Dual Tableaux: Foundations, Methodology, Case Studies

Springer

Dual Tableaux: Foundations, Methodology, Case Studies

TRENDS IN LOGIC

Studia Logica Library

VOLUME 33

SCOPE OF THE SERIES

Trends in Logic is a bookseries covering essentially the same area as the journal *Studia Logica* – that is, contemporary formal logic and its applications and relations to other disciplines. These include artificial intelligence, informatics, cognitive science, philosophy of science, and the philosophy of language. However, this list is not exhaustive, moreover, the range of applications, comparisons and sources of inspiration is open and evolves over time.

Volume Editor
Ryszard Wójcicki

For further volumes:
http://www.springer.com/series/6645

Ewa Orłowska • Joanna Golińska-Pilarek

Dual Tableaux: Foundations, Methodology, Case Studies

Springer

Prof. Ewa Orłowska
National Institute of
Telecommunications
Szachowa 1
04-894, Warszawa, Poland
orlowska@itl.waw.pl

Joanna Golińska-Pilarek
University of Warsaw
and National Institute
of Telecommunications
Krakowskie Przedmieście 3
00-927, Warszawa, Poland
j.golinska@uw.edu.pl

ISBN 978-94-007-0004-8 e-ISBN 978-94-007-0005-5
DOI 10.1007/978-94-007-0005-5
Springer Dordrecht Heidelberg London New York

Cover design: eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To the Memory of Helena Rasiowa

Preface

The origin of dual tableaux goes back to the paper by Helena Rasiowa and Roman Sikorski 'On the Gentzen theorem' published in Fundamenta Mathematicae in 1960. The authors presented a cut free deduction system for the classical first-order logic without identity. Since then the deduction systems in the Rasiowa–Sikorski style have been constructed for a great variety of theories, ranging from well established non-classical logics such as intuitionistic, modal, relevant, and multiple-valued logics, to important applied theories such as, among others, temporal, in particular interval temporal logics, various logics of programs, fuzzy logics, logics of rough sets, theories of spatial reasoning including region connection calculus, theories of order of magnitude reasoning, and formal concept analysis.

Specific methodological principles of construction of dual tableaux which make possible such a broad applicability of these systems are:

- First, given a theory, a truth preserving translation is defined of the language of the theory into an appropriate language of relations (most often binary);
- Second, a dual tableau is constructed for this relational language so that it provides a deduction system for the original theory.

This methodology, reflecting the paradigm 'Formulas are Relations', enables us to represent within a uniform formalism the three basic components of formal systems: syntax, semantics, and deduction apparatus. The essential observation, leading to a relational formalization of theories, is that a standard relational structure (i.e., a Boolean algebra together with a monoid) constitutes a common core of a great variety of theories. Exhibiting this common core on all the three levels of syntax, semantics and deduction, enables us to create a general framework for representation, investigation and implementation of theories.

The relational approach enables us to build dual tableaux in a systematic, modular way. First, deduction rules are defined for the common relational core of the theories. These rules constitute a basis of all the relational dual tableau proof systems. Next, for any particular theory specific rules are added to the basic set of rules. They reflect the semantic constraints assumed in the models of the theory. As a consequence, we need not implement each deduction system from scratch, we should only extend the basic system with a module corresponding to the specific part of a theory under consideration.

Relational dual tableaux are powerful tools which perform not only verification of validity (i.e., verification of truth of the statements in all the models of a theory) but often they can also be used for proving entailment (i.e., verification that truth of a finite number of statements implies truth of some other statement), model checking (i.e., verification of truth of a statement in a particular fixed model), and satisfaction (i.e., verification that a statement is satisfied by some fixed objects of a model).

Part I of the book is concerned with the two systems which provide a foundation for all of the dual tableau systems presented in this book. In Chap. 1 we recall the original Rasiowa–Sikorski system and we extend it to the system for first-order logic with identity. We discuss relationships of dual tableaux with other deduction systems, namely, tableau systems, Hilbert-style systems, Gentzen-style systems, and resolution. In Chaps. 2 and 3 classical theories of binary relations and their dual tableaux are presented. It is shown how dual tableaux of these theories perform the above mentioned tasks of verification of validity, entailment, model checking, and verification of satisfaction. Some decidable classes of relational formulas are presented in this part together with dual tableau decision procedures.

Part II is concerned with some non-classical theories of relations. In Chap. 4 we present a theory of Peirce algebras and its dual tableau. Peirce algebras provide a means for representation of interactions between binary relations and sets. In Chap. 5 a theory of fork algebras and its dual tableau are presented. Fork algebras are the algebras of binary relations which, together with all the classical relational operations, have a special operation, referred to as fork of relations. While the relational theories of Chap. 2 serve as means of representation for propositional languages, the fork operation enables us a translation of first-order languages into a language of binary relations. In Chap. 6 we present a theory of typed relations and its dual tableau. The theory enables us to represent relations as they are understood in relational databases. The theory deals with relations of various finite arities and, moreover, each relation has its type which is meant to be a representation of a subset of attributes on which the relation is defined.

In Parts III–V relational formalizations of various theories are presented. In Part III relational dual tableaux are constructed for modal (Chap. 7), intuitionistic (Chap. 8), relevant (Chap. 9), and finitely many-valued (Chap. 10) logics.

Part IV is concerned with the major theories of reasoning with incomplete information. In Chaps. 11 and 12 we deal with logics of rough sets and their relational dual tableaux. Chapter 13 presents a relational treatment of formal concept analysis. In Chap. 14 a monoidal t-norm fuzzy logic is considered and a relational dual tableau for this logic is constructed. In this system ternary relations are needed for representation of the monoid product operation. Next, in Chap. 15 theories of order of magnitude reasoning are considered and their dual tableaux are presented.

Part V is concerned with dual tableaux for temporal reasoning, spatial reasoning, and for logics of programs. The first two chapters of that part refer to temporal logics. In Chap. 16 some classical temporal logics are dealt with and in Chap. 17 relational dual tableaux for a class of interval temporal logics are presented. In Chap. 18 dual tableaux for theories of spatial reasoning are constructed, including a system for the region connection calculus. Chapter 19 includes dual tableaux for various versions of propositional dynamic logic and for an event structure logic.

In Part VI we consider some theories for which dual tableau systems are constructed directly within the theory, without translation into any relational theory. In Chap. 20 we present a class of threshold logics where both weights of formulas and thresholds are elements of a commutative group. In Chap. 21 we present a construction of a signed dual tableau which is a decision procedure for a well known intermediate logic. Chapter 22 includes dual tableaux for a class of first-order Post logics. The reduct of this dual tableau for the propositional part of the logic is a decision procedure. Chapter 23 presents a propositional logic endowed with identity treated as a propositional operation and some theories based on this logic. Dual tableaux for all of these theories are presented. In Chap. 24 logics and algebras of conditional decisions are considered together with their dual tableau decision procedures.

The book concludes with Part VII. In the single Chap. 25 of this part we make a synthesis of what we learned in the process of developing dual tableaux in the preceding chapters. We collect observations on how the dual tableaux rules should be designed once the constraints on the models of the theories or definitions of some specific constants are given. We also discuss some useful strategies for construction of dual tableaux proofs.

All the dual tableau systems considered in the book are proved to be sound and complete. We present a general method of proving completeness of dual tableaux which is shown to be broadly applicable to many theories.

Researchers working in any of the theories mentioned in the titles of the chapters will receive in the book a formal tool of specification and verification of those problems in their theories which involve checking validity, satisfaction, or entailment. Every theory whose dual tableau is presented in a chapter of the book is briefly introduced at the beginning of the chapter and a bibliography is indicated where an interested reader could trace developments, major results, and applications of the theory.

To get an idea of what dual tableaux are and how they are related to the other major types of deduction systems, reading Chap. 1 is recommended. After reading the introductory material from Sects. 1.1, ..., 1.4, and Sects. 2.1, ..., 2.8, each chapter in Parts III, IV, and V may be read independently. The material of Chap. 7 may be helpful in reading Chapters 11, 12, 16, 17, and 19, since they are concerned with modal-style logics.

Readers interested in the formal methods of deduction and their application to specification and verification will find in the book an exhaustive exposition and discussion of dual tableaux and their methodology illustrated with several case studies.

Acknowledgments

Special thanks are due to Wendy MacCaull who suggested writing this book and discussed its scope with Ewa Orłowska during her stay as the F. W. James chair professor at St. Francis Xavier University in Antigonish, Canada. Ewa Orłowska is grateful to the colleagues from the RelMiCS (Relational Methods in Computer Science) community and the participants of the COST Action 274 TARSKI (Theory and Applications of Relational Structures as Knowledge Instruments) for cooperation, inspiration, and stimulating discussions on the subject of the book. The authors thank the colleagues who read and commented on some chapters of this book.

Partial supports from the Polish Ministry of Science and Higher Education grant 126 N N206 399134 and from the Spanish Ministry of Science Research Project TIN09-14562-C05-01 are gratefully acknowledged.

Warszawa, 2010 *Ewa Orłowska, Joanna Golińska-Pilarek*

Contents

Part I
Foundations

Chapter 1
Dual Tableau for Classical First-Order Logic

1.1 Introduction

In [RS60] Rasiowa and Sikorski developed a deduction system for classical first-order logic without identity. Their aim was to present a system which is a realization of the Beth idea of the analytic tableau [Bet59] and, in contrast with the Gentzen system [Gen34] which required the cut rule in the proof of completeness, was cut free. In this chapter we present an extension of the dual tableau of Rasiowa and Sikorski to first-order logic with the identity predicate. This deduction system is an implicit foundation of all the dual tableaux presented in this book.

In this chapter the notions and terminology which will be used throughout the book for presentation of dual tableaux is established. In particular, we discuss various types of dual tableaux rules, the notion of correctness of a rule in a proof system, and a form of dual tableaux proofs. We present a detailed proof of completeness of the dual tableau for first-order logic with identity. The main steps of this proof determine a paradigm which will be relevant to all the dual tableaux completeness proofs in the subsequent chapters of the book.

Next, we recall the tableau system for first-order logic introduced in [Smu68] and we discuss how it is related to the Rasiowa and Sikorski system. Following [GPO07b] and some ideas from [SOH04] we show that the two systems are dual to each other. We present a principle of this duality and we show how proofs in one of those systems can be transformed into proofs in the other system. We also discuss a relationship between dual tableaux and Hilbert-style systems, Gentzen-style systems, and resolution. Following [Kon02], we show that the dual tableau may be seen as Gentzen system with the rules where sequents have the empty precedents. We also compare dual tableaux proofs with resolution proofs in a similar way as tableaux and resolution are compared in [OdS93, Sch06]. A section of this chapter is devoted to a discussion of various ways the identity predicate may be treated in dual tableaux. We compare the dual tableaux rules for identity with the corresponding rules from some other deduction systems.

E. Orłowska and J. Golińska-Pilarek, *Dual Tableaux: Foundations, Methodology, Case Studies*, Trends in Logic 33, DOI 10.1007/978-94-007-0005-5_1,

1.2 Classical First-Order Logic with Identity

In this section we recall the language and the semantics of the classical first-order logic with identity. We consider the first-order logic without function symbols. It is known that these symbols are definable in terms of predicate symbols, therefore this is not a severe limitation. Throughout the book, this logic will be denoted by F.

The vocabulary of the logic F consists of the following pairwise disjoint sets of symbols:

- $\mathbb{OV}_{\mathsf{F}}$ – a countable infinite set of individual variables (also referred to as object variables);
- $\mathbb{P}_{\mathsf{F}}$ – a countable set of predicate symbols; we assume that the identity predicate '$=$' belongs to $\mathbb{P}_{\mathsf{F}}$;
- $\{\neg, \wedge, \vee\}$ – the set of propositional operations of negation, conjunction and disjunction, respectively;
- $\{\forall, \exists\}$ – the set of the universal and existential quantifier, respectively.

The set of *atomic formulas* of the logic F is the smallest set such that:

- $x = y$ is an atomic formula for all $x, y \in \mathbb{OV}_{\mathsf{F}}$;
- $P(x_1, \dots, x_k)$ is an atomic formula, for every k-ary predicate $P \in \mathbb{P}_{\mathsf{F}}$, $k \geq 1$, and for all $x_1, \dots, x_k \in \mathbb{OV}_{\mathsf{F}}$.

The set of F-formulas is the smallest set including the set of atomic formulas and closed on propositional operations and quantifiers. Throughout the book, a formula of the form $\neg(x = y)$ will be denoted by $x \neq y$. A *literal* is an atomic formula or a negated atomic formula.

As usual, propositional operations of implication, $\rightarrow$, and equivalence, $\leftrightarrow$, are definable:

For all F-formulas φ and ψ,

$$\varphi \rightarrow \psi \stackrel{\mathrm{df}}{=} \neg\varphi \vee \psi,$$

$$\varphi \leftrightarrow \psi \stackrel{\mathrm{df}}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi).$$

Let φ be an F-formula and let x be an individual variable occurring in φ. A variable x is said to be *free* in φ whenever at least one of its occurrences in φ is not in the scope of any quantifier, and it is said to be *bound* if it is not free. We write $\varphi(x)$ to say that a variable x is free in φ.

An F-*model* is a pair $\mathcal{M} = (U, m)$ satisfying the following conditions:

- U is a non-empty set;
- m is a meaning function assigning relations on U to predicates, i.e., for every k-ary predicate P, $m(P) \subseteq U^k$;
- $m(=)$ is an equivalence relation on U;

- The extensionality property (also referred to as a congruence property) is satisfied: for all $a_i, b_i \in U, i = 1, \ldots, k$, and for every k-ary predicate symbol P, if $(a_1, b_1) \in m(=), \ldots, (a_k, b_k) \in m(=)$, and $(a_1, \ldots, a_k) \in m(P)$, then $(b_1, \ldots, b_k) \in m(P)$.

An F-model is *standard* whenever the meaning of the predicate $=$ is the identity, i.e., $m(=) = \{(a, a) : a \in U\}$.

Let $\mathcal{M}$ be an F-model. A *valuation in* $\mathcal{M}$ is a mapping $v\colon \mathbb{OV}_{\mathsf{F}} \to U$. We write $\mathcal{M}, v \models \varphi$ to denote that φ *is satisfied in* $\mathcal{M}$ *by* v. The relation $\models$ is defined inductively as follows:

- $\mathcal{M}, v \models (x = y)$ iff $(v(x), v(y)) \in m(=)$;
- $\mathcal{M}, v \models P(x_1, \ldots, x_k)$ iff $(v(x_1), \ldots, v(x_k)) \in m(P)$;
- $\mathcal{M}, v \models \neg\varphi$ iff not $\mathcal{M}, v \models \varphi$;
- $\mathcal{M}, v \models \varphi \wedge \psi$ iff $\mathcal{M}, v \models \varphi$ and $\mathcal{M}, v \models \psi$;
- $\mathcal{M}, v \models \varphi \vee \psi$ iff $\mathcal{M}, v \models \varphi$ or $\mathcal{M}, v \models \psi$;
- $\mathcal{M}, v \models \forall x\varphi$ iff for every valuation v' in $\mathcal{M}$ such that v and v' coincide on $\mathbb{OV}_{\mathsf{F}} \setminus \{x\}$, $\mathcal{M}, v' \models \varphi$;
- $\mathcal{M}, v \models \exists x\varphi$ iff for some valuation v' in $\mathcal{M}$ such that v and v' coincide on $\mathbb{OV}_{\mathsf{F}} \setminus \{x\}$, $\mathcal{M}, v' \models \varphi$.

A formula φ is *true in* $\mathcal{M}$ if and only if $\mathcal{M}, v \models \varphi$ for every valuation v in $\mathcal{M}$. An F-formula is F-valid whenever it is true in all F-models. Throughout the book, ‘not $\mathcal{M}, v \models \varphi$’ will be written as $\mathcal{M}, v \not\models \varphi$.

Clearly, F-validity of a formula implies its truth in all standard F-models. The following fact is well known.

Proposition 1.2.1. *For every* F*-model* $\mathcal{M}$ *and for every valuation* v *in* $\mathcal{M}$*, there exist a standard* F*-model* $\mathcal{M}'$ *and a valuation* v' *in* $\mathcal{M}'$ *such that for every* F*-formula* φ*,* $\mathcal{M}, v \models \varphi$ *iff* $\mathcal{M}', v' \models \varphi$*.*

1.3 Rasiowa–Sikorski Proof System for Classical First-Order Logic with Identity

In this section we present the Rasiowa–Sikorski system (RS for short) for the logic F as presented in [RS63] and we expand it with a rule for identity. The rules of RS-system preserve and reflect validity of the sets of formulas, which are their conclusions and premises. Validity of a finite set of formulas is defined as validity of the disjunction of its elements.

The rules of dual tableau for logic F are of the forms:

$$(\text{rule}_1)\quad \frac{\Phi(\overline{x})}{\Phi_0(\overline{x}_0, z)} \qquad\qquad (\text{rule}_2)\quad \frac{\Phi(\overline{x})}{\Phi_0(\overline{x}_0, z) \mid \Phi_1(\overline{x}_1, z)}$$

where $\Phi(\overline{x})$ is a finite set of formulas whose individual variables are among the elements of set$(\overline{x})$, where $\overline{x}$ is a finite sequence of individual variables and set$(\overline{x})$ is the set of elements of sequence $\overline{x}$; every $\Phi_j(\overline{x}_j, z)$, $j = 0, 1$, is a finite non-empty set of formulas, whose individual variables are among the elements of set$(\overline{x}_j) \cup \{z\}$, where z is either instantiated to arbitrary individual variable (usually to the individual variable that appears in the set of formulas to which the rule is being applied) or z must be instantiated to a new variable (not appearing as a free variable in the formulas of the set to which the rule is being applied). A rule of the form (rule$_2$) is a branching rule. In a rule, the set above the line is referred to as its *premise* and the set(s) below the line is (are) its *conclusion(s)*. A rule of the form (rule$_1$) (resp. (rule$_2$)) is said to be *applicable* to a finite set X of formulas whenever $\Phi(\overline{x}) \subseteq X$. As a result of an application of a rule of the form (rule$_1$) (resp. (rule$_2$)) to a set X, we obtain the set $(X \setminus \Phi(\overline{x})) \cup \Phi_0(\overline{x}_0, z)$ (resp. the sets $(X \setminus \Phi(\overline{x})) \cup \Phi_i(\overline{x}_i, z)$, $i \in \{0, 1\}$). As usual, we will write premises and conclusions of the rules as sequences of formulas rather than sets.

Let φ and ψ be F-formulas. RS-dual tableau consists of *decomposition rules* of the following forms:

(RS$\vee$) $$\frac{\varphi \vee \psi}{\varphi, \psi}$$

(RS$\neg\vee$) $$\frac{\neg(\varphi \vee \psi)}{\neg\varphi \mid \neg\psi}$$

(RS$\wedge$) $$\frac{\varphi \wedge \psi}{\varphi \mid \psi}$$

(RS$\neg\wedge$) $$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi, \neg\psi}$$

(RS$\neg$) $$\frac{\neg\neg\varphi}{\varphi}$$

(RS$\forall$) $$\frac{\forall x\varphi(x)}{\varphi(z)}$$

z is a new variable

(RS$\neg\forall$) $$\frac{\neg\forall x\varphi(x)}{\neg\varphi(z), \neg\forall x\varphi(x)}$$

z is any variable

(RS$\exists$) $$\frac{\exists x\varphi(x)}{\varphi(z), \exists x\varphi(x)}$$

z is any variable

(RS$\neg\exists$) $$\frac{\neg\exists x\varphi(x)}{\neg\varphi(z)}$$

z is a new variable

and the *specific rule* of the following form:

(RS $=$) $$\frac{\varphi(x)}{x = z, \varphi(x) \mid \varphi(z), \varphi(x)}$$

where z is any variable, $\varphi(x)$ is an atomic formula, and $\varphi(z)$ is obtained from $\varphi(x)$ by replacing all the occurrences of x in $\varphi(x)$ with z.

A finite set of formulas is *RS-axiomatic* whenever it includes a subset of the form (RSAx1) or (RSAx2):

(RSAx1) $\{x = x\}$, where x is any variable;
(RSAx2) $\{\varphi, \neg\varphi\}$, where φ is any formula.

A finite set of formulas $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$, $n \geq 1$, is said to be an *RS-set* whenever the disjunction of its elements is F-valid. It follows that comma (,) in the rules is interpreted as disjunction.

A rule of the form (rule$_1$) (resp. (rule$_2$)) is *RS-correct* whenever for every finite set X of F-formulas, $X \cup \Phi(\overline{x})$ is an RS-set iff $X \cup \Phi_0(\overline{x}_0, z)$ is an RS-set(resp. $X \cup \Phi_0(\overline{x}_0, z)$ and $X \cup \Phi_1(\overline{x}_1, z)$ are RS-sets). It follows that branching (|) in the rules is interpreted as conjunction. Note that, as mentioned earlier, the definition of correctness establishes preservation and reflection of validity by the rules. It is a characteristic feature of all Rasiowa–Sikorski style deduction systems (see [RS63, GPO07b]). A transfer of validity from the conclusion of a rule to the premise is used for proving soundness of the system and the other direction for proving completeness.

According to the semantics of propositional operations and quantifiers we obtain:

Proposition 1.3.1.

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

Proof. By way of example, we prove the proposition for rules (RS$\forall$), (RS$\exists$), and (RS$=$). Let X be a finite set of F-formulas and let $\varphi(x)$ be an F-formula with a free variable x.

(RS$\forall$) Let z be a variable that does not occur as a free variable in the formulas of the set $X \cup \{\forall x \varphi(x)\}$. Then $X \cup \{\varphi(z)\}$ is an RS-set if and only if for every F-model $\mathcal{M}$ and for every valuation v in $\mathcal{M}$, either there exists $\psi \in X$ such that $\mathcal{M}, v \models \psi$ or for every valuation v' in $\mathcal{M}$ such that v and v' coincide on $\mathbb{OV}_{\mathsf{F}} \setminus \{z\}$, $\mathcal{M}, v' \models \varphi(z)$. The latter is equivalent to F-validity of disjunction of formulas of the set $X \cup \{\forall x \varphi(x)\}$, from which RS-correctness of the rule (RS$\forall$) follows.

(RS$\exists$) Let z be any variable. Clearly, if the premise of the rule is an RS-set, then also the conclusion of the rule is an RS-set. Now, assume $X \cup \{\varphi(z), \exists x \varphi(x)\}$ is an RS-set and suppose $X \cup \{\exists x \varphi(x)\}$ is not an RS-set. Then there exist an F-model $\mathcal{M}$ and a valuation v in $\mathcal{M}$ such that $\mathcal{M}, v \not\models \exists x \varphi(x)$. However, by the assumption, $\mathcal{M}, v \models \varphi(z)$. Let v' be a valuation in $\mathcal{M}$ such that $v'(x) = v(z)$ and for every $y \in \mathbb{OV}_{\mathsf{F}} \setminus \{x\}$, $v'(y) = v(y)$. Thus, $\mathcal{M}, v \models \exists x \varphi(x)$, a contradiction.

(RS$=$) Let $\varphi(x)$ be an atomic formula. Clearly, if $X \cup \{\varphi(x)\}$ is an RS-set, then so are $X \cup \{x = z, \varphi(x)\}$ and $X \cup \{\varphi(z), \varphi(x)\}$. Assume that $X \cup \{x = z, \varphi(x)\}$ and $X \cup \{\varphi(z), \varphi(x)\}$ are RS-sets. Suppose $X \cup \{\varphi(x)\}$ is not an RS-set. Then there exist an F-model $\mathcal{M}$ and a valuation v in $\mathcal{M}$ such that for every formula $\vartheta \in X \cup \{\varphi(x)\}$, $\mathcal{M}, v \not\models \vartheta$. By the assumption, $\mathcal{M}, v \models x = z$ and $\mathcal{M}, v \models \varphi(z)$. Then by the extensionality property $\mathcal{M}, v \models \varphi(x)$, a contradiction. □

Given a formula, successive applications of the rules result in a tree whose nodes consist of finite sets of formulas.

Let φ be an F-formula. An *RS-proof tree for* φ is a tree with the following properties:

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

A branch of an RS-proof tree is said to be *closed* whenever it contains a node with an RS-axiomatic set of formulas. An RS-proof tree is closed whenever all of its branches are closed. Note that every closed branch is finite. A formula φ is *RS-provable* whenever there is a closed RS-proof tree for φ which is then referred to as its *RS-proof*.

From Proposition 1.3.1 we get soundness of RS-system.

Proposition 1.3.2. *If an* F*-formula* φ *is RS-provable, then* φ *is* F*-valid.*

Corollary 1.3.1. *If an* F*-formula* φ *is RS-provable, then* φ *is true in all standard* F*-models.*

As usual in proof theory a concept of completeness of a proof tree is needed. Intuitively, completeness of a tree means that all the rules that can be applied have been applied. By abusing the notation, for a branch b and a formula φ, we write $\varphi \in b$ if φ belongs to the set of formulas of a node of branch b.

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

Cpl(RS$\vee$) (resp. Cpl(RS$\neg\wedge$)) If $(\varphi \vee \psi) \in b$ (resp. $\neg(\varphi \wedge \psi) \in b$), then both $\varphi \in b$ (resp. $\neg\varphi \in b$) and $\psi \in b$ (resp. $\neg\psi \in b$), obtained by an application of the rule (RS$\vee$) (resp. (RS$\neg\wedge$));

Cpl(RS$\wedge$) (resp. Cpl(RS$\neg\vee$)) If $(\varphi \wedge \psi) \in b$ (resp. $\neg(\varphi \vee \psi) \in b$), then either $\varphi \in b$ (resp. $\neg\varphi \in b$) or $\psi \in b$ (resp. $\neg\psi \in b$), obtained by an application of the rule (RS$\wedge$) (resp. (RS$\neg\vee$));

Cpl(RS$\neg$) If $(\neg\neg\varphi) \in b$, then $\varphi \in b$, obtained by an application of the rule (RS$\neg$);

Cpl(RS$\forall$) (resp. Cpl(RS$\neg\exists$)) If $\forall x\varphi(x) \in b$ (resp. $\neg\exists x\varphi(x) \in b$), then for some individual variable z, $\varphi(z) \in b$ (resp. $\neg\varphi(z) \in b$), obtained by an application of the rule (RS$\forall$) (resp. (RS$\neg\exists$));

Cpl(RS$\exists$) (resp. Cpl(RS$\neg\forall$)) If $\exists x\varphi(x) \in b$ (resp. $\neg\forall x\varphi(x) \in b$), then for every individual variable z, $\varphi(z) \in b$ (resp. $\neg\varphi(z) \in b$), obtained by an application of the rule (RS$\exists$) (resp. (RS$\neg\forall$));

Cpl(RS$=$) If $\varphi(x) \in b$ and $\varphi(x)$ is an atomic formula, then for every individual variable z, either $(x = z) \in b$ or $\varphi(z) \in b$, obtained by an application of the rule (RS$=$).

An RS-proof tree is said to be *complete* if and only if all of its branches are complete. A complete non-closed branch is said to be *open*. Note that the rules guarantee that every RS-proof tree can be extended to a complete RS-proof tree. A procedure for constructing a complete proof tree can be found in [DO96]. Observe also that every open branch of an F-proof tree that contains an atomic formula is infinite, since the specific rule (RS=) can be applied infinitely many times to any atomic formula.

Observe that the rules of RS-dual tableau preserve the literals, that is any application of a rule transfers the literals from the premises to the conclusions. Hence, we have:

Fact 1.3.1 (Preservation of literals). *If a node of an RS-proof tree contains a literal, then all of its successors contain this literal as well.*

Proposition 1.3.3. *For any branch of an RS-proof tree, if the literals φ and $\neg\varphi$ belong to the branch, then the branch is closed.*

Proof. Let b be a branch of an RS-proof tree. Fact 1.3.1 implies that if $\varphi \in b$ and $\neg\varphi \in b$, for an atomic formula φ, then eventually both of these formulas appear in a node of branch b. Since the set containing a subset $\{\varphi, \neg\varphi\}$ is F-axiomatic, b is closed. □

Let b be an open branch of an RS-proof tree. We define a *branch structure* $\mathcal{M}^b = (U^b, m^b)$ as follows:

- $U^b = \mathbb{OV}_{\mathsf{F}}$;
- $m^b(P) = \{(x_1, \ldots, x_k) \in (U^b)^k : P(x_1, \ldots, x_k) \notin b\}$, for every k-ary predicate symbol $P \in \mathbb{P}_{\mathsf{F}}, k \geq 1$.

Proposition 1.3.4. *For every open branch b of an RS-proof tree, $\mathcal{M}^b$ is an F-model.*

Proof. First, we show that $m^b(=)$ is an equivalence relation on the set U^b. If for some $x \in \mathbb{OV}_{\mathsf{F}}$, $(x, x) \notin m^b(=)$, then $(x = x) \in b$, which means that b is closed, a contradiction. Let $(x, y) \in m^b(=)$ and suppose $(y, x) \notin m^b(=)$. Then $(x = y) \notin b$ and $(y = x) \in b$. By completion condition Cpl(RS=), either $(x = y) \in b$ or $(y = y) \in b$. In the first case we have a contradiction, in the second case the branch b is closed, which contradicts the assumption. Let $(x, y) \in m^b(=)$ and $(y, z) \in m^b(=)$, which means that $(x = y), (y = z) \notin b$. Suppose $(x, z) \notin m^b(=)$, that is $(x = z) \in b$. By the completion condition Cpl(RS=), either $(x = y) \in b$ or $(y = z) \in b$, a contradiction.

Now, we show that $\mathcal{M}^b$ satisfies the extensionality property. We prove it for $k = 1$. In the general case the proof is similar. Let $(x, y) \in m^b(=)$ and let $x \in m^b(P)$, for some $x, y \in U^b$ and some unary predicate symbol P. Suppose $y \notin m^b(P)$. By the definition of $\mathcal{M}^b$, we obtain $(x = y) \notin b$, $P(x) \notin b$, and $P(y) \in b$. By the completion condition Cpl(RS=), either $(y = x) \in b$ or $P(x) \in b$. Applying once again the completion condition Cpl(RS=) with $\varphi(x)$ being $(y = x)$, we get either $(x = y) \in b$ or $P(x) \in b$, a contradiction. □

Any such model $\mathcal{M}^b$ is referred to as a *branch model*. It is constructed from the syntactic resources of the tree built during the proof search process.

Let $v^b \colon \mathbb{OV}_{\mathsf{F}} \to U^b$ be a valuation in $\mathcal{M}^b$ such that $v^b(x) = x$, for every $x \in \mathbb{OV}_{\mathsf{F}}$.

Proposition 1.3.5. *For every open branch b of an RS-proof tree and for every* F*-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. For atomic formulas the proposition holds by the definitions of $\mathcal{M}^b$ and v^b. If φ is a negated atomic formula, then the proposition follows from the definition of $\mathcal{M}^b$ and Proposition 1.3.3. Assume that the proposition holds for ψ, ϑ, and their negations.

Let $\varphi = \neg\neg\psi$. Assume $\mathcal{M}^b, v^b \models \neg\neg\psi$. Then $\mathcal{M}^b, v^b \models \psi$, hence by the induction hypothesis $\psi \notin b$. Suppose $\neg\neg\psi \in b$. By the completion condition Cpl(RS$\neg$), $\psi \in b$, a contradiction.

Let $\varphi = \forall x\psi(x)$. Assume that $\mathcal{M}^b, v^b \models \forall x\psi(x)$. Then for every $z \in U^b$, $\mathcal{M}^b, v^b \models \psi(z)$, thus by the induction hypothesis, $\psi(z) \notin b$. Suppose $\forall x\psi(x) \in b$. By the completion condition Cpl(RS$\forall$), for some $z \in U^b$, $\psi(z) \in b$, a contradiction.

Let $\varphi = \neg\forall x\psi(x)$. Assume $\mathcal{M}^b, v^b \models \neg\forall x\psi(x)$. Then for some $z \in U^b$, $\mathcal{M}^b, v^b \not\models \psi(z)$. Suppose that $\neg\forall x\psi(x) \in b$. By the completion condition Cpl(RS$\neg\forall$), for every $z \in U^b$, $\neg\psi(z) \in b$. Thus, by the induction hypothesis, $\mathcal{M}^b, v^b \models \psi(z)$, a contradiction.

In the remaining cases the proofs are similar. □

Given a branch model $\mathcal{M}^b$, we define the quotient model $\mathcal{M}^b_q = (U^b_q, m^b_q)$ as follows:

- $U^b_q = \{\|x\| : x \in U^b\}$, where $\|x\|$ is the equivalence class of $m^b(=)$ determined by x;
- $m^b_q(P) = \{(\|x_1\|, \ldots, \|x_k\|) \in (U^b_q)^k : (x_1, \ldots, x_k) \in m^b(P)\}$, for every k-ary predicate symbol P, $k \geq 1$.

Since the branch model satisfies the extensionality property, the definition of $m^b_q(P)$ is correct, i.e., if $(x_1, \ldots, x_k) \in m^b(P)$ and $(x_1, y_1), \ldots, (x_k, y_k) \in m^b(=)$, then $(y_1, \ldots, y_k) \in m^b(P)$.

Let v^b_q be a valuation in $\mathcal{M}^b_q$ such that $v^b_q(x) = \|x\|$, for every $x \in \mathbb{OV}_{\mathsf{F}}$.

Proposition 1.3.6.

1. *The model $\mathcal{M}^b_q$ is a standard* F*-model;*
2. *For every* F*-formula φ, $\mathcal{M}^b, v^b \models \varphi$ iff $\mathcal{M}^b_q, v^b_q \models \varphi$.*

Proof.

1. We have to show that $m^b_q(=)$ is the identity on U^b_q. Indeed, we have:

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

2. The proof is by an easy induction on the complexity of formulas. For example, for the formulas of the form $x = y$ we have: $\mathcal{M}^b, v^b \models (x = y)$ iff $(x, y) \in m^b(=)$ iff $(\|x\|, \|y\|) \in m^b_q(=)$ $\mathcal{M}^b_q, v^b_q \models (x = y)$. □

Proposition 1.3.7. *If a formula φ is true in all standard* F*-models, then φ is RS-provable.*

Proof. Suppose there is no any closed RS-proof tree of φ. Consider a complete RS-proof tree with φ at its root. Let b be an open branch in this tree. Since $\varphi \in b$, by Proposition 1.3.5, $\mathcal{M}^b, v^b \not\models \varphi$. Therefore, by Proposition 1.3.6(2.), we have $\mathcal{M}^b_q, v^b_q \not\models \varphi$. Since $\mathcal{M}^b_q$ is a standard F-model, we get a contradiction. □

In this proof the branch model is constructed from a failed proof search.

Corollary 1.3.2. *If a formula φ is* F*-valid, then φ is RS-provable.*

Summarizing, RS-system provides a deduction tool for the logic F which has the same power as the Hilbert-style axiomatization, namely we have the following theorem which results from Corollaries 1.3.1 and 1.3.2, Propositions 1.3.2 and 1.3.7.

Theorem 1.3.1 (Soundness and Completeness of the RS-system). *Let φ be an* F*-formula. The following conditions are equivalent:*

1. *φ is* F*-valid;*
2. *φ is true in all standard* F*-models;*
3. *φ is RS-provable.*

Example. Consider the following F-formula:

$$\forall x(\varphi \vee \psi(x)) \rightarrow (\varphi \vee \forall x \psi(x)).$$

It can be equivalently presented in the form:

$$\neg\forall x(\varphi \vee \psi(x)) \vee (\varphi \vee \forall x \psi(x)).$$

This formula is F-valid. In Fig. 1.1 its RS-proof is presented.

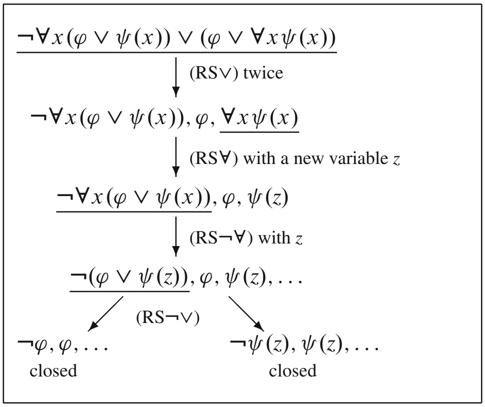

Fig. 1.1 An RS-proof of the formula $\forall x(\varphi \vee \psi(x)) \rightarrow (\varphi \vee \forall x \psi(x))$

Throughout the book, in each node of proof trees presented in the examples we underline the formulas which determine the rule that has been applied during the construction of the tree and we indicate which rule has been applied. If a rule introduces a variable, then we write how the variable has been instantiated. This concerns both the rules which introduce a new or an arbitrary variable. Furthermore, in each node we write only those formulas which are essential for the application of a rule and the succession of these formulas in the node is usually motivated by the reasons of formatting.

1.4 Tableau System for Classical First-Order Logic with Identity

In this section we present a tableau system for the logic F formulated in a way analogous to the formulation of the RS-system. In particular, we indicate explicitly in the rules the repetition of a decomposed formula if needed, in order to make the rules semantically correct. In the original presentation of Smullyan [Smu68] the repetition is shifted to a strategy of building a proof tree. Therefore in our case the Smullyan notation for the rules ($\alpha, \beta, \gamma, \delta$-rules) cannot be applied directly.

The rules of the tableau system preserve and reflect unsatisfiability of the sets of formulas which are their conclusions and premises. There are many versions of tableau systems. They were studied for example in [Fit90]. The specific rule for identity presented here differs from that known in the literature. Such a choice of the rules enables us to see an analogy between tableau and dual tableau treatment of identity (see Sect. 1.8).

Let φ and ψ be any F-formulas. The tableau system for the logic F consists of *decomposition rules* of the following forms:

(T∨) $\dfrac{\varphi \vee \psi}{\varphi \quad | \quad \psi}$ (T¬∨) $\dfrac{\neg(\varphi \vee \psi)}{\neg\varphi, \neg\psi}$

(T∧) $\dfrac{\varphi \wedge \psi}{\varphi, \psi}$ (T¬∧) $\dfrac{\neg(\varphi \wedge \psi)}{\neg\varphi \quad | \quad \neg\psi}$

(T¬) $\dfrac{\neg\neg\varphi}{\varphi}$

(T∀) $\dfrac{\forall x \varphi(x)}{\varphi(z), \forall x \varphi(x)}$ (T¬∀) $\dfrac{\neg\forall x \varphi(x)}{\neg\varphi(z)}$

z is any variable z is a new variable

(T∃) $\dfrac{\exists x \varphi(x)}{\varphi(z)}$ (T¬∃) $\dfrac{\neg\exists x \varphi(x)}{\neg\varphi(z), \neg\exists x \varphi(x)}$

z is a new variable z is any variable and the *specific rule* of the following form:

$$(\mathrm{T}{=})\ \frac{\neg\varphi(x)}{x \neq z, \neg\varphi(x) \quad | \quad \neg\varphi(z), \neg\varphi(x)}$$

where z is any variable, $\varphi(x)$ is an atomic formula, and $\varphi(z)$ is obtained from $\varphi(x)$ by replacing all the occurrences of x in $\varphi(x)$ with z.

A finite set of formulas is *T-axiomatic* whenever it includes a subset of the form (TAx1) or (TAx2):

(TAx1) $\{x \neq x\}$, where x is any variable;
(TAx2) $\{\varphi, \neg\varphi\}$, where φ is any formula.

A finite set of formulas $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ is said to be a *T-set* whenever the conjunction of its elements is unsatisfiable, that is for every F-model $\mathcal{M}$ and for every valuation v in $\mathcal{M}$ there exists $i \in \{1, \ldots, n\}$ such that $\mathcal{M}, v \not\models \varphi_i$. It follows that in this case comma in the rules is interpreted as conjunction.

A rule of the form $\frac{\Phi(\overline{x})}{\Phi_0(\overline{x}_0, z)}$ (resp. $\frac{\Phi(\overline{x})}{\Phi_0(\overline{x}_0, z) \mid \Phi_1(\overline{x}_1, z)}$) is *T-correct* whenever for every finite set X of F-formulas, $X \cup \Phi(\overline{x})$ is a T-set if and only if $X \cup \Phi_0(\overline{x}_0, z)$ is a T-set (resp. $X \cup \Phi_0(\overline{x}_0, z)$ and $X \cup \Phi_1(\overline{x}_1, z)$ are T-sets). That is branching in the rules is interpreted as disjunction. Thus T-rules preserve and reflect unsatisfiability of the sets of formulas. The classical tableau system for first-order logic presented in [Smu68] has also the property of preserving and reflecting unsatisfiability. Although this fact is not provable directly from the definition of the classical tableau rules, it can be proved under the additional assumptions on repetition of some formulas in the process of application of the rules. In the classical tableau system this assumption is hidden, it is shifted to a strategy of building the proof trees. In our T-system the required repetitions are explicitly indicated in the rules.

It is easy to show that all the rules of T-system for the logic F are T-correct, and all its axiomatic sets are T-sets. These facts follow from the semantics of the propositional operations and quantifiers as in the case of the RS-system.

A proof in the T-system has the form of a finitely branching tree whose nodes are finite sets of formulas. Let φ be an F-formula. A *T-proof tree for* φ is a tree with the following properties:

- The formula $\neg\varphi$ is at the root of this tree;
- Each node except the root is obtained by the application of a T-rule to its predecessor node;
- A node does not have successors whenever its set of formulas is a T-axiomatic set or none of the rules is applicable to its set of formulas.

A branch of a T-proof tree is said to be *closed* whenever it contains a node with a T-axiomatic set of formulas. A T-proof tree is closed whenever all of its branches are closed. A formula φ is *T-provable* whenever there is a T-closed proof tree for φ which is then referred to as its *T-proof*.

Completion conditions and the branch model are defined in a similar way as in the RS-proof system. For instance, the completion conditions determined by the rules (T$\vee$), (T$\neg\vee$), (T$\forall$), and (T$\neg\forall$) are:

Cpl(T$\vee$) If $\varphi \vee \psi \in b$, then either $\varphi \in b$ or $\psi \in b$;
Cpl(T$\neg\vee$) If $\neg(\varphi \vee \psi) \in b$, then both $\neg\varphi \in b$ and $\neg\psi \in b$;
Cpl(T$\forall$) If $\forall x \varphi(x) \in b$, then for every individual variable z, $\varphi(z) \in b$;
Cpl(T$\neg\forall$) If $\neg\forall x \varphi(x) \in b$, then for some individual variable z, $\neg\varphi(z) \in b$.

Given an open branch b of a T-proof tree, we define a branch structure $\mathcal{M}^b = (U^b, m^b)$ as follows:

- $U^b = \mathbb{OV}_{\mathsf{F}}$;
- $m^b(P) = \{(x_1, \ldots, x_k) \in (U^b)^k : \neg P(x_1, \ldots, x_k) \in b\}$, for every k-ary predicate symbol $P \in \mathbb{P}_{\mathsf{F}}, k \geq 1$.

In a similar way as in RS-dual tableau, the following can be proved:

Proposition 1.4.1. *For every open branch b of a T-proof tree, $\mathcal{M}^b$ is an* F*-model.*

Proposition 1.4.2. *For every open branch b of a T-proof tree and for every* F*-formula φ, if $\mathcal{M}^b, v^b \models \varphi$, then $\neg\varphi \notin b$.*

The proof of soundness and completeness of the tableau proof system is based on the same idea as in the RS-proof system. Then, we have:

Theorem 1.4.1 (Soundness and Completeness of the T-system). *Let φ be an* F*-formula. Then the following conditions are equivalent:*

1. *φ is* F*-valid;*
2. *φ is true in all standard* F*-models;*
3. *φ is T-provable.*

1.5 Quasi Proof Trees

Let P $\in$ {RS, T} be one of the proof systems. Our aim is to define a transformation of a proof tree in one of the systems into a proof tree in the other system. For that purpose it is useful to modify the concept of a proof tree by defining a quasi proof tree. A quasi proof tree is in fact a proof tree modulo the double negation rule.

An F-formula is said to be positive whenever negation is not its principal operation. Let $n \geq 0$ and let φ be a positive F-formula. We define:

$$\neg^0 \varphi \stackrel{\text{df}}{=} \varphi;$$
$$\neg^{n+1} \varphi \stackrel{\text{df}}{=} \neg(\neg^n \varphi).$$

We define the rules $(\mathrm{P}\neg)^*$:

$$(\mathrm{P}\neg)^* \quad \frac{\neg^n \varphi}{\neg^{n \bmod 2} \varphi}$$

where $n \geq 0$ and φ is a positive formula.

As usual, this rule is applicable to a set X of formulas whenever $\neg^n \varphi \in X$ for some $n \geq 0$ and for a positive formula φ. Its application to a set X may be seen as the iteration of applications of rule $(\mathrm{P}\neg)$.

Let $\# \in \{\vee, \neg\vee, \wedge, \neg\wedge, \forall, \neg\forall, \exists, \neg\exists, =\}$. Let $(\mathrm{P}\#\neg^*)$ be a rule defined as a composition of the rules $(\mathrm{P}\#)$ and $(\mathrm{P}\neg)^*$ treated as maps on the family of finite subsets of formulas and returning a finite subset of formulas (or a pair of subsets in case $(\mathrm{P}\#)$ is a branching rule).

$$(\mathrm{P}\#\neg^*) \stackrel{\mathrm{df}}{=} (\mathrm{P}\neg)^* \circ (\mathrm{P}\#)$$

This rule is applicable to a set X of formulas whenever the rule $(\mathrm{P}\#)$ is applicable to X. Let X_0 (resp. X_0 and X_1 if $(\mathrm{P}\#)$ is a branching rule) be the set(s) obtained from X by an application of rule $(\mathrm{P}\#)$. Given a finite set Z of formulas, by $Z^{\bmod 2}$ we mean the set of formulas obtained from Z by replacing every formula of the form $\neg^l \varphi$, where $l \geq 0$ and φ is a positive formula, by the formula $\neg^{l \bmod 2} \varphi$. Then the result of application of rule $(\mathrm{P}\#\neg^*)$ to X is the set ${X_0}^{\bmod 2}$ (resp. ${X_0}^{\bmod 2}$ and ${X_1}^{\bmod 2}$ if $(\mathrm{P}\#)$ is a branching rule), where X_0 (resp. X_0 and X_1) is (are) the result(s) of application of rule $(\mathrm{P}\#)$ to X.

Let $\neg^n \varphi$ be an F-formula, where $n \geq 0$ and φ is a positive formula. A *P-quasi proof tree for* $\neg^n \varphi$ is a tree with the following properties:

- Its root consists of the formula ψ, where:

$$\psi = \begin{cases} \neg^{n \bmod 2} \varphi, & \text{if P=RS,} \\ \neg^{(n+1) \bmod 2} \varphi, & \text{if P=T;} \end{cases}$$

- Each node except the root is obtained by the application of a rule $(\mathrm{P}\#\neg^*)$ to its predecessor node;
- A node does not have successors if its set of formulas is a P-axiomatic set or none of the rules is applicable to its set of formulas.

An example of an RS-quasi proof tree is presented in Fig. 1.2, while Fig. 1.3 presents a T-quasi proof tree for the same formula. Observe that in a diagram of Fig. 1.2, after applying the rule $(\mathrm{RS}\neg\exists)$ to the set $Z_1 = \{\neg\exists x \exists y \neg(x \neq y \vee y = x)\}$ we obtain the set $\{\neg\exists y \neg(x_1 \neq y \vee y = x_1)\}$ to which the rule $(\mathrm{RS}\neg)^*$ is applied with $n = 1$. Thus, the application of the rule $(\mathrm{RS}\neg\exists\neg^*)$ to Z_1 results in Z_2. Then, we apply the rule $(\mathrm{RS}\neg\exists)$ to Z_2, so that we obtain the set $\{\neg\neg(x_1 \neq x_2 \vee x_2 = x_1)\}$ to which we apply the rule $(\mathrm{RS}\neg)^*$. Since $\{\neg\neg(x_1 \neq x_2 \vee x_2 = x_1)\}^{\bmod 2} = Z_3$, the application of the rule $(\mathrm{RS}\neg\exists\neg^*)$ to Z_2 results in Z_3. The application of rule $(\mathrm{RS}\vee)$ to Z_3 results in Z_4 such that $Z_4^{\bmod 2} = Z_4$. Therefore, Z_4 is the result of application of the rule $(\mathrm{RS}\vee\neg^*)$ to Z_3. Similarly, the application of rule $(\mathrm{RS}{=})$ to Z_4 results in