar{u}_1, \bar{w}_1) \mid \dots \mid \Phi_n(\bar{x}_n, \bar{u}_n, \bar{w}_n)}$$

where $\Phi(\bar{x})$ is a finite (possibly empty) set of formulas whose object symbols are among the elements of $\text{set}(\bar{x})$, where \bar{x} is a finite sequence of object symbols and $\text{set}(\bar{x})$ is a set of elements of sequence \bar{x} ; every $\Phi_j(\bar{x}_j, \bar{u}_j, \bar{w}_j)$, $1 \leq j \leq n$, is a finite non-empty set of formulas, whose object symbols are among the elements of $\text{set}(\bar{x}_j) \cup \text{set}(\bar{u}_j) \cup \text{set}(\bar{w}_j)$, where $\bar{x}_j, \bar{u}_j, \bar{w}_j$ are finite sequences of object symbols such that $\text{set}(\bar{x}_j) \subseteq \text{set}(\bar{x})$, $\text{set}(\bar{u}_j)$ consists of the variables that may be instantiated to arbitrary object symbols when the rule is applied (usually to the object symbols

that appear in the set to which the rule is being applied), $\text{set}(\bar{w}_j)$ consists of the variables that must be instantiated to pairwise distinct new variables (not appearing in the set to which the rule is being applied) and distinct from any variable of sequence \bar{u}_j . A rule of the form (rule) is the n -fold branching rule, where the j th successor of $\Phi(\bar{x})$ is the set $\Phi_j(\bar{x}_j, \bar{u}_j, \bar{w}_j)$. A rule of the form (rule) is *applicable* to a finite set X of formulas whenever $\Phi(\bar{x}) \subseteq X$. As a result of an application of a rule of the form (rule) to set X , we obtain the sets $(X \setminus \Phi(\bar{x})) \cup \Phi_j(\bar{x}_j, \bar{u}_j, \bar{w}_j)$, $j \in \{1, \dots, n\}$. A set to which a rule has been applied is called the *premise* of the rule, and the sets obtained by an application of the rule are called its *conclusions*.

Let L be a relational logic. A finite set $\{\varphi_1, \dots, \varphi_n\}$ of L -formulas is said to be an L -set whenever for every L -model \mathcal{M} and for every valuation v in \mathcal{M} there exists $i \in \{1, \dots, n\}$ such that φ_i is satisfied by v in \mathcal{M} . It follows that the first-order disjunction of all the formulas from an L -set is valid in first-order logic. A rule of the form (rule) is L -correct whenever for every finite set X of L -formulas, $X \cup \Phi(\bar{x})$ is an L -set if and only if $X \cup \Phi_j(\bar{x}_j, \bar{u}_j, \bar{w}_j)$ is an L -set, for every $j \in \{1, \dots, n\}$, i.e., the rule preserves and reflects validity. If \mathcal{K} is a class of L -structures, then we define the notion of a \mathcal{K} -set and the notion of \mathcal{K} -correctness in a similar way. A finite set of L -formulas $\{\varphi_1, \dots, \varphi_n\}$ is said to be a \mathcal{K} -set whenever for every \mathcal{K} -structure \mathcal{M} and for every valuation v in \mathcal{M} there exists $i \in \{1, \dots, n\}$ such that φ_i is satisfied by v in \mathcal{M} . A rule of the form (rule) is \mathcal{K} -correct whenever for every finite set X of L -formulas, $X \cup \Phi(\bar{x})$ is a \mathcal{K} -set if and only if $X \cup \Phi_j(\bar{x}_j, \bar{u}_j, \bar{w}_j)$ is a \mathcal{K} -set, for every $j \in \{1, \dots, n\}$.

Let xRy be an L -formula. The notion of an L -proof tree is defined in a similar way as in logic F in Sect. 1.3, that is an L -proof tree for xRy is a tree with the following properties:

- The formula xRy is at the root of the tree;
- Each node except the root is obtained by an application of an L -rule to its predecessor node;
- A node does not have successors whenever its set of formulas is an L -axiomatic set or none of the rules is applicable to its set of formulas.

Similarly as in the dual tableau for logic F , a branch of an L -proof tree is said to be *closed* whenever it contains a node with an L -axiomatic set of formulas. A tree is *closed* whenever all of its branches are closed. An L -formula xRy is L -provable whenever there is a closed L -proof tree for it which is then referred to as its L -proof.

2.5 A Basic Relational Logic

The logic RL presented in this section is a common core of all the logics of binary relations presented in this book. RL represents, in fact, a family of logics which possibly differ in the object constants admitted in their languages. We do not assume here any specific properties of these constants, so all the developments of this section

are relevant for any such logic. Throughout the book, by **RL**-logic we shall mean any logic from this family. The language of **RL**-logic is defined as in Sect. 2.3 with:

- $\mathbb{RC}_{\text{RL}} = \emptyset$;
- The relational operations are $\neg, \cup, \cap, ;, ^{-1}$.

RL-models coincide with the **RL**-structures defined in Sect. 2.3 adjusted to the **RL**-language.

An **RL**-dual tableau consists of decomposition rules and axiomatic sets. Decomposition rules have the following forms:

For any object symbols x and y and for any relational terms R and S ,

$$\begin{array}{ll}
 (\cup) & \frac{x(R \cup S)y}{xRy, xSy} & (-\cup) & \frac{x-(R \cup S)y}{x-Ry \mid x-Sy} \\
 (\cap) & \frac{x(R \cap S)y}{xRy \mid xSy} & (-\cap) & \frac{x-(R \cap S)y}{x-Ry, x-Sy} \\
 (-) & \frac{x--Ry}{xRy} & & \\
 (-1) & \frac{xR^{-1}y}{yRx} & (-^{-1}) & \frac{x-R^{-1}y}{y-Rx} \\
 (;) & \frac{x(R; S)y}{xRz, x(R; S)y \mid zSy, x(R; S)y} & & z \text{ is any object symbol} \\
 (-;) & \frac{x-(R; S)y}{x-Rz, z-Sy} & & z \text{ is a new object variable}
 \end{array}$$

A set of **RL**-formulas is said to be an **RL**-axiomatic set whenever it includes a subset of the following form:

(Ax) $\{xRy, x-Ry\}$, where x, y are object symbols and R is a relational term.

Most of relational logics of binary relations studied in the book are obtained from **RL** by postulating some constraints on their object and/or relational constants. Dual tableaux for these logics include the decomposition rules of **RL**-dual tableau, where in each particular logic **L** the terms and object symbols range over the corresponding entities of the language of **L**. Specific rules reflect the properties of constants assumed in the **L**-language in question. In all relational dual tableaux considered in the book the sets including a subset of the form (Ax) are assumed to be **L**-axiomatic sets.

Proposition 2.5.1.

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

Proof. By way of example, we prove correctness of the rules (;) and (-;). Let X be a finite set of **RL**-formulas.

(;) Clearly, if $X \cup \{x(R; S)y\}$ is an RL-set, then so are $X \cup \{xRz, x(R; S)y\}$, $X \cup \{zSy, x(R; S)y\}$. Assume $X \cup \{xRz, x(R; S)y\}$ and $X \cup \{zSy, x(R; S)y\}$ are RL-sets, and suppose $X \cup \{x(R; S)y\}$ is not an RL-set. Then there exist an RL-model $\mathcal{M} = (U, m)$ and a valuation v in \mathcal{M} such that for every $\varphi \in X \cup \{x(R; S)y\}$, $\mathcal{M}, v \not\models \varphi$. It follows that for every $a \in U$, $(v(x), a) \notin m(R)$ or $(a, v(y)) \notin m(S)$. However, by the assumption, model \mathcal{M} and valuation v satisfy $(v(x), v(z)) \in m(R)$ and $(v(z), v(y)) \in m(S)$, a contradiction.

(-;) Assume $X \cup \{x-(R; S)y\}$ is an RL-set. Suppose $X \cup \{x-Rz, z-Sy\}$ is not an RL-set, where z does not occur in X and $z \neq x, y$. Then there exist an RL-model $\mathcal{M} = (U, m)$ and a valuation v in \mathcal{M} such that for every $\varphi \in X \cup \{x-Rz, z-Sy\}$, $\mathcal{M}, v \not\models \varphi$. Thus, $(v(x), v(z)) \in m(R)$ and $(v(z), v(y)) \in m(S)$. However, by the assumption, for every $a \in U$, $(v(x), a) \notin m(R)$ or $(a, v(y)) \notin m(S)$, a contradiction. Now, assume $X \cup \{x-Rz, z-Sy\}$ is an RL-set. Then, by the assumption on variable z , for every RL-model $\mathcal{M} = (U, m)$ and for every valuation v in \mathcal{M} , either there exists $\varphi \in X$ such that $\mathcal{M}, v \models \varphi$ or for every $a \in U$, either $(v(x), a) \notin m(R)$ or $(a, v(y)) \notin m(S)$. That is for every RL-model $\mathcal{M} = (U, m)$ and for every valuation v in \mathcal{M} either there exists $\varphi \in X$ such that $\mathcal{M}, v \models \varphi$ or $\mathcal{M}, v \models x-(R; S)y$. Hence, $X \cup \{x-(R; S)y\}$ is an RL-set. \square

Due to the above proposition, we obtain:

Proposition 2.5.2. *Let φ be an RL-formula. If φ is RL-provable, then it is RL-valid.*

A branch b of an RL-proof tree is said to be *complete* whenever it is closed or it satisfies the following RL-completion conditions:

For all object symbols x and y and for all relational terms R and S ,

Cpl(\cup) (resp. Cpl($-\cap$)) If $x(R \cup S)y \in b$ (resp. $x-(R \cap S)y \in b$), then both $xRy \in b$ (resp. $x-Ry \in b$) and $xSy \in b$ (resp. $x-Sy \in b$), obtained by an application of the rule (\cup) (resp. ($-\cap$));

Cpl(\cap) (resp. Cpl($-\cup$)) If $x(R \cap S)y \in b$ (resp. $x-(R \cup S)y \in b$), then either $xRy \in b$ (resp. $x-Ry \in b$) or $xSy \in b$ (resp. $x-Sy \in b$), obtained by an application of the rule (\cap) (resp. ($-\cup$));

Cpl($-$) If $x(-R)y \in b$, then $xRy \in b$, obtained by an application of the rule ($-$);

Cpl($^{-1}$) If $xR^{-1}y \in b$, then $yRx \in b$, obtained by an application of the rule ($^{-1}$);

Cpl($^{-1}$) If $x-R^{-1}y \in b$, then $y-Rx \in b$, obtained by an application of the rule ($^{-1}$);

Cpl(;) If $x(R; S)y \in b$, then for every object symbol z , either $xRz \in b$ or $zSy \in b$, obtained by an application of the rule (;);

Cpl(-;) If $x-(R; S)y \in b$, then for some object variable z , both $x-Rz \in b$ and $z-Sy \in b$, obtained by an application of the rule (-;).

An RL-proof tree is said to be *complete* if and only if all of its branches are complete. A complete non-closed branch of an RL-proof tree is said to be *open*. Note that every RL-proof tree can be extended to a complete RL-proof tree, i.e., for every RL-formula φ there exists a complete RL-proof tree for φ .

Due to the forms of RL-decomposition rules, Fact 1.3.1 and Proposition 1.3.3 transfer to logic RL.

Fact 2.5.1. *If a node of an RL-proof tree contains an RL-formula xRy or $x-Ry$, for an atomic R , then all of its successors contain this formula as well.*

This property will be referred to as preservation of formulas built with atomic terms or their complements.

Proposition 2.5.3. *For every branch b of an RL-proof tree and for every atomic term R , if $xRy \in b$ and $x-Ry \in b$, then b is closed.*

We can prove a stronger form of Proposition 2.5.3:

Proposition 2.5.4. *Let b be a complete branch of an RL-proof tree. If there is a relational term R such that $xRy \in b$ and $x-Ry \in b$, then b is closed.*

Proof. Clearly, if branch b is closed, then the proposition holds. Assume that branch b is open. The proof of the proposition is by induction on the complexity of relational terms. For atomic relational terms the statement holds due to Proposition 2.5.3. By way of example, we prove the statement for terms of the form $R; S$. Suppose $x(R; S)y$ and $x-(R; S)y$ belong to b , for some relational terms R and S , and some object symbols x and y . Since $x-(R; S)y \in b$, by the completion condition $\text{Cpl}(-;)$, both $x-Rz \in b$ and $z-Sy \in b$, for some object variable z . Furthermore, since $x(R; S)y \in b$, by the completion condition $\text{Cpl}(;)$, for every object variable z , either $xRz \in b$ or $zSy \in b$. Thus either both xRz and $x-Rz$ belong to b or both zSy and $z-Sy$ belong to b . Hence, by the induction hypothesis, b is closed. \square

In order to prove completeness of RL-dual tableau, first, we construct a branch structure $\mathcal{M}^b = (U^b, m^b)$ determined by an open branch b of a complete RL-proof tree as follows:

- $U^b = \mathbb{O}\mathbb{S}_{\text{RL}}$;
- $m^b(c) = c$, for every $c \in \mathbb{O}\mathbb{C}_{\text{RL}}$;
- $m^b(R) = \{(x, y) \in U^b \times U^b : xRy \notin b\}$, for every relational variable R ;
- m^b extends to all the compound relational terms as in the RL-models.

Directly from this definition, we obtain:

Fact 2.5.2. *For every open branch b of an RL-proof tree, \mathcal{M}^b is an RL-model.*

Any structure \mathcal{M}^b is referred to as an RL-branch model. Let $v^b: \mathbb{O}\mathbb{S}_{\text{RL}} \rightarrow U^b$ be a valuation in \mathcal{M}^b such that $v^b(x) = x$, for every $x \in \mathbb{O}\mathbb{S}_{\text{RL}}$.

Proposition 2.5.5. *For every open branch b of an RL-proof tree and for every RL-formula φ , if $\mathcal{M}^b, v^b \models \varphi$, then $\varphi \notin b$.*

Proof. The proof is by induction on the complexity of formulas.

Let $\varphi = xRy$ be an atomic RL-formula. Assume that $\mathcal{M}^b, v^b \models xRy$, that is $(x, y) \in m^b(R)$. By the definition of the branch model $xRy \notin b$. Let $R \in \mathbb{R}\mathbb{V}_{\text{RL}}$ and $\mathcal{M}^b, v^b \models x-Ry$, that is $(x, y) \notin m^b(R)$. Therefore $xRy \in b$. Then, by Proposition 2.5.3, $x-Ry \notin b$, for otherwise b would be closed.

Let $\mathcal{M}^b, v^b \models x(S;T)y$. Then $(x, y) \in m^b(S;T)$, that is there exists an object symbol $z \in U^b$ such that $(x, z) \in m^b(S)$ and $(z, y) \in m^b(T)$. By the induction hypothesis, $xSz \notin b$ and $zTy \notin b$. Suppose $x(S;T)y \in b$. By the completion condition $\text{Cpl}(\cdot)$, for every object symbol $z \in U^b$, either $xSz \in b$ or $zTy \in b$, a contradiction.

Let $\mathcal{M}^b, v^b \models x-(S;T)y$. Then $(x, y) \notin m^b(S;T)$, that is for every object symbol $z \in U^b$, either $(x, z) \notin m^b(S)$ or $(z, y) \notin m^b(T)$. Suppose that $x-(S;T)y \in b$. By the completion condition $\text{Cpl}(-)$, for some object variable $z \in U^b$, both $x-Sz \in b$ and $z-Ty \in b$. By the induction hypothesis, $(x, z) \in m^b(S)$ and $(z, y) \in m^b(T)$, a contradiction.

The proofs of the remaining cases are similar. \square

Fact 2.5.2 and Proposition 2.5.5 enable us to prove:

Proposition 2.5.6. *Let φ be an RL-formula. If φ is RL-valid, then φ is RL-provable.*

Proof. Assume φ is RL-valid. Suppose there is no any closed RL-proof tree for φ . Then there exists a complete RL-proof tree for φ with an open branch, say b . Since $\varphi \in b$, by Proposition 2.5.5, φ is not satisfied by valuation v^b in the branch model \mathcal{M}^b . Hence φ is not RL-valid, a contradiction. \square

By Propositions 2.5.2 and 2.5.6, we have:

Theorem 2.5.1 (Soundness and Completeness of RL). *For every RL-formula φ , the following conditions are equivalent:*

1. φ is RL-valid;
2. φ is RL-provable.

Example. We show that $(R \cap S);T \subseteq (R;T) \cap (S;T)$ by proving the formula:

$$x(-((R \cap S);T) \cup ((R;T) \cap (S;T)))y.$$

Figure 2.2 presents its RL-proof.

2.6 A Method of Proving Soundness and Completeness of Relational Dual Tableaux

The method applied to proving soundness and completeness of RL-dual tableau determines a paradigm for all the soundness and completeness proofs presented in this book.

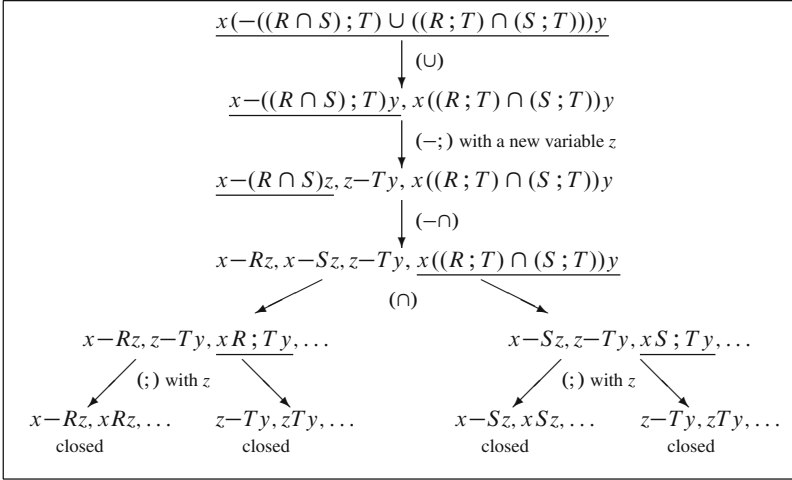


Fig. 2.2 An RL-proof of $(R \cap S); T \subseteq (R; T) \cap (S; T)$

Let L be a relational logic. In order to prove that an L -provable formula is L -valid it suffices to show that all the rules of an L -dual tableau are L -correct and all the axiomatic sets are L -sets. In some cases we also show correspondences between the semantic constraints posed on relational constants and/or relational operations and the rules and/or axiomatic sets reflecting those constraints. More precisely, we say that a rule (resp. an axiomatic set) of an L -dual tableau *reflects a constraint* assumed in L -models whenever for every class \mathcal{K} of L -structures, \mathcal{K} -correctness of the rule (resp. \mathcal{K} -validity of the axiomatic set) implies that every structure from \mathcal{K} satisfies the constraint.

In order to prove that an L -valid formula has an L -proof, we suppose that the formula does not have any L -proof. It follows that there exists a complete L -proof tree for this formula with an open branch b . Then we construct a branch structure $\mathcal{M}^b = (U^b, m^b)$ determined by b , where U^b consists of all the object symbols of L -language and m^b is the meaning function which assigns relations to atomic relational terms and extends homomorphically to all the terms as in RL -models. Most often the meaning function m^b of \mathcal{M}^b is defined by:

$$m^b(R) \stackrel{\text{df}}{=} \{(x, y) \in U^b \times U^b : xRy \notin b\}, \text{ for every atomic relational term } R.$$

In that case we will say that \mathcal{M}^b is *defined in a standard way*. We define the valuation v^b in the model \mathcal{M}^b as the identity function, i.e., $v^b(x) = x$ for every object symbol x .

Then the major steps of the proof are the propositions showing the following three properties:

- (1) **Closed Branch Property:** For any branch of an L -proof tree, if xRy and $x-Ry$, for an atomic term R , belong to the branch, then the branch can be closed.

- (2) **Branch Model Property:** Let \mathcal{M}^b be a branch structure determined by an open branch b of an L-proof tree. Then \mathcal{M}^b is an L-model.
- (3) **Satisfaction in Branch Model Property:** For every L-formula φ and for every open branch b of an L-proof tree for φ , the branch model \mathcal{M}^b and valuation v^b in \mathcal{M}^b satisfy:

$$\text{If } \mathcal{M}^b, v^b \models \varphi, \text{ then } \varphi \notin b.$$

Then we reason as in the proof of Proposition 2.5.6.

Roughly speaking, the method can be successfully applied provided that the following conditions will be satisfied: a sufficient condition for (1) is the preservation of formulas built with atomic terms or their complements by the applications of the rules; for (2), the L-dual tableau must have enough rules in order to reflect all the semantic properties of relational constants and relational operations of the L-language; for (3), the branch structure must provide a model that falsifies the non-provable formulas.

2.7 Relational Logic with Relations 1 and 1'

The logic considered in this section is obtained from RL-logic by expanding its language with constants 1 and 1'. The vocabulary of the language of RL(1, 1')-logic is defined as in Sect. 2.3 with:

- $\mathbb{R}\mathbb{C}_{\text{RL}(1,1')} = \{1, 1'\}$.

An RL(1, 1')-structure is an RL-model $\mathcal{M} = (U, m)$ such that $m(1)$ and $m(1')$ are binary relations on U . An RL(1, 1')-model is an RL(1, 1')-structure $\mathcal{M} = (U, m)$ such that the following conditions are satisfied:

- $m(1) = U \times U$;
- $m(1')$ is an equivalence relation on U ;
- $m(1') ; m(R) = m(R) ; m(1') = m(R)$, for every atomic relational term R (extensionality);
- m extends to all the compound relational terms as in RL-models.

Proposition 2.7.1. *Let $\mathcal{M} = (U, m)$ be an RL(1, 1')-model. Then, for every relational term R of RL(1, 1')-language, the extensionality property holds:*

$$m(1') ; m(R) = m(R) ; m(1') = m(R).$$

Proof. The proof is by induction on the complexity of relational terms. By way of example, we show the extensionality property for $R = -S$. Since $m(1')$ is reflexive, $m(-S) \subseteq m(1') ; m(-S)$. For \supseteq , assume that there exists $z \in U$ such that $(x, z) \in m(1')$ and $(z, y) \notin m(S)$. By the induction hypothesis, for every $u \in U$, $(z, u) \notin m(1')$ or $(u, y) \notin m(S)$. In particular, $(z, x) \notin m(1')$ or $(x, y) \notin m(S)$. Since $m(1')$ is symmetric, $(x, y) \in m(-S)$.

The equality $m(-S) = m(-S); m(1')$ can be proved in a similar way. The extensionality condition for the terms built with $\cup, \cap, ;$, and $^{-1}$ follows easily from the definition of the meaning function, properties of relational operations, and the induction hypothesis. \square

Proposition 2.7.2. *Let $\mathcal{M} = (U, m)$ be an $\text{RL}(1, 1')$ -structure satisfying the following conditions:*

1. $m(1) = U \times U$;
2. $m(1')$ is reflexive;
3. $m(1'); m(R) = m(R); m(1') = m(R)$, for every relational term R .

Then \mathcal{M} is an $\text{RL}(1, 1')$ -model.

Proof. Since $m(1')$ is reflexive, so is $m(1'^{-1})$. Thus $m(1') \subseteq m(1'^{-1}); m(1')$. By 3., $m(1'^{-1}); m(1') = m(1'^{-1})$. Therefore $m(1')$ is symmetric. Transitivity follows directly from 3. \square

It follows that an equivalent set of conditions characterizing the $\text{RL}(1, 1')$ -models could be reflexivity of $m(1')$ and the extensionality property for all the relational terms.

An $\text{RL}(1, 1')$ -model $\mathcal{M} = (U, m)$ is said to be *standard* whenever $m(1')$ is the identity on U , that is $m(1') = \{(x, x) : x \in U\}$.

Fact 2.7.1. *If a formula is $\text{RL}(1, 1')$ -valid, then it is true in all standard $\text{RL}(1, 1')$ -models.*

$\text{RL}(1, 1')$ -dual tableau includes the decomposition rules of RL -dual tableau (see Section 2.5) adjusted to $\text{RL}(1, 1')$ -language and the specific rules of the following forms:

For all object symbols x and y and for every atomic relational term R ,

$$(1'1) \quad \frac{xRy}{xRz, xRy \mid y1'z, xRy} \quad z \text{ is any object symbol}$$

$$(1'2) \quad \frac{xRy}{x1'z, xRy \mid zRy, xRy} \quad z \text{ is any object symbol}$$

The rule (1'1) reflects symmetry, transitivity, and the part $m(R); m(1') \subseteq m(R)$ of the extensionality property of $1'$. The rule (1'2) reflects the part $m(1'); m(R) \subseteq m(R)$ of extensionality property of $1'$. The formal presentation of these facts is given in Theorem 2.7.1.

A finite set of formulas is said to be $\text{RL}(1, 1')$ -axiomatic whenever it is an RL -axiomatic set adjusted to $\text{R}(1, 1')$ -language or it includes either of the sets of the following forms:

For all object symbols x and y ,

$$(\text{Ax1}) \quad \{x1'x\};$$

$$(\text{Ax2}) \quad \{x1y\}.$$

The axiomatic set $\{x1'x\}$ reflects reflexivity of $1'$.

Theorem 2.7.1 (Correspondence). *Let \mathcal{K} be a class of $\mathbf{RL}(1, 1')$ -structures. Then the following conditions are equivalent:*

1. \mathcal{K} is a class of $\mathbf{RL}(1, 1')$ -models;
2. The $\mathbf{RL}(1, 1')$ -axiomatic sets $(Ax1)$ and $(Ax2)$ are \mathcal{K} -sets and the rules $(1'1)$ and $(1'2)$ are \mathcal{K} -correct.

Proof. (1. \rightarrow 2.) Let \mathcal{K} be a class of $\mathbf{RL}(1, 1')$ -models. Since $m(1) = U \times U$, every superset of $\{x1y\}$ is an $\mathbf{RL}(1, 1')$ -set. Furthermore, since $m(1')$ is reflexive, $X \cup \{x1'x\}$ is a \mathcal{K} -set, for every finite set X of $\mathbf{RL}(1, 1')$ -formulas.

Now, we prove that the rule $(1'1)$ is \mathcal{K} -correct. Let X be a finite set of $\mathbf{RL}(1, 1')$ -formulas. It is easy to see that if $X \cup \{xRy\}$ is a \mathcal{K} -set, then so are $X \cup \{xRz, xRy\}$ and $X \cup \{y1'z, xRy\}$. Now, assume $X \cup \{xRz, xRy\}$ and $X \cup \{y1'z, xRy\}$ are \mathcal{K} -sets. Suppose $X \cup \{xRy\}$ is not a \mathcal{K} -set, that is there exist an $\mathbf{RL}(1, 1')$ -model \mathcal{M} in \mathcal{K} and a valuation v in \mathcal{M} such that for every formula $\varphi \in X \cup \{xRy\}$, $\mathcal{M}, v \models \varphi$, in particular $\mathcal{M}, v \models xRy$. Then, by the assumption and since $m(1')$ is symmetric, $\mathcal{M}, v \models xRz$ and $\mathcal{M}, v \models z1'y$, that is $(v(x), v(z)) \in m(R)$ and $(v(z), v(y)) \in m(1')$. Since $m(R); m(1') \subseteq m(R)$, $(v(x), v(y)) \in m(R)$. Hence, $\mathcal{M}, v \models xRy$, a contradiction. The proof for the rule $(1'2)$ is similar.

(2. \rightarrow 1.) Let \mathcal{K} be a class of $\mathbf{RL}(1, 1')$ -structures. We need to show that every \mathcal{K} -structure $\mathcal{M} = (U, m)$ is an $\mathbf{RL}(1, 1')$ -model, i.e., $m(1) = U \times U$, $m(1')$ is an equivalence relation on U , and the extensionality property is satisfied. Since every finite superset of $\{x1y\}$ is a \mathcal{K} -set, the formula $x1y$ is valid in every structure from \mathcal{K} . Hence, the first condition holds.

For reflexivity, observe that $\{x1'x\}$ is a \mathcal{K} -set. Thus, in every \mathcal{K} -structure \mathcal{M} , $\mathcal{M} \models x1'x$.

For symmetry, let $X \stackrel{\text{df}}{=} \{y-1'x\}$. Since the rule $(1'1)$ is \mathcal{K} -correct and the sets $X \cup \{x1'x, x1'y\}$ and $X \cup \{y1'x, x1'y\}$ are \mathcal{K} -sets, $X \cup \{x1'y\}$ is also a \mathcal{K} -set. Therefore, for every \mathcal{K} -structure $\mathcal{M} = (U, m)$ and for every valuation v in \mathcal{M} , if $(v(y), v(x)) \in m(1')$, then $(v(x), v(y)) \in m(1')$.

For transitivity, let $X \stackrel{\text{df}}{=} \{x-1'z, z-1'y\}$. Then, $X \cup \{x1'z, x1'y\}$ and $X \cup \{z1'y, x1'y\}$ are \mathcal{K} -sets. Thus, by \mathcal{K} -correctness of the rule $(1'2)$, $X \cup \{x1'y\}$ is also a \mathcal{K} -set. Therefore, for every \mathcal{K} -structure $\mathcal{M} = (U, m)$ and for every valuation v in \mathcal{M} , if $(v(x), v(z)) \in m(1')$ and $(v(z), v(y)) \in m(1')$, then $(v(x), v(y)) \in m(1')$.

Now, we prove the extensionality property, i.e., we show that for every atomic relational term R , $m(1'); m(R) = m(R); m(1') = m(R)$.

Since $m(1')$ is reflexive, $m(R) \subseteq m(1'); m(R)$ and $m(R) \subseteq m(R); m(1')$.

To show $m(1'); m(R) \subseteq m(R)$, consider $X \stackrel{\text{df}}{=} \{x-1'z, z-Ry\}$. By \mathcal{K} -correctness of the rule $(1'2)$, $X \cup \{xRy\}$ is a \mathcal{K} -set. Therefore, for every \mathcal{K} -structure $\mathcal{M} = (U, m)$ and for every valuation v in \mathcal{M} , if $(v(x), v(z)) \in m(1')$ and $(v(z), v(y)) \in m(R)$, then $(v(x), v(y)) \in m(R)$, which implies that if $(v(x), v(y)) \in m(1'); m(R)$, then $(v(x), v(y)) \in m(R)$.

For $m(R); m(1') \subseteq m(R)$, let $X \stackrel{\text{df}}{=} \{x-Rz, y-1'z\}$. Then, by \mathcal{K} -correctness of the rule $(1'1)$, $X \cup \{xRy\}$ is a \mathcal{K} -set. Therefore, for every \mathcal{K} -structure $\mathcal{M} = (U, m)$

and for every valuation v in \mathcal{M} , if $(v(x), v(z)) \in m(R)$ and $(v(y), v(z)) \in m(1')$, then $(v(x), v(y)) \in m(R)$. \mathcal{K} -correctness of the rule (1'1) implies symmetry of $m(1')$, so if $(v(x), v(y)) \in m(R) ; m(1')$, then $(v(x), v(y)) \in m(R)$.

Hence, the extensionality property follows. \square

RL(1, 1')-correctness of the decomposition rules can be proved as in Proposition 2.5.1. Thus, by Theorem 2.7.1, we have:

Proposition 2.7.3.

1. The RL(1, 1')-rules are RL(1, 1')-correct;
2. The RL(1, 1')-axiomatic sets are RL(1, 1')-sets.

Following the general method of proving soundness described in Sect. 2.6, the above proposition implies:

Proposition 2.7.4. *Let φ be an RL(1, 1')-formula. If φ is RL(1, 1')-provable, then it is RL(1, 1')-valid.*

Corollary 2.7.1. *Let φ be an RL(1, 1')-formula. If φ is RL(1, 1')-provable, then it is true in all standard RL(1, 1')-models.*

The notions of an RL(1, 1')-proof tree, a closed branch, a closed RL(1, 1')-proof tree, and an RL(1, 1')-proof of an RL(1, 1')-formula are defined as in Sect. 2.4. Observe that any application of the rules of RL(1, 1')-dual tableau, in particular an application of the specific rules (1'1) and (1'2), preserves the formulas built with atomic terms or their complements (see Fact 2.5.1). Therefore, whenever an atomic formula xRy and the formula $x - Ry$ appear in a branch, then the branch is closed. Thus, the closed branch property holds.

A branch b of an RL(1, 1')-proof tree is said to be RL(1, 1')-complete whenever it is closed or it satisfies RL(1')-completion conditions which consist of the completion conditions of RL-dual tableau adjusted to RL(1, 1')-language and the following completion conditions determined by the specific rules of RL(1, 1')-dual tableau:

For all object symbols x and y and for every atomic relational term R ,

- Cpl(1'1) If $xRy \in b$, then for every object symbol z , either $xRz \in b$ or $y1'z \in b$, obtained by an application of the rule (1'1);
- Cpl(1'2) If $xRy \in b$, then for every object symbol z , either $x1'z \in b$ or $zRy \in b$, obtained by an application of the rule (1'1).

Let b be an open branch of an RL(1, 1')-proof tree. We define a branch structure $\mathcal{M}^b = (U^b, m^b)$ as in RL-logic adapting it to the RL(1, 1')-language. In particular, $m^b(R) \stackrel{\text{df}}{=} \{(x, y) \in U^b \times U^b : xRy \notin b\}$, for $R \in \{1, 1'\}$.

Proposition 2.7.5 (Branch Model Property). *For every open branch b of an RL(1, 1')-proof tree, a branch structure \mathcal{M}^b is an RL(1, 1')-model.*

Proof. We need to prove: (1) $m^b(1) = U^b \times U^b$, (2) $m^b(1')$ is an equivalence relation on U^b , and (3) $m^b(1'); m^b(R) = m^b(R); m^b(1') = m^b(R)$, for every atomic relational term R .

Proof of (1). Clearly, for all object symbols x and y it must be $x1y \notin b$, for otherwise b would be closed. Thus $m^b(1) = U^b \times U^b$.

Proof of (2). For every $x \in U^b$, $x1'x \notin b$, for otherwise b would be closed. Therefore $(x, x) \in m^b(1')$, hence $m^b(1')$ is reflexive. Assume $(x, y) \in m^b(1')$, that is $x1'y \notin b$. Suppose $(y, x) \notin m^b(1')$. Then $y1'x \in b$. By the completion condition Cpl(1'1), either $y1'y \in b$ or $x1'y \in b$, a contradiction. Therefore $m^b(1')$ is symmetric. To prove transitivity, assume that $(x, y) \in m^b(1')$ and $(y, z) \in m^b(1')$, that is $x1'y \notin b$ and $y1'z \notin b$. Suppose $(x, z) \notin m^b(1')$. Then $x1'z \in b$. By the completion condition Cpl(1'1), either $x1'y \in b$ or $z1'y \in b$. In the first case we get a contradiction, so $z1'y \in b$. By the completion condition Cpl(1'1) applied to $z1'y$, either $z1'z \in b$ or $y1'z \in b$. In both cases we get a contradiction.

Proof of (3). Since $m^b(1')$ is reflexive, we have $m^b(R) \subseteq m^b(1'); m^b(R)$ and $m^b(R) \subseteq m^b(R); m^b(1')$.

Now, assume $(x, y) \in m^b(1'); m^b(R)$, that is there exists $z \in U^b$ such that $x1'z \notin b$ and $zRy \notin b$. Suppose $(x, y) \notin m^b(R)$. Then $xRy \in b$. By the completion condition Cpl(1'2), for every $z \in U^b$, either $x1'z \in b$ or $zRy \in b$, a contradiction.

Assume $(x, y) \in m^b(R); m^b(1')$, that is, by symmetry of $m^b(1')$, there exists $z \in U^b$ such that $xRz \notin b$ and $y1'z \notin b$. Suppose $(x, y) \notin m^b(R)$. Then $xRy \in b$. By the completion condition Cpl(1'1), for every $z \in U^b$, either $xRz \in b$ or $y1'z \in b$, a contradiction. \square

Any structure \mathcal{M}^b defined as above is referred to as an $\text{RL}(1, 1')$ -branch model. Let $v^b: \text{OS}_{\text{RL}(1, 1')} \rightarrow U^b$ be a valuation in \mathcal{M}^b such that $v^b(x) = x$, for every object symbol x .

Proposition 2.7.6 (Satisfaction in Branch Model Property). *For every open branch b of an $\text{RL}(1, 1')$ -proof tree and for every $\text{RL}(1, 1')$ -formula φ , if $\mathcal{M}^b, v^b \models \varphi$, then $\varphi \notin b$.*

The proof of the above proposition is similar to the proof of Proposition 2.5.5.

Given an $\text{RL}(1, 1')$ -branch model \mathcal{M}^b , since $m^b(1')$ is an equivalence relation on U^b , we may define the quotient model $\mathcal{M}_q^b = (U_q^b, m_q^b)$ as follows:

- $U_q^b = \{\|x\| : x \in U^b\}$, where $\|x\|$ is the equivalence class of $m^b(1')$ generated by x ;
- $m_q^b(c) = \|c\|$, for every object constant c ;
- $m_q^b(R) = \{(\|x\|, \|y\|) \in U_q^b \times U_q^b : (x, y) \in m^b(R)\}$, for every atomic relational term R ;
- m_q^b extends to all the compound relational terms as in the $\text{RL}(1, 1')$ -models.

Since a branch model satisfies the extensionality property, the definition of $m_q^b(R)$ is correct, that is:

$$\text{If } (x, y) \in m^b(R) \text{ and } (x, z), (y, t) \in m^b(1'), \text{ then } (z, t) \in m^b(R).$$

Let v_q^b be a valuation in \mathcal{M}_q^b such that $v_q^b(x) = \|x\|$, for every object symbol x .

Proposition 2.7.7.

1. The model \mathcal{M}_q^b is a standard $\text{RL}(1, 1')$ -model;
2. For every $\text{RL}(1, 1')$ -formula φ , $\mathcal{M}^b, v^b \models \varphi$ iff $\mathcal{M}_q^b, v_q^b \models \varphi$.

Proof.

1. We have to show that $m_q^b(1')$ is the identity on U_q^b . Indeed, we have:

$$(\|x\|, \|y\|) \in m_q^b(1') \text{ iff } (x, y) \in m^b(1') \text{ iff } \|x\| = \|y\|.$$

2. The proof is by an easy induction on the complexity of relational terms. □

Proposition 2.7.8. *Let φ be an $\text{RL}(1, 1')$ -formula. If φ is true in all standard $\text{RL}(1, 1')$ -models, then φ is $\text{RL}(1, 1')$ -provable.*

Proof. Assume φ is true in all standard $\text{RL}(1, 1')$ -models. Suppose there is no any closed $\text{RL}(1, 1')$ -proof tree for φ . Then there exists a complete $\text{RL}(1, 1')$ -proof tree for φ with an open branch, say b . Since $\varphi \in b$, by Proposition 2.7.6, φ is not satisfied by v^b in the branch model \mathcal{M}^b . By Proposition 2.7.7(2.), φ is not satisfied by v_q^b in the quotient model \mathcal{M}_q^b . Since \mathcal{M}_q^b is a standard $\text{RL}(1, 1')$ -model, φ is not true in all standard $\text{RL}(1, 1')$ -models, a contradiction. □

From Fact 2.7.1, Propositions 2.7.4 and 2.7.8 we get:

Theorem 2.7.2 (Soundness and Completeness of $\text{RL}(1, 1')$). *Let φ be an $\text{RL}(1, 1')$ -formula. Then the following conditions are equivalent:*

1. φ is $\text{RL}(1, 1')$ -valid;
2. φ is true in all standard $\text{RL}(1, 1')$ -models;
3. φ is $\text{RL}(1, 1')$ -provable.

The class of $\text{RL}(1, 1')$ -models corresponds to the class **FRA** of full relation algebras, as it will be proved in Sect. 2.9.

2.8 Discussion of Various Rules for Relation $1'$

As in the case of **F**-proof system, in $\text{RL}(1, 1')$ -dual tableau we can admit some other rules for relation $1'$. There are at least three possibilities of choosing rules for $1'$. In this section, rules (1'1) and (1'2) (see p. 46) will be called *classical*.

In many relational proof systems in the style of Rasiowa–Sikorski the following specific rules are used:

Standard

$$(1'1)^1 \frac{xRy}{xRz, xRy \mid z1'y, xRy} \quad (1'2)^1 \frac{xRy}{x1'z, xRy \mid zRy, xRy}$$

$$(\text{sym})^1 \frac{x1'y}{y1'x}$$

where R is any relational variable or relational constant, x, y, z are any object symbols.

Then, the axiomatic sets are those of $\text{RL}(1, 1')$ -dual tableau, i.e., (Ax) from Sect. 2.5 and (Ax1) and (Ax2) from Sect. 2.7.

Completion conditions for rules $(1'1)^1$ and $(1'2)^2$ are analogous to those for the classical rules, and for the rule $(\text{sym})^1$ we have:

$\text{Cpl}(\text{sym})^1$ If $x1'y \in b$, then $y1'x \in b$.

Note that these standard specific rules differ from the classical ones in the rules $(1'2)^1$ and $(\text{sym})^1$. The rule $(1'2)^1$ seems to be more natural than the rule $(1'2)$. On the metalogical level the rule $(1'2)^1$ says: xRy is valid iff the conjunction of $x1'z \vee xRy$ and $zRy \vee xRy$ is valid. In the rule $(1'2)$ positions of variables x and z in a formula $x1'z$ are interchanged and this, in some sense, expresses symmetry of $1'$. From a logical point of view it is immaterial which of these sets of specific rules is used. The standard rules are correct and give the complete proof system. The proof of completeness is similar to the proof for the system with the classical rules. The only difference is that symmetry of $m^b(1')$ follows from the completion condition corresponding to the rule $(\text{sym})^1$.

The following rules are essentially different from the classical ones.

Negative standard

$$(1'1)^2 \frac{x-1'y, y-Rz}{x-Rz, x-1'y, y-Rz} \quad (1'2)^2 \frac{x-1'y, z-Rx}{z-Ry, x-1'y, z-Rx}$$

$$(\text{sym})^2 \frac{x-1'y}{y-1'x} \quad (\text{ref})^2 \frac{}{x-1'x}$$

$$(1)^2 \frac{}{x-1y}$$

where R is any relational variable or relational constant, and x, y, z are any object symbols.

The axiomatic sets are those of \mathbf{RL} -dual tableau, i.e., (Ax) from Sect. 2.5.

The negative forms of specific rules are dual to the rules of a tableau system for the relational logic. Note that the rules $(1'1)^2$ and $(1'2)^2$ can be applied only to negated atomic formulas. Moreover, contrary to the classical rules, they do not branch a proof tree and do not involve introduction of a variable which makes them more suitable for implementation. Our conclusions from the discussion of different forms of the specific rule for the first-order logic in Sect. 1.8 are also relevant to this case.

Below we prove completeness of the proof system with the negative form of the rules for $1'$. Moreover, we show that all these rules are needed to prove it.

The completion conditions corresponding to the negative rules are:

For all object symbols x , y , and z and for every atomic relational term R ,

$\text{Cpl}(1'1)^2$ If $x-1'y \in b$ and $y-Rz \in b$, then $x-Rz \in b$, obtained by an application of the rule $(1'1)^2$;

$\text{Cpl}(1'2)^2$ If $x-1'y \in b$ and $z-Rx \in b$, then $z-Ry \in b$, obtained by an application of the rule $(1'2)^2$;

$\text{Cpl}(\text{sym})^2$ If $x-1'y \in b$, then $y-1'x \in b$, obtained by an application of the rule $(\text{sym})^2$;

$\text{Cpl}(\text{ref})^2$ For every object symbol x , $x-1'x \in b$, obtained by an application of the rule $(\text{ref})^2$;

$\text{Cpl}(1)^2$ For all object symbols x and y , $x-1y \in b$, obtained by an application of the rule $(1)^2$.

Proposition 2.8.1 (Closed Branch Property). *For every branch of a proof tree in the relational dual tableau with negative standard rules, if xRy and $x-Ry$, for an atomic R , belong to the branch, then the branch can be closed.*

Proof. Although the rule $(\text{sym})^2$ does not preserve the formula $x-1'y$, if $x-1'y$ appears in a node of a branch, say b , then all the successors of this node in b include either $x-1'y$, if the rule $(\text{sym})^2$ has not been applied yet, or $y-1'x$ if the rule $(\text{sym})^2$ has been applied. If $x1'y$ is in a node, say n , of branch b , then either $x-1'y \in n$ and $x1'y \in n$ or $y-1'x \in n$ and $x1'y \in n$. In the former case, the branch is closed. In the latter, we can apply rule $(\text{sym})^2$ to $y-1'x$ in node n . Then we obtain a node which includes $x-1'y$ and $x1'y$ and we close the branch. All the remaining rules preserve the formulas built with atomic terms or their complements. Thus, the closed branch property holds. \square

Since the rules are applied to formulas built with complements of atomic terms, we need to change the definition of a branch structure. Let b be an open branch of a proof tree. We define a branch structure $\mathcal{M}^b = (U^b, m^b)$ as follows:

- $U^b = \mathbb{O}\mathbb{S}_{\mathbf{RL}(1,1')}$;
- $m^b(R) = \{(x, y) \in U^b \times U^b : x-Ry \in b\}$, for every atomic relational term R ;
- m^b extends to all the compound terms as in \mathbf{RL} -models.

Let $v^b: \mathbb{O}\mathbb{S}_{\mathbf{RL}(1,1')} \rightarrow U^b$ be a valuation in \mathcal{M}^b such that $v^b(x) = x$, for every object symbol x . First, we prove that $m^b(1')$ is an equivalence relation.

Proposition 2.8.2. $m^b(1')$ is an equivalence relation.

Proof. By the completion condition $\text{Cpl}(\text{ref})^2$, for every $x \in U^b$, $x-1'x \in b$, which means that $(x, x) \in m^b(1')$. Hence, $m^b(1')$ is reflexive. Assume $(x, y) \in m^b(1')$, that is $x-1'y \in b$. By the completion condition $\text{Cpl}(\text{sym})^2$, $y-1'x \in b$, hence $(y, x) \in m^b(1')$. Thus, $m^b(1')$ is symmetric. Assume $(x, y) \in m^b(1')$, $(y, z) \in m^b(1')$, that is $x-1'y \in b$ and $y-1'z \in b$. By the completion condition $\text{Cpl}(1'1)^2$, $x-1'z \in b$, hence $(x, z) \in m^b(1')$, which proves transitivity of $m^b(1')$. \square

Note that the rules $(\text{ref})^2$ and $(\text{sym})^2$ are needed to prove reflexivity and symmetry of $m^b(1')$, respectively. The rule $(1'1)^2$ is needed to prove transitivity of $m^b(1')$.

Proposition 2.8.3. For every atomic relational term R :

1. $m^b(1'); m^b(R) \subseteq m^b(R)$;
2. $m^b(R); m^b(1') \subseteq m^b(R)$.

Proof. For 1., assume $(x, y) \in m^b(1'); m^b(R)$. Then there exists $z \in U^b$ such that $(x, z) \in m^b(1')$ and $(z, y) \in m^b(R)$, that is $x-1'z \in b$ and $z-Ry \in b$. By the completion condition $\text{Cpl}(1'1)^2$, $x-Ry \in b$, hence $(x, y) \in m^b(R)$.

The proof of 2. is similar. \square

By the above proposition, the branch structure satisfies the extensionality property assumed in $\text{RL}(1, 1')$ -models. Moreover, due to the completion condition $\text{Cpl}(1)^2$, it satisfies the constraint $m^b(1) = U^b \times U^b$. We conclude that the branch structure \mathcal{M}^b is an $\text{RL}(1, 1')$ -model. Therefore, the quotient model can be defined in a similar way as in the proof of the completeness of $\text{RL}(1, 1')$ (see p. 49). The satisfaction in branch model property can be proved by induction on the complexity of relational terms, that is we have:

Proposition 2.8.4 (Satisfaction in Branch Model Property). For every open branch b and for every $\text{RL}(1, 1')$ -formula φ , if $\mathcal{M}^b, v^b \models \varphi$, then $\varphi \notin b$.

Moreover, the following can be proved in a similar way as in $\text{RL}(1, 1')$ -dual tableau:

Proposition 2.8.5.

1. \mathcal{M}_q^b is a standard $\text{RL}(1, 1')$ -model;
2. $\mathcal{M}^b, v^b \models \varphi$ iff $\mathcal{M}_q^b, v_q^b \models \varphi$.

Then completeness of $\text{RL}(1, 1')$ -dual tableau with negative specific rules follows from the above propositions.

We may refine the set of negative rules in analogy with the classical set of specific rules by deleting the symmetry rule and interchanging the positions of x and y in the rule $(1'2)^2$. Then we have less specific rules than in the negative standard set.

$$\begin{array}{ll}
 (1'1)^3 \frac{x-1'y, y-Rz}{x-Rz, x-1'y, y-Rz} & (1'2)^3 \frac{y-1'x, z-Rx}{z-Ry, y-1'x, z-Rx} \\
 (\text{ref})^3 \frac{}{x-1'x} & (1)^3 \frac{}{x-1y}
 \end{array}$$

where R is any relational variable or relational constant, and x, y, z are any object symbols.

Then, the axiomatic sets are those of RL-dual tableau, i.e., (Ax) from Sect. 2.5.

This set of specific rules together with the decomposition rules provides a complete proof system for $\text{RL}(1, 1')$. The proof of completeness is similar to the previous one except for the parts where rules $(1'2)^2$ and $(\text{sym})^2$ are used.

2.9 Full Relation Algebras and Relational Logics

We recall that the class of full relation algebras, FRA, is the class of algebras of the form $(\mathcal{P}(U \times U), \cup, \cap, -, \emptyset, U \times U, ;, ^{-1}, 1')$, where U is a non-empty set, $1'$ is the identity on U , operations $-, \cup$, and \cap are Boolean operations, $^{-1}$ and $;$ are converse and composition of binary relations, respectively.

Let \mathcal{C} be a class of relation algebras. An equation $R_1 = R_2$, where R_1 and R_2 are relation algebra terms, is said to be \mathcal{C} -valid whenever it is true in all algebras of \mathcal{C} . The following theorem follows directly from the definition of validity in relational logics.

Theorem 2.9.1. *Let R be any relational term and let x and y be any two different object variables. Then xRy is true in all standard $\text{RL}(1, 1')$ -models iff $R = 1$ is FRA-valid.*

Proof. (\rightarrow) Assume xRy is true in all standard $\text{RL}(1, 1')$ -models and suppose that $R = 1$ is not FRA-valid. Then there exists a FRA-algebra \mathfrak{A} of binary relations on a set U such that $R^{\mathfrak{A}} \neq 1^{\mathfrak{A}}$, where $1^{\mathfrak{A}}$ is the universal relation on U . Hence, for some $u, u' \in U$, $(u, u') \notin R^{\mathfrak{A}}$. Consider an $\text{RL}(1, 1')$ -model $\mathcal{M}_{\mathfrak{A}} = (U, m_{\mathfrak{A}})$ such that:

- $m_{\mathfrak{A}}(P) = P^{\mathfrak{A}}$, for every relational variable P and for $P \in \{1, 1'\}$;
- $m_{\mathfrak{A}}$ extends to all the compound relational terms as in $\text{RL}(1, 1')$ -models (see Sect. 2.3).

It follows that for every relational term Q , $m_{\mathfrak{A}}(Q) = Q^{\mathfrak{A}}$. Consider a valuation v in model $\mathcal{M}_{\mathfrak{A}}$ such that $v(x) = u$ and $v(y) = u'$. Such a valuation v exists, since variables x and y are different. Then $(v(x), v(y)) \notin m_{\mathfrak{A}}(R)$, a contradiction.

(\leftarrow) Assume $R = 1$ is FRA-valid and suppose xRy is not true in a standard $\text{RL}(1, 1')$ -model $\mathcal{M} = (U, m)$. Then there exists a valuation v in \mathcal{M} such that $(v(x), v(y)) \notin m(R)$. Consider a FRA-algebra $\mathfrak{A}_{\mathcal{M}} = (\mathcal{P}(U \times U), \cup, \cap, -, \emptyset, U \times U, ;, ^{-1}, m(1'))$ such that $P^{\mathfrak{A}_{\mathcal{M}}} = m(P)$, for every relational variable P . It follows that for every relational term Q , $Q^{\mathfrak{A}_{\mathcal{M}}} = m(Q)$. Since $R = 1$ is FRA-valid, it must be true in $\mathfrak{A}_{\mathcal{M}}$. Hence, $R^{\mathfrak{A}_{\mathcal{M}}} = 1^{\mathfrak{A}_{\mathcal{M}}} = U \times U$. This yields $m(R) = U \times U$, a contradiction. \square

Due to the above theorem and Theorem 2.7.2, we obtain:

Theorem 2.9.2. *Let xRy be an $\text{RL}(1, 1')$ -formula. Then xRy is $\text{RL}(1, 1')$ -provable iff $R = 1$ is FRA -valid.*

An example of $\text{RL}(1, 1')$ -provable formula which represents a FRA -valid equation which is not RA -valid is presented in Sect. 2.10.

As a consequence of Proposition 2.2.1(1.), we have:

Proposition 2.9.1. *Let R and S be $\text{RL}(1, 1')$ -terms. Then $R \subseteq S$ holds in every FRA -algebra iff $x(-R \cup S)y$ is $\text{RL}(1, 1')$ -valid.*

Since the class RRA is a variety generated by FRA , it is known that:

Theorem 2.9.3. *The set of equations valid in RRA coincides with the set of equations valid in FRA .*

Hence, $\text{RL}(1, 1')$ -logic is a tool for verification of validity of equations in the class RRA of representable relation algebras.

2.10 An Example of a Relational Dual Tableau Proof

As an example of an $\text{RL}(1, 1')$ -provable equation let us consider the equation $\tau = 1$, for $\tau \stackrel{\text{df}}{=} (1; \rho; 1)$ and $\rho \stackrel{\text{df}}{=} (A \cup B \cup C \cup D \cup E)$, where A, B, C, D , and E are defined as follows, for relational variables R and N :

- $A = -(1; R; 1)$,
- $B = [R \cap -(N; N) \cap (R; N)]$,
- $C = (N; N \cup R; R) \cap N$,
- $D = [(R \cup R^{-1} \cup 1') \cap N]$,
- $E = -(R \cup R^{-1} \cup 1' \cup N)$.

This equation is RRA -valid, while it is not RA -valid.

Proposition 2.10.1. *The equation τ is true in all representable relation algebras.*

Proof. Suppose τ is not true in some $\mathfrak{A} \in \text{FRA}$, that is $\tau \neq 1$. Then, by Proposition 2.2.1(3.), $\rho = \emptyset$ in \mathfrak{A} . It means that the following are true in \mathfrak{A} :

- (1) $-(1; R; 1) = \emptyset$,
- (2) $(R \cap -(N; N) \cap (R; N)) = \emptyset$,
- (3) $[(R; R \cup N; N) \cap N] = \emptyset$,
- (4) $[(R \cup R^{-1} \cup 1') \cap N] = \emptyset$,
- (5) $-(R \cup R^{-1} \cup 1' \cup N) = \emptyset$.

The first condition implies that $R \neq \emptyset$, since the following holds:

$$-(1 ; R ; 1) = \emptyset \text{ iff } (1 ; R ; 1) = 1 \text{ iff } R \neq \emptyset.$$

The condition (2) is equivalent to $R \subseteq (N ; N) \cap (R ; N)$, which means that $R \subseteq N ; N$ and $R \subseteq R ; N$. From condition (3) it follows that $(R ; R) \cap N = \emptyset$ and $(N ; N) \cap N = \emptyset$. Condition (4) implies that relations $R \cap N$, $R^{-1} \cap N$, $1' \cap N$ are empty. The condition (5) means that $R \cup R^{-1} \cup 1' \cup N$ is the universal relation 1.

Summarizing, if τ is not true in \mathfrak{A} , then the following must hold:

- (a) $R \neq \emptyset$,
- (b1) $R \subseteq N ; N$,
- (b2) $R \subseteq R ; N$,
- (c1) $(R ; R) \cap N = \emptyset$,
- (c2) $(N ; N) \cap N = \emptyset$,
- (d1) $R \cap N = \emptyset$,
- (d2) $R^{-1} \cap N = \emptyset$,
- (d3) $1' \cap N = \emptyset$,
- (e) $(R \cup R^{-1} \cup 1' \cup N) = 1$.

By (a), (b2), and (d1) there are different x, y in the universe of \mathfrak{A} such that $(x, y) \in R$. By (b1) an element z must exist such that $(x, z), (z, y) \in N$. Since $R \cap N = \emptyset$, we have $z \neq x, y$. Figure 2.3 presents this part of \mathfrak{A} . Thick lines and thin lines correspond to relations R and N , respectively.

By (b2) there is v such that $(x, v) \in R$ and $(v, y) \in N$. Since by (d3) $1' \cap N = \emptyset$, $v \neq y$. If $v = x$ or $v = z$, then $R \cap N \neq \emptyset$. Hence $v \neq x, y, z$, $(x, v) \in R$, and $(v, y) \in N$ as presented in Fig. 2.4a. By condition (e), $(z, v) \in R$ or $(v, z) \in R$ or $(z, v) \in N$. Suppose $(v, z) \in R$. Then $(x, v), (v, z) \in R$ and $(x, z) \in N$ which implies that $(R ; R) \cap N$ is not empty, a contradiction with condition (c1). Suppose $(z, v) \in N$. Then $(z, v), (v, y), (z, y) \in N$, which contradicts condition (c2). So $(z, v) \in R$, and hence the algebra \mathfrak{A} must look as shown in Fig. 2.4b. Since $(x, v) \in R$ and by (b1) $R \subseteq (N ; N)$, there must exist an element s such that $(x, s), (s, v) \in N$. By condition (d1), $s \neq y, v$, for otherwise $R \cap N \neq \emptyset$. By condition (d3), $s \neq x$, for otherwise $1' \cap N \neq \emptyset$. Suppose $s = z$. Then $(z, y), (v, y), (z, v) \in N$, which means that $(N ; N) \cap N \neq \emptyset$, a contradiction with condition (c2). Therefore, s is distinct from x, y, z , and v (see Fig. 2.5a). Hence, $(x, y), (x, v) \in R$ and $(z, y), (x, z), (v, y), (x, s), (s, v) \in N$.

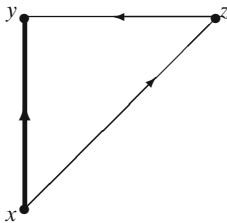


Fig. 2.3 Algebra \mathfrak{A} , step 1

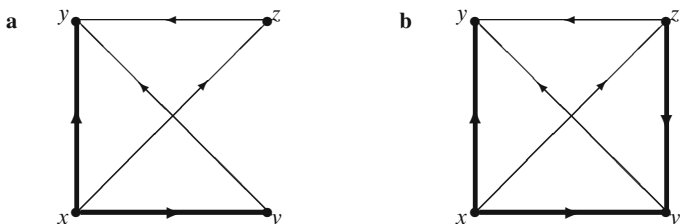


Fig. 2.4 Algebra \mathfrak{A} , step 2

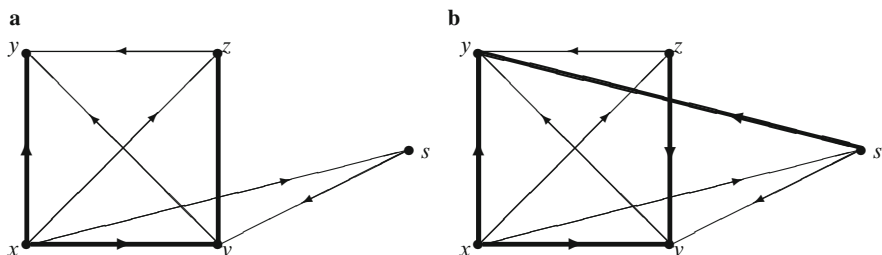


Fig. 2.5 Algebra \mathfrak{A} , step 3

Again by condition (e), $(y, s) \in R$ or $(s, y) \in R$ or $(s, y) \in N$. Assumptions $(s, y) \in N$ and $(y, s) \in R$ lead to a contradiction with conditions (c2) and (c1), respectively. Therefore $(s, y) \in R$ as shown in Fig. 2.5b. By condition (e), s and z must be in one of the relations R, R^{-1} or N . If $(s, z) \in R$, then we have $(s, z), (z, v) \in R$ and $(s, v) \in N$, a contradiction with condition (c1). Similarly, we can prove that $(z, s) \notin R$. Therefore $(s, z) \in N$, but then $(N ; N) \cap N$ is non-empty, a contradiction. \square

Proposition 2.10.2. *There exists a relation algebra \mathfrak{A} in which the equation τ is not true.*

Proof. Let $\mathfrak{A} = (W, +, \cdot, -, 0, 1, ;, \smile, 1')$ be a relation algebra such that $(W, +, \cdot, -, 0, 1)$ is a Boolean algebra generated by the four atoms a, b, c , and d and the operations $;$ and \smile are defined as follows:

	\smile
a	a
b	c
c	b
d	d

$;$	a	b	c	d
a	a	b	c	d
b	b	b	1	$b + d$
c	c	1	c	$c + d$
d	d	$b + d$	$c + d$	$a + b + c$

Let v be a valuation such that $v(1') = a, v(R) = b, v(N) = d$. We show that the equation τ is not satisfied in \mathfrak{A} by v .

From Proposition 2.10.1 it is known that unsatisfiability of τ in a relation algebra implies that all the equations (1)–(5) are true in such an algebra. Therefore it suffices to show that these equations are true in \mathfrak{A} .

Since $(W, +, \cdot, -, 0, 1)$ is a Boolean algebra generated by the atoms a, b, c, d and $1 = a + b + c + d$, the following hold:

- (1) $-(1; b; 1) = -[(a + b + c + d); b; (a + b + c + d)] = -(1; 1) = 0$,
since $;$ is an associative operation that distributes over $+$,
- (2) $(b \cdot -[(d; d) \cdot (b; d)]) = b \cdot -[(a + b + c) \cdot (b + d)] = b \cdot -b = 0$,
- (3) $[(b; b + d; d) \cdot d] = [b + (a + b + c) \cdot d] = 0$,
- (4) $[(b + b^\sim + 1') \cdot d] = (a + b + c) \cdot d = 0$,
- (5) $-(b \cdot b^\sim + 1' + d) = -(a + b + c + d) = -1 = 0$.

Therefore we may conclude that τ is not true in \mathfrak{A} . □

Below we present a construction of an $\text{RL}(1, 1')$ -proof of the formula $u\tau w$ which provides a proof of RRA -validity of the equation $\tau = 1$.

It is easy to show that in a proof tree of $u\tau w$, if a formula $u\tau w$ occurs in a node of this tree, then it is possible to build a subtree of this proof tree with this formula at the root such that it ends with exactly one non-axiomatic node containing at least one of the following formulas: zAv , zBv , zCv , zDv , and zEv , for some variables z and v . Therefore, in such cases instead of building long subtrees we will use the following derived rules:

For all object symbols u, w, z , and v ,

$$(Azv) \frac{u\tau w}{zAv, u\tau w} \quad (Bzv) \frac{u\tau w}{zBv, u\tau w} \quad (Czv) \frac{u\tau w}{zCv, u\tau w}$$

$$(Dzv) \frac{u\tau w}{zDv, u\tau w} \quad (Ezv) \frac{u\tau w}{zEv, u\tau w}$$

By way of example, in Fig. 2.6 we present a derivation of rule (Azv) .

Similarly, we can admit the following derived rules:

$$(1'^*) \frac{x1'y}{y1'x} \quad (Rxyz) \frac{x-Ry, u\tau w}{x-Nz, z-Ny, x-(R; N)y, x-Ry, u\tau w}$$

z is a new variable

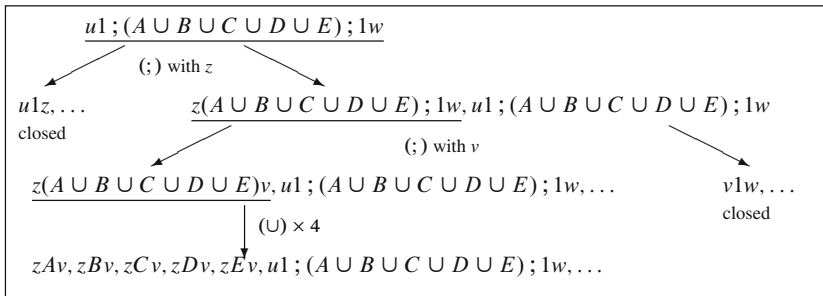


Fig. 2.6 A derivation of rule (Azv)

$$\begin{aligned}
 (RN1'xyz)_1 & \frac{z-Rx, z-Ny, u\tau w}{x1'y, z-Rx, z-Ny, u\tau w} \frac{z-Rx, z-Ny, u\tau w}{y1'x, z-Rx, z-Ny, u\tau w} \\
 (RN1'xyz)_2 & \frac{x-Rz, y-Nz, u\tau w}{x1'y, x-Rz, y-Nz, u\tau w} \frac{x-Rz, y-Nz, u\tau w}{y1'x, x-Rz, y-Nz, u\tau w} \\
 (RRNxyz) & \frac{x-Ry, y-Rz, x-Nz, u\tau w}{closed} \\
 (NNNxyz) & \frac{x-Ny, y-Nz, x-Nz, u\tau w}{closed}
 \end{aligned}$$

By way of example, in Figs. 2.10 and 2.11 we show how to obtain the derived rules $(Rxyz)$ and $(RN1'xyz)_1$, respectively. The remaining derived rules are obtained in a similar way. It is easy to check that the derived rule (Cxy) is needed to get $(RRNxyz)$ and $(NNNxyz)$, while (Dxy) is needed in the proofs of $(RN1'xyz)_1$ and $(RN1'xyz)_2$. Figure 2.7 presents an $RL(1, 1')$ -proof of $u\tau w$.

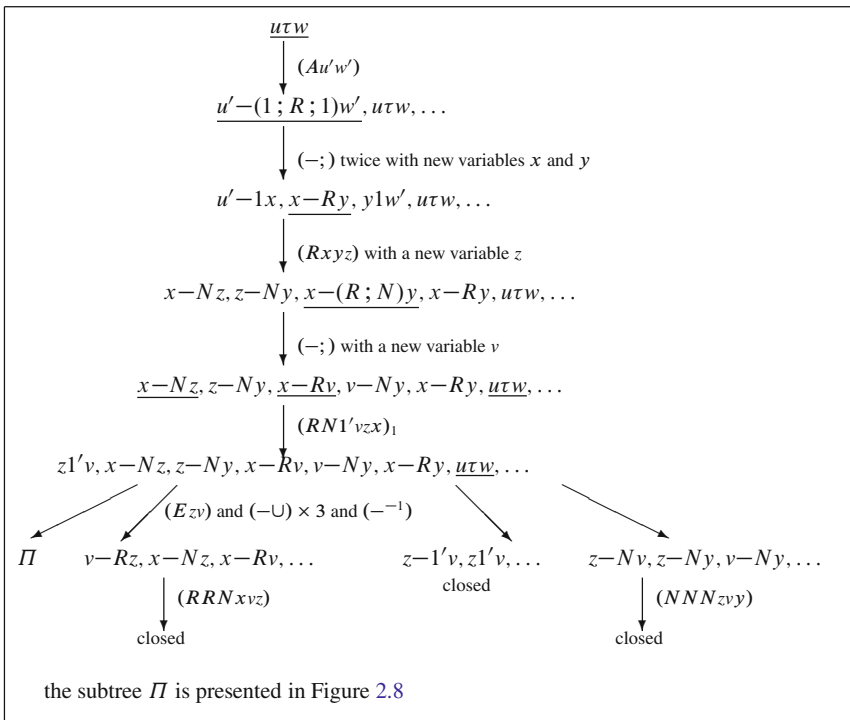


Fig. 2.7 An $RL(1, 1')$ -proof of $u\tau w$

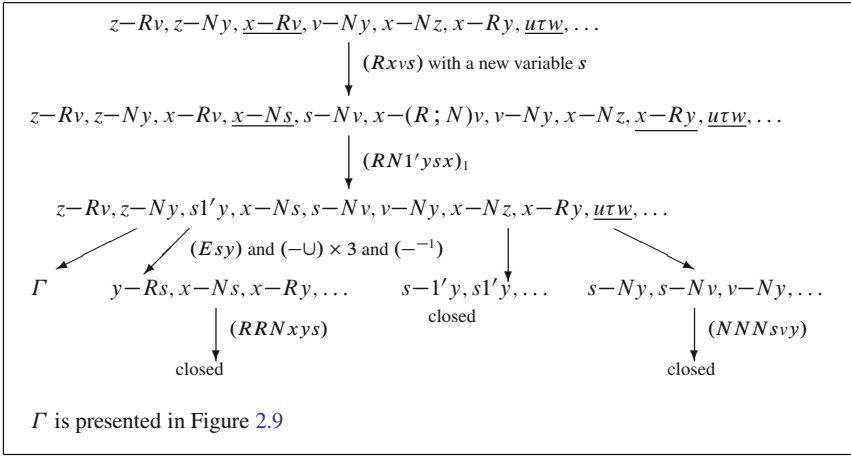


Fig. 2.8 The subtree Π

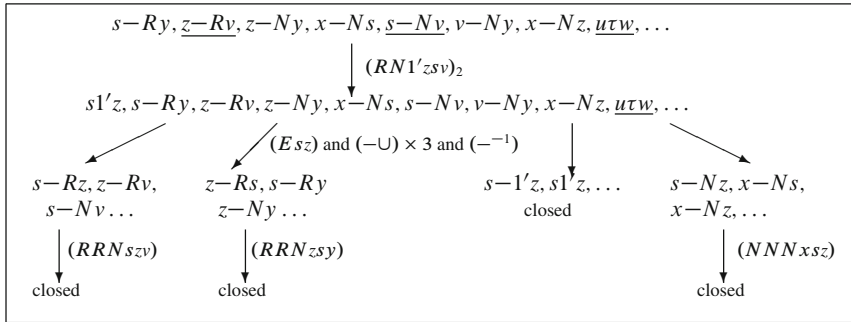


Fig. 2.9 The subtree Γ

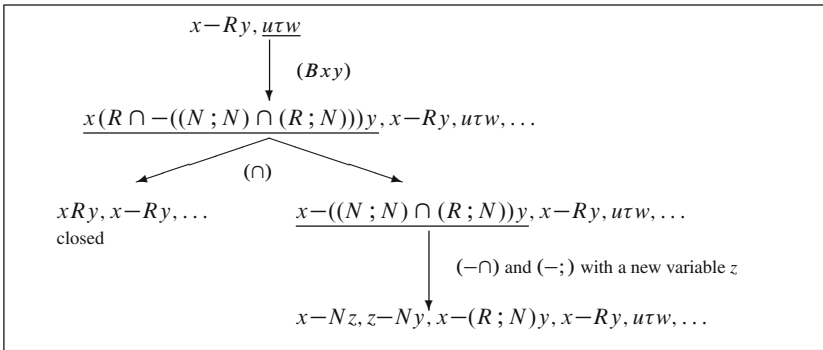


Fig. 2.10 A derivation of the rule (Rxyz)

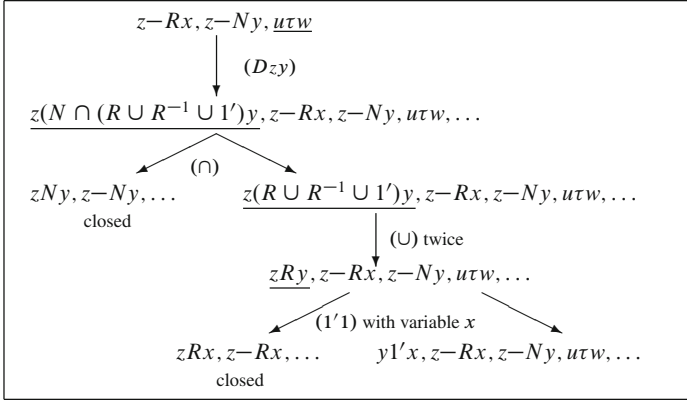


Fig. 2.11 A derivation of the rule $(RN1'xyz)_1$

2.11 Relational Entailment

The logic $RL(1, 1')$ can be used to verify entailment of formulas from a finite set of formulas. The method is based on the following fact. Let $n \geq 1$ and let R_1, \dots, R_n, R be binary relations on a set U and let $1 = U \times U$. It is known ([Tar41], see also Proposition 2.2.1(7.)), that $R_1 = 1, \dots, R_n = 1$ imply $R = 1$ iff $(1; -(R_1 \cap \dots \cap R_n); 1) \cup R = 1$. It follows that for every $RL(1, 1')$ -model \mathcal{M} , $\mathcal{M} \models xR_1y, \dots, \mathcal{M} \models xR_ny$ imply $\mathcal{M} \models xRy$ iff $\mathcal{M} \models x(1; -(R_1 \cap \dots \cap R_n); 1) \cup R)y$ which means that entailment in $RL(1, 1')$ can be expressed in its language.

Example. We prove that $x-(R; -P)y$ and $x-(R; -(-P \cup Q))y$ entail $x-(R; -Q)y$, by showing that the formula:

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

is $RL(1, 1')$ -provable. Figure 2.12 presents an $RL(1, 1')$ -proof of this formula.

Now, we show that for every relation R , $R; 1 = 1$ or $1; -R = 1$. Applying Proposition 2.2.1(6.), we need to show that the formula

$$x1; -(1; -(R; 1); 1); 1) \cup (1; -R)y$$

is $RL(1, 1')$ -provable. Figure 2.13 presents an $RL(1, 1')$ -proof of this formula.

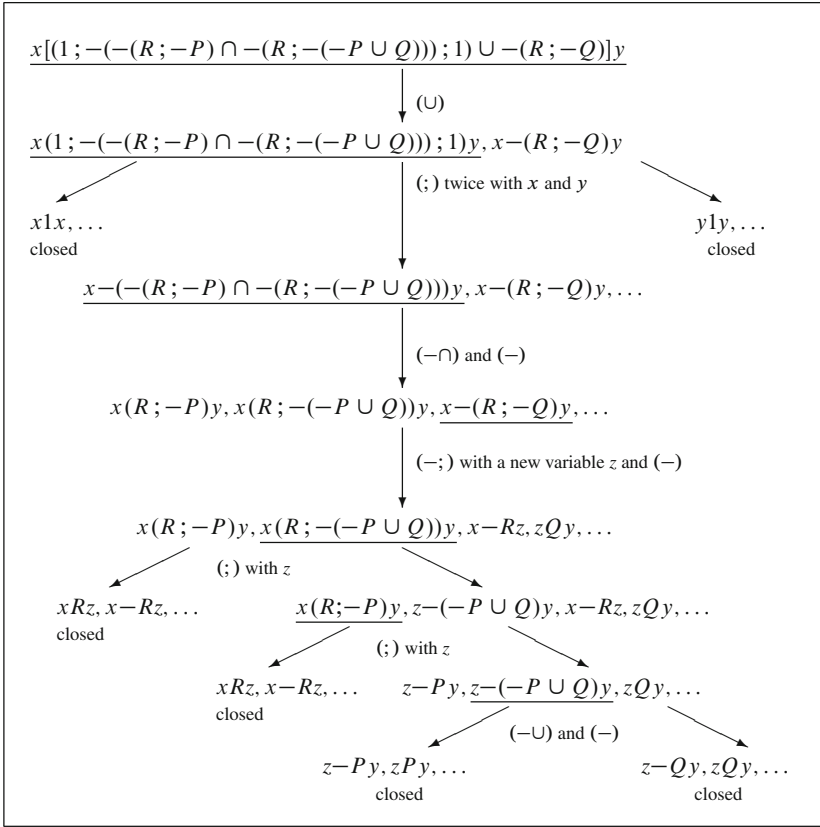


Fig. 2.12 An $RL(1, 1')$ -proof showing that $x \neg(R; -P)y$ and $x \neg(R; \neg(-P \cup Q))y$ entail $x \neg(R; -Q)y$

2.12 Decision Procedures for Some Relational Logics

In this section we present some decidable subclasses of formulas of the basic relational logic RL chosen from those presented in [BO97, Dob96a, Dob96b]. The classes are described in the syntactic terms. Usually, a distinguishing feature of a class is formulated in terms of a condition on the occurrences of the composition symbol in the formulas of the class. For the sake of simplicity, we will often say composition when referring to its occurrences.

An occurrence of the symbol of composition $(;)$ is said to be *positive* (resp. *negative*) in a formula iff it is in the scope of an even (resp. odd) number of the complement symbols in the formula. Similarly, an occurrence of the symbol of composition is *negative-positive* (resp. *positive-negative*) iff it is positive (resp. negative) and it falls into the scope of a negative (resp. positive) composition. An occurrence c of the composition is between the compositions c_1 and c_2 iff c falls into the scope of c_1 and c_2 falls into the scope of c .

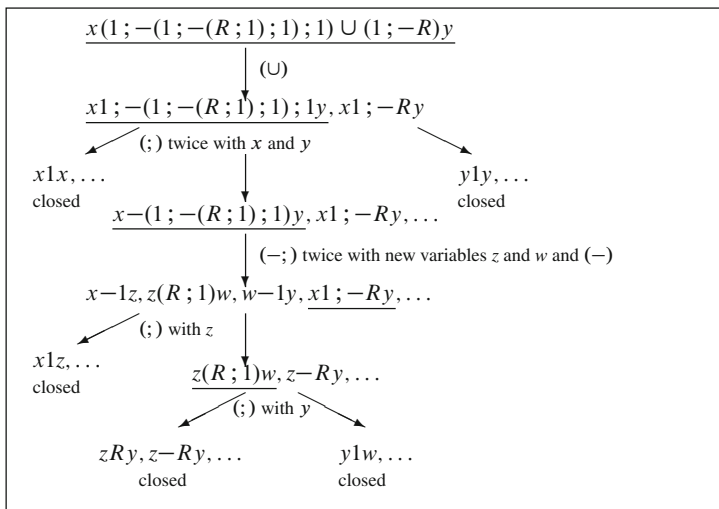


Fig. 2.13 An $RL(1, 1')$ -proof showing that $R; 1 = 1$ or $1; -R = 1$

A formula φ of the logic RL is said to be:

- *Positive* (resp. *negative*) iff all the compositions in φ are positive (resp. negative);
- *Negative-positive* if there is no any positive-negative composition in φ ;
- *1-positive* iff there is exactly one positive composition in φ (and possibly some negative ones);
- *2-positive* iff there are exactly two positive compositions in φ and there is no any negative composition in φ between the positive compositions.

The classes of formulas considered above will be referred to as:

- RL_{POS} – the class of all positive formulas;
- RL_{NEG} – the class of all negative formulas;
- RL_{NP} – the class of all negative-positive formulas;
- $RL_{POS(1)}$ – the class of all 1-positive formulas;
- $RL_{POS(2)}$ – the class of all 2-positive formulas.

Example. Let P, Q be relational variables. Consider the following RL -terms:

$$\begin{aligned}
 T_1 &= -(- (P; (Q \cap P)) \cup - (Q; P)), \\
 T_2 &= -((P; Q) \cap - (Q \cup (Q; P))), \\
 T_3 &= -((-(P; Q); -(Q; P)); -(Q; -P); -(P; -Q)), \\
 T_4 &= -((P \cup Q); -(-(Q; P); Q)), \\
 T_5 &= -(-P; -((Q; -(Q; P)); Q)).
 \end{aligned}$$

The formula xT_1y is positive, since all of its compositions are in the scope of an even number of complements, and similarly, xT_2y is negative, since all of its compositions are negative. The formula xT_3y is negative-positive, and its

positive compositions occur in the following terms that do not contain any negative occurrence of composition: $(P ; Q)$, $(Q ; P)$, $(Q ; -P)$, and $(P ; -Q)$. The formula xT_4y is 1-positive, since the only positive composition of this formula is the principal composition of $(-(Q ; P) ; Q)$ and all the other compositions are negative. The formula xT_5y is 2-positive. In order to see this, observe that xT_5y contains only two positive compositions, namely those in the term $((Q ; -(Q ; P)) ; Q)$, and there is no any negative composition between these two positive compositions.

Decidability of these classes of formulas is proved by showing that the first-order translations of their members form decidable subclasses of the classical first-order predicate logic F . A natural translation τ of relational formulas into first-order formulas is defined as follows. Let a one-to-one assignment τ' be given of object variables and relational variables of RL to individual variables and predicate symbols from the language of F , respectively. Then we define:

- $\tau(xPy) = \tau'(P)(\tau'(x), \tau'(y))$, for every atomic relational variable P ;

For all relational terms R and S :

- $\tau(x-Ry) = \neg\tau(xRy)$;
- $\tau(x(R \cup S)y) = \tau(xRy) \vee \tau(xSy)$;
- $\tau(x(R \cap S)y) = \tau(xRy) \wedge \tau(xSy)$;
- $\tau(xR^{-1}y) = \tau(yRx)$;
- $\tau(x(R ; S)y) = \exists\tau'(z)(\tau(xRz) \wedge \tau(zSy))$.

The translations of relational formulas from the classes defined above belong to the following decidable classes of first-order formulas. Let (\forall) and (\exists) denote finite strings of universal and existential quantifiers, respectively. The classes are denoted by indicating a form of the string of quantifiers in the prenex normal form of the formulas in the class:

- (\exists) – the class of existential formulas;
- (\forall) – the class of universal formulas;
- $(\forall)(\exists)$ – the class of formulas with universal quantifiers followed by existential quantifiers;
- $(\forall)\exists(\forall)$ – the class of formulas with exactly one existential quantifier between the universal quantifiers;
- $(\forall)\exists_1\exists_2(\forall)$ – the class of formulas with exactly two existential quantifiers between the universal quantifiers.

The following is well known (see [Ack54]):

Proposition 2.12.1. *The classes (\exists) , (\forall) , $(\forall)(\exists)$, $(\forall)\exists(\forall)$, and $(\forall)\exists_1\exists_2(\forall)$ have a decidable validity problem.*

The translations of relational formulas from the classes listed above fall into the following classes of first-order logic:

Proposition 2.12.2. *For every relational formula φ , the prenex form of $\tau(\varphi)$, $pf(\tau(\varphi))$, satisfies:*

1. *If φ is positive, then $pf(\tau(\varphi)) \in \forall_1 \forall_2 (\exists)$;*
2. *If φ is negative, then $pf(\tau(\varphi)) \in (\forall)$;*
3. *If φ is negative-positive, then $pf(\tau(\varphi)) \in (\forall)(\exists)$;*
4. *If φ is 1-positive, then $pf(\tau(\varphi)) \in (\forall)\exists(\forall)$;*
5. *If φ is 2-positive, then $pf(\tau(\varphi)) \in (\forall)\exists_1\exists_2(\forall)$.*

By Propositions 2.12.1 and 2.12.2, we get:

Theorem 2.12.1 (Decidable Classes of Relational Formulas). *The classes RL_{POS} , RL_{NEG} , RL_{NP} , $RL_{POS(1)}$, and $RL_{POS(2)}$ of relational formulas have a decidable validity problem.*

Below we present sound and complete dual tableaux that are decision procedures for the classes RL_{POS} , RL_{NEG} , RL_{NP} . Note that RL -dual tableau is not a decision procedure for these classes, since if the rule $(;)$ is applicable, then it can be applied infinitely many times. Observe that the rule $(;)$ is the only rule of RL -dual tableau with that property. Therefore, in order to obtain decision procedures for the classes defined in terms of positive formulas we need to modify the rule for the composition or to restrict its applicability. According to the definition of the classes $RL_{\mathcal{K}}$, for $\mathcal{K} \in \{POS, NEG, NP\}$ given above, we observe that:

- If φ is positive, then none of its proof trees involves any application of the rule $(-;)$,
- If φ is negative, then none of its proof trees involves any application of the rule $(;)$,
- If φ is negative-positive, then in each of its proof trees the applications of the rule $(-;)$ precede the applications of the rule $(;)$.

These observations suggest that dual tableaux for these classes should be constructed from RL -dual tableau as follows:

- RL_{POS} -dual tableau consists of the axiomatic sets and decomposition rules of RL -dual tableau except the rule $(-;)$,
- RL_{NEG} -dual tableau consists of the axiomatic sets and decomposition rules of RL -dual tableau except the rule $(;)$,
- RL_{NP} -dual tableau consists of the axiomatic sets and decomposition rules of RL -dual tableau.

In order to obtain decision procedures for the classes listed above we modify the applicability of the rule $(;)$ as follows. The rule $(;)$ of the form:

$$\frac{x(R;S)y}{xRz, x(R;S)y \mid zSy, x(R;S)y}$$

may be applied to a finite set X of relational formulas if and only if the following conditions are satisfied:

- $x(R; S)y \in X$;
- z occurs in X ;
- The rule (;) introduces a new formula, i.e., $xRz \notin X$ or $zSy \notin X$;
- No other rule is applicable to X .

The second condition implies that there are finitely many possibilities of choosing a variable z . Hence, together with the third condition, they prevent needless expansion of sets of formulas ad infinitum by repeated applications of the rule. The last condition guarantees that the rule (;) can be applied only after the applications of the rule (–;), since all the classes in question do not contain positive–negative formulas.

As in RL-logic, the rules of all dual tableaux considered above guarantee that whenever xRy and $x-Ry$, for an atomic R , belong to a branch of a proof tree, then the branch is closed. Thus the closed branch property holds. Since all the rules are RL-correct, we have:

Proposition 2.12.3. *For every $\mathcal{K} \in \{\text{POS}, \text{NEG}, \text{NP}\}$, all $\text{RL}_{\mathcal{K}}$ -rules are $\text{RL}_{\mathcal{K}}$ -correct.*

Moreover, since formulas of the class RL_{NEG} do not contain positive compositions, the completeness of the reduct of RL-dual tableau for RL_{NEG} can be proved as for the appropriate part of RL-dual tableau.

Theorem 2.12.2 (Soundness and Completeness of RL_{NEG}). *For every RL_{NEG} -formula φ , the following conditions are equivalent:*

1. φ is RL_{NEG} -valid;
2. φ is RL_{NEG} -provable.

By the above theorem and due to the forms of RL_{NEG} -rules, RL_{NEG} -dual tableau is a decision procedure for the class RL_{NEG} .

Now, we aim at proving completeness of RL_{POS} -dual tableau. Let b be a branch of an RL_{POS} -proof tree. Let $\mathbb{O}\mathbb{V}(b)$ be the set of all variables that occur in formulas appearing in the nodes of b . The branch b is said to be complete if it is closed or it satisfies all the completion conditions of RL-dual tableau except $\text{Cpl}(\cdot)$ and, in addition, the following completion condition:

$\text{Cpl}'(\cdot)$: If $x(R; S)y \in b$, then for every $z \in \mathbb{O}\mathbb{V}(b)$, either $xRz \in b$ or $zSy \in b$.

We define the branch structure $\mathcal{M}^b = (U^b, m^b)$ as follows:

- $U^b = \mathbb{O}\mathbb{V}(b)$;
- $m^b(P) = \{(x, y) \in U^b \times U^b : xPy \notin b\}$, for every atomic term P ;
- m^b extends to all the compound relational terms as in RL-models.

Clearly, \mathcal{M}^b is an RL-model. Hence, the branch model property holds. Let v^b be a valuation in \mathcal{M}^b such that $v^b(x) = x$ for every object variable x .

Proposition 2.12.4 (Satisfaction in Branch Model Property). *Let b be an open branch of an RL_{POS} -proof tree. Then, for every RL_{POS} -formula φ , if $\varphi \in b$, then $\mathcal{M}^b, v^b \models \varphi$.*

Proof. The proof is by induction on the complexity of terms. If P is a relational variable and $xPy \in b$, then the proposition holds by the definition of \mathcal{M}^b . If $x-Py \in b$, then $xPy \notin b$, since otherwise, due to the closed branch property, b would be closed. Thus, $(x, y) \in m^b(P)$. Suppose $\mathcal{M}^b, v^b \models x-Py$. Then $(x, y) \notin m^b(P)$, a contradiction.

Now, assume $x(R; S)y \in b$. Then, by the completion condition $\text{Cpl}'(\cdot)$, for every $z \in U^b$, either $xRz \in b$ or $zSy \in b$. By the induction hypothesis, for every $z \in U^b$, either $(x, z) \notin m^b(R)$ or $(z, y) \notin m^b(S)$. Suppose $\mathcal{M}^b, v^b \models x(R; S)y$. Then there exists $z \in U^b$ such that $(x, z) \in m^b(R)$ and $(z, y) \in m^b(S)$, a contradiction.

The proofs for terms of the form $R \cup S$, $R \cap S$, and their complements are similar as in the completeness proof of RL -dual tableau. \square

We conclude:

Theorem 2.12.3 (Soundness and Completeness of RL_{POS}). *For every RL_{POS} -formula φ , the following conditions are equivalent:*

1. φ is RL_{POS} -valid;
2. φ is RL_{POS} -provable.

Note that none of the RL_{POS} -rules introduces a new variable. Therefore, the rule (\cdot) can be applied to a given set of formulas only finitely many times. Hence, RL_{POS} -dual tableau is a decision procedure for the class RL_{POS} .

Completeness of RL_{NP} -dual tableau can be proved in a similar way as completeness of RL_{POS} -dual tableau. Namely, RL_{NP} -completion conditions are RL_{POS} -completion condition and the following:

$\text{Cpl}'(-;)$ If $x-(R; S)y \in b$, then for some $z \in \mathbb{O}\mathbb{V}(b)$, both $xRz \in b$ and $zSy \in b$.

We have:

Theorem 2.12.4 (Soundness and Completeness of RL_{NP}). *For every RL_{NP} -formula φ , the following conditions are equivalent:*

1. φ is RL_{NP} -valid;
2. φ is RL_{NP} -provable.

Recall that RL_{NP} -formulas do not contain positive-negative compositions, which means that each positive composition is in the scope of a negative composition. Although, the rule $(-;)$ introduces a new variable, it is never applied after the application of the rule (\cdot) . Consequently, there are finitely many possibilities of choosing a variable in the application of the rule (\cdot) , and hence the rule (\cdot) can be applied to a given set of formulas only finitely many times. Therefore, we conclude that RL_{NP} -dual tableau is a decision procedure for the class RL_{NP} .