

Chapter 8

Dual Tableaux for Some Logics Based on Intuitionism

8.1 Introduction

Intuitionistic logic encompasses the principles of reasoning which were used by L. Brouwer in developing his intuitionistic mathematics, beginning in [Bro07]. Intuitionistic logic can be succinctly described as classical logic without the law $\varphi \vee \neg\varphi$ of excluded middle. Brouwer observed that this law was abstracted from finite situations and its application to statements about infinite collections is not justified. One of the consequences of the rejection of the law of excluded middle is that every intuitionistic proof of an existential sentence can be effectively transformed into an intuitionistic proof of an instance of that sentence. More precisely, if a formula of the form $\exists x\varphi(x)$ without free variables is provable in the intuitionistic predicate logic, then there is a term t without free variables such that $\varphi(t)$ is provable. In particular, if $\varphi \vee \psi$ is provable, then either φ or ψ is provable. In that sense intuitionistic logic may provide a logical basis for constructive reasoning. A formal system of intuitionistic logic was proposed in [Hey30]. A relationship between the classical propositional logic PC and intuitionistic propositional logic INT was proved by Glivenko in [Gli29], namely a PC-formula φ is PC-provable if and only if $\neg\neg\varphi$ is INT-provable. Kripke semantics for intuitionistic logic was developed in [Kri65].

In this chapter we present relational dual tableaux for the propositional intuitionistic logic INT, for the minimal intuitionistic logic of Johansson, and for some axiomatic extensions of INT, namely for the Scott's logic and the logic of weak excluded middle. All these logics are decidable. We also develop a dual tableau for the logic proposed in [FM95] for hardware verification. It is an intuitionistic logic endowed with a propositional operation which enables us to specify the propagation of signals along the gates of combinatorial circuits. Dual tableaux presented in this chapter are modifications of the deduction systems presented in [FO95] within the context of relational logics of Chap. 2.

8.2 Relational Formalization of Intuitionistic Logic

In this section we present the relational formalization of an intuitionistic logic, INT. The vocabulary of INT-language consists of symbols from the following pairwise disjoint sets:

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

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

An INT-*model* is a structure $\mathcal{M} = (U, R, m)$ such that U is a non-empty set of states, R is a reflexive and transitive relation on U , and m is a meaning function such that the following conditions are satisfied:

- $m(p) \subseteq U$, for every propositional variable $p \in \mathbb{V}$;
- For all $s, s' \in U$ the following heredity condition holds:

$$(\text{her}) \text{ If } (s, s') \in R \text{ and } s \in m(p), \text{ then } s' \in m(p).$$

Let $\mathcal{M} = (U, R, m)$ be an INT-model and let $s \in U$. Satisfaction of INT-formula φ in \mathcal{M} by state s , $\mathcal{M}, s \models \varphi$ for short, is defined inductively as follows:

- $\mathcal{M}, s \models p$ iff $s \in m(p)$, for any propositional variable p ;
- $\mathcal{M}, s \models \neg\varphi$ iff for every $s' \in U$, if $(s, s') \in R$, then $\mathcal{M}, s' \not\models \varphi$;
- $\mathcal{M}, s \models \varphi \vee \psi$ iff $\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$;
- $\mathcal{M}, s \models \varphi \wedge \psi$ iff $\mathcal{M}, s \models \varphi$ and $\mathcal{M}, s \models \psi$;
- $\mathcal{M}, s \models (\varphi \rightarrow \psi)$ iff for every $s' \in U$, if $(s, s') \in R$ and $\mathcal{M}, s' \models \varphi$, then $\mathcal{M}, s' \models \psi$.

It is known that the heredity condition holds for every formula φ , i.e., if $(s, s') \in R$ and $\mathcal{M}, s \models \varphi$, then $\mathcal{M}, s' \models \varphi$.

An INT-formula φ is said to be *true* in an INT-model $\mathcal{M} = (U, R, m)$, $\mathcal{M} \models \varphi$, whenever for every $s \in U$, $\mathcal{M}, s \models \varphi$, and it is said to be INT-*valid* whenever it is true in all INT-models.

In the relational formalization of logic INT we follow the main ideas of the method presented in Sect. 7.4. The vocabulary of the language of the relational logic \mathbf{RL}_{INT} consists of the symbols from the following pairwise disjoint sets:

- $\mathbb{O}\mathbb{V}_{\mathbf{RL}_{\text{INT}}}$ – a countable infinite set of object variables;
- $\mathbb{R}\mathbb{V}_{\mathbf{RL}_{\text{INT}}}$ – a countable infinite set of relational variables;
- $\mathbb{R}\mathbb{C}_{\mathbf{RL}_{\text{INT}}} = \{1, 1', R\}$ – the set of relational constants;
- $\{-, \cup, \cap, ;, ^{-1}\}$ – the set of relational operations.

As in case of modal logics of Sect. 7.4, the constant R represents the accessibility relation from INT-models. Relational terms and formulas of the logic \mathbf{RL}_{INT} are defined as described in Sect. 2.3. The relational variables are intended to represent intuitionistic formulas.

An \mathbf{RL}_{INT} -model is an $\mathbf{RL}(1, 1')$ -model $\mathcal{M} = (U, R, m)$ such that the following conditions are satisfied:

- $m(P)$ is a right ideal relation on U , for every $P \in \mathbb{R}\mathbb{V}_{\mathbf{RL}_{\text{INT}}}$;
- $R = m(R)$ is a reflexive and transitive relation on U such that for every $P \in \mathbb{R}\mathbb{V}_{\mathbf{RL}_{\text{INT}}}$ and for all $x, y, z \in U$, the following holds:

(her') If $(x, y) \in R$ and $(x, z) \in m(P)$, then $(y, z) \in m(P)$.

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

- $\tau(p) = \tau'(p)$, for any propositional variable p ;
- $\tau(\neg\varphi) = -(R; \tau(\varphi))$;
- $\tau(\varphi \vee \psi) = \tau(\varphi) \cup \tau(\psi)$;
- $\tau(\varphi \wedge \psi) = \tau(\varphi) \cap \tau(\psi)$;
- $\tau(\varphi \rightarrow \psi) = -(R; (\tau(\varphi) \cap \neg\tau(\psi)))$.

The translation of atomic INT-formulas cannot be defined as in modal logics (see Sect. 7.4) due to the assumption of heredity. Therefore, in the \mathbf{RL}_{INT} -models relational variables are assumed to be right ideal relations and in the \mathbf{RL}_{INT} -dual tableau the rule (ideal) reflecting this condition is included.

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

Proposition 8.2.1. *Let φ be an INT-formula. Then for every INT-model $\mathcal{M} = (U, R, m)$ there exists an \mathbf{RL}_{INT} -model $\mathcal{M}' = (U, R, m')$ with the same universe and the same relation R as those in \mathcal{M} such that for all $s, s' \in U$, $\mathcal{M}, s \models \varphi$ iff $(s, s') \in m'(\tau(\varphi))$.*

Proof. The proof is similar to the proof of Proposition 7.4.2. Let φ be an INT-formula and let $\mathcal{M} = (U, R, m)$ be an INT-model. We define an \mathbf{RL}_{INT} -model $\mathcal{M}' = (U, R, m')$ as follows:

- $m'(1) = U \times U$;
- $m'(1')$ is the identity on U ;
- $m'(\tau(p)) = \{(x, y) \in U \times U : x \in m(p)\}$, for every propositional variable p ;
- $m'(R) = R$;
- m' extends to all the compound terms as in $\mathbf{RL}(1, 1')$ -models.

The proof is by induction on the complexity of formulas. By way of example, we show the proposition for a compound formula built with the implication operation which has a non-classical semantics in INT-logic.

Let $\varphi = \psi \rightarrow \vartheta$ and let $s' \in U$. Then $\mathcal{M}, s \models \psi \rightarrow \vartheta$ iff for every $t \in U$, if $(s, t) \in R$ and $\mathcal{M}, t \models \psi$, then $\mathcal{M}, t \models \vartheta$ iff, by the induction hypothesis, for every $t \in U$, if $(s, t) \in R$ and $(t, s') \in m'(\tau(\psi))$, then $(t, s') \in m'(\tau(\vartheta))$ iff $(s, s') \in m'(-(R; (\tau(\psi) \cap \neg\tau(\vartheta))))$ iff $(s, s') \in m'(\tau(\varphi))$. \square

Proposition 8.2.2. *Let φ be an INT-formula. Then for every standard \mathbf{RL}_{INT} -model $\mathcal{M}' = (U, R, m')$ there exists an INT-model $\mathcal{M} = (U, R, m)$ with the same universe and the same relation R as those in \mathcal{M}' such that for all $s, s' \in U$, the condition of Proposition 8.2.1 holds.*

Proof. Let φ be an INT-formula, let $\mathcal{M}' = (U, R, m')$ be a standard \mathbf{RL}_{INT} -model. We define an INT-model $\mathcal{M} = (U, R, m)$ as follows:

$$m(p) = \{x \in U : \text{for some } y \in U, (x, y) \in m'(\tau(p))\}, \text{ for every propositional variable } p.$$

Now, the proposition can be proved as Proposition 8.2.1. \square

Due to the above propositions we can prove the following (see also Propositions 7.4.4 and 7.4.5):

Proposition 8.2.3. *Let φ be an INT-formula. Then for every INT-model \mathcal{M} there exists an \mathbf{RL}_{INT} -model \mathcal{M}' such that for any object variables x and y , $\mathcal{M} \models \varphi$ iff $\mathcal{M}' \models x\tau(\varphi)y$.*

Proposition 8.2.4. *Let φ be an INT-formula. Then for every standard \mathbf{RL}_{INT} -model \mathcal{M}' there exists an INT-model \mathcal{M} such that for any object variables x and y , the condition of Proposition 8.2.3 holds.*

Finally, from Propositions 8.2.3 and 8.2.4, we get:

Theorem 8.2.1. *For every INT-formula φ and for all object variables x and y , the following conditions are equivalent:*

1. φ is INT-valid;
2. $x\tau(\varphi)y$ is \mathbf{RL}_{INT} -valid.

A dual tableau for the logic \mathbf{RL}_{INT} consists of the axiomatic sets and the rules of dual tableau for $\mathbf{RL}(1, 1')$ -logic adjusted to the language of \mathbf{RL}_{INT} , the rules (ref R) and (tran R) presented in Sects. 6.6 (see also Sect. 7.4) that reflect reflexivity and transitivity of relation R from INT-models, respectively. Furthermore, we have the rules of the following forms:

For every relational variable P and for all object variables x and y ,

$$\text{(rher')} \frac{xPy}{zRx, xPy \mid zPy, xPy} \quad z \text{ is any object variable}$$

$$\text{(ideal)} \frac{xPy}{xPz, xPy} \quad z \text{ is any object variable}$$

The rule (rher') reflects the heredity condition, while the rule (ideal) reflects the fact that every relational variable is interpreted in an \mathbf{RL}_{INT} -model as a right ideal relation. Note that any application of the rules of \mathbf{RL}_{INT} -dual tableau, in particular an application of the specific rules listed above, preserves the formulas built with atomic terms or their complements, and hence the closed branch property holds.

The notions of an \mathbf{RL}_{INT} -set of formulas and correctness of a rule are defined as in Sect. 2.4.

Proposition 8.2.5. *The specific RL_{INT} -rules are RL_{INT} -correct.*

Proof. By way of example, we show correctness of the rule (rher'). Correctness of the remaining rules can be proved as in standard modal logics. Let X be a finite set of RL_{INT} -formulas. Clearly, if $X \cup \{xPy\}$ is an RL_{INT} -set, then so are $X \cup \{zRx, xPy\}$ and $X \cup \{zPy, xPy\}$. Now, assume that $X \cup \{zRx, xPy\}$ and $X \cup \{zPy, xPy\}$ are RL_{INT} -sets and suppose $X \cup \{xPy\}$ is not an RL_{INT} -set. Then there exist an RL_{INT} -model $\mathcal{M} = (U, R, m)$ and a valuation v in \mathcal{M} such that for every $\varphi \in X \cup \{xPy\}$ $\mathcal{M}, v \models \varphi$, which implies $(v(x), v(y)) \notin m(P)$. By the assumption, $\mathcal{M}, v \models zRx$ and $\mathcal{M}, v \models zPy$. Thus, $(v(z), v(x)) \in R$ and $(v(z), v(y)) \in m(P)$. Then, by the condition (her'), $(v(x), v(y)) \in m(P)$, a contradiction. \square

It is easy to see that all the remaining rules of RL_{INT} -dual tableau are RL_{INT} -correct and all the RL_{INT} -axiomatic sets are RL_{INT} -sets.

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

A branch b of an RL_{INT} -proof tree is complete whenever it is either closed or it satisfies the completion conditions of $\text{RL}(1, 1')$ -dual tableau and the following completion conditions determined by the rules specific for the RL_{INT} -dual tableau:

For every relational variable P and for all object variables x and y ,

Cpl(rher') If $xPy \in b$, then for every object variable z , either $zRx \in b$ or $zPy \in b$, obtained by an application of the rule (rher');

Cpl(ideal) If $xPy \in b$, then for every object variable z , $xPz \in b$, obtained by an application of the rule (ideal).

The notions of a complete RL_{INT} -proof tree and an open branch of an RL_{INT} -proof tree are defined as in RL -logic (see Sect. 2.5).

The branch structure $\mathcal{M}^b = (U^b, R^b, m^b)$ is defined in a standard way, that is $m^b(Q) = \{(x, y) \in U^b \times U^b : xQy \notin b\}$, for every $Q \in \mathbb{R}\mathbb{V}_{\text{RL}_{\text{INT}}} \cup \{R\}$, and $R^b = m^b(R)$. The completion conditions enable us to show that R^b satisfies all the conditions assumed in RL_{INT} -models. Therefore, the following holds:

Proposition 8.2.6 (Branch Model Property). *For every open branch b of an RL_{INT} -proof tree, \mathcal{M}^b is an RL_{INT} -model.*

Proof. We show that \mathcal{M}^b satisfies the condition (her') and every relational variable P is interpreted in \mathcal{M}^b as a right ideal relation. Assume that $(x, y) \in R^b$ and $(x, z) \in m^b(P)$, that is $xRy \notin b$ and $xPz \notin b$. Suppose $(y, z) \notin m^b(P)$, that is $yPz \in b$. Then, by the completion condition Cpl(rher'), either $xRy \in b$ or $xPz \in b$, a contradiction. Hence, \mathcal{M}^b satisfies the condition (her'). Now, assume that $(x, y) \in m^b(P)$, that is $xPy \notin b$. Suppose that for some $z \in U^b$, $(x, z) \notin m^b(P)$. Then $xPz \in b$ and by the completion condition Cpl(ideal), $xPy \in b$, a contradiction. Therefore, for every relational variable P , $m^b(P)$ is a right ideal relation on U^b . \square

Since branch model \mathcal{M}^b is defined in a standard way, the satisfaction in branch model property can be proved as in $\text{RL}(1, 1')$ -logic (see Sects. 2.5 and 2.7).

Proposition 8.2.7 (Satisfaction in Branch Model Property). *For every open branch b of an RL_{INT} -proof tree and for every RL_{INT} -formula φ , if $\mathcal{M}^b, v^b \models \varphi$, then $\varphi \notin b$.*

Hence, the completeness of RL_{INT} -dual tableau follows.

Theorem 8.2.2 (Soundness and Completeness of RL_{INT}). *Let φ be an RL_{INT} -formula. Then the following conditions are equivalent:*

1. φ is RL_{INT} -valid;
2. φ is true in all standard RL_{INT} -models;
3. φ is RL_{INT} -provable.

By the above theorem and Theorem 8.2.1, we obtain:

Theorem 8.2.3 (Relational Soundness and Completeness of INT). *Let φ be an INT-formula. Then for all object variables x and y , the following conditions are equivalent:*

1. φ is INT-valid;
2. $x(\tau(\varphi))y$ is RL_{INT} -provable.

Example. Consider the following INT-formula φ :

$$\varphi = \neg p \rightarrow (p \rightarrow t).$$

The translation of this formula into RL_{INT} -term is:

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

where $\tau(p) = P$ and $\tau(t) = T$. INT-validity of φ is equivalent to RL_{INT} -provability of the formula $x\tau(\varphi)y$. Figure 8.1 presents its RL_{INT} -proof.

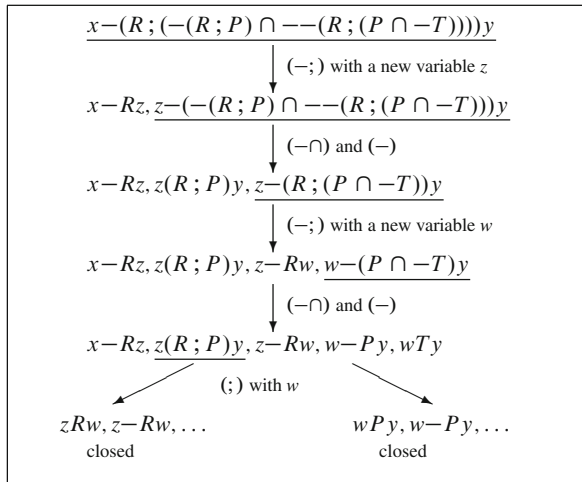


Fig. 8.1 An RL_{INT} -proof of the formula $\neg p \rightarrow (p \rightarrow t)$

8.3 Relational Formalization of Minimal Intuitionistic Logic

Minimal intuitionistic logic **J** was introduced by Johansson in [Joh36]. It differs from the intuitionistic logic **INT** in that the formula $\neg\varphi \rightarrow (\varphi \rightarrow \psi)$ is not a theorem of the logic. The language of the logic **J** is the same as **INT**-language. However, these logics differ in semantics. A **J**-model is a structure $\mathcal{M} = (U, R, Q, m)$ such that the following conditions are satisfied:

- (U, R, m) is an **INT**-model;
- $Q \subseteq U$ is such that for every $s \in U$, if $s \in Q$ and $(s, s') \in R$, then $s' \in Q$.

The satisfaction of a **J**-formula φ in a **J**-model \mathcal{M} by a state $s \in U$ is defined as in **INT**-logic except for the clause for negated formulas:

$$\mathcal{M}, s \models \neg\psi \text{ iff for all } s' \in U, \text{ if } (s, s') \in R, \text{ then } \mathcal{M}, s' \not\models \psi \text{ or } s' \in Q.$$

Intuitively, Q is thought of as the set of those states which are inconsistent.

The language of the relational logic \mathbf{RL}_J is an $\mathbf{RL}_{\mathbf{INT}}$ -language with the following set of relational constants: $\mathbb{RC}_{\mathbf{RL}_J} = \{1, 1', R, Q\}$.

An \mathbf{RL}_J -model is an $\mathbf{RL}_{\mathbf{INT}}$ -model $\mathcal{M} = (U, R, Q, m)$ such that the following conditions are satisfied:

- (Q1) Q is a right ideal relation on U ;
 (Q2) If $(x, z) \in Q$ and $(x, y) \in R$, then $(y, z) \in Q$, for all $x, y, z \in U$.

The translation τ from **J**-formulas into relational terms coincides with the translation of **INT**-formulas defined in Sect. 8.2 except for the translation of negated formulas:

$$\tau(\neg\varphi) \stackrel{\text{df}}{=} \neg(R; (\tau(\varphi) \cap -Q)).$$

Following the method of proving Theorem 8.2.1, we can prove:

Theorem 8.3.1. *For every **J**-formula φ and for all object variables x and y , the following conditions are equivalent:*

1. φ is **J**-valid;
2. $x\tau(\varphi)y$ is \mathbf{RL}_J -valid.

A dual tableau for the logic \mathbf{RL}_J consists of the axiomatic sets and the rules of dual tableau for $\mathbf{RL}_{\mathbf{INT}}$ -logic presented in Sect. 8.2 adjusted to the language of \mathbf{RL}_J and in addition the rules of the following forms:

For all object variables x and y ,

$$(rQ1) \frac{xQy}{xQz, xQy} \quad z \text{ is any object variable}$$

$$(rQ2) \frac{xQy}{zQy, xQy \mid zRx, xQy} \quad z \text{ is any object variable}$$

The above rules reflect the properties of relation Q assumed in RL_J -models. As in RL_{INT} -dual tableau, any application of the rules listed above preserves the formulas built with atomic terms or their complements, and hence the closed branch property holds.

Proposition 8.3.1. *The RL_J -rules are RL_J -correct.*

Proof. By way of example, we prove correctness of the rule (rQ2). Correctness of the other rules can be proved as in standard modal logics. Let X be any finite set of RL_J -formulas. Clearly, if $X \cup \{xQy\}$ is an RL_J -set, then so are $X \cup \{zQy, xQy\}$ and $X \cup \{zRx, xQy\}$. Now, assume that $X \cup \{zQy, xQy\}$ and $X \cup \{zRx, xQy\}$ are RL_J -sets. Suppose $X \cup \{xQy\}$ is not an RL_J -set. Then there exist an RL_J -model $\mathcal{M} = (U, R, Q, m)$ and a valuation v in \mathcal{M} such that for every $\varphi \in X \cup \{xQy\}$ $\mathcal{M}, v \models \varphi$, which implies $(v(x), v(y)) \notin m(Q)$. By the assumption, $\mathcal{M}, v \models zQy$ and $\mathcal{M}, v \models zRx$. Thus, $(v(z), v(y)) \in Q$ and $(v(z), v(x)) \in R$. By property (Q2) assumed in RL_J -models, $(v(x), v(y)) \in Q$, a contradiction. Therefore, the rule (rQ2) is RL_J -correct. \square

The completion conditions determined by the rules (rQ1) and (rQ2) are:

For all object variables x and y ,

Cpl(rQ1) If $xQy \in b$, then for every object variable z , $xQz \in b$;

Cpl(rQ2) If $xQy \in b$, then for every object variable z , either $zQy \in b$ or $zRx \in b$.

The branch model is defined in a standard way, i.e., $m^b(T) = \{(x, y) \in U^b \times U^b : xTy \notin b\}$, for every $T \in \mathbb{R}V_{RL_J} \cup \{R, Q\}$. The relation Q^b satisfies all the conditions assumed in RL_J -models, therefore we have:

Proposition 8.3.2 (Branch Model Property). *For every open branch b of an RL_J -proof tree, \mathcal{M}^b is an RL_J -model.*

Proof. We show that Q^b is well defined. It can be easily proved that Q^b is a right ideal relation on U^b . Now, assume that $(x, z) \in Q^b$ and $(x, y) \in R^b$, that is $xQz \notin b$ and $xRy \notin b$. Suppose $(y, z) \notin Q^b$, that is $yQz \in b$. Then, by the completion condition Cpl(rQ2), either $xQz \in b$ or $xRy \in b$, a contradiction. \square

Satisfaction in branch model property can be proved as in $RL(1, 1')$ -logic, hence we get:

Theorem 8.3.2 (Soundness and Completeness of RL_J). *Let φ be an RL_J -formula. Then the following conditions are equivalent:*

1. φ is RL_J -valid;
2. φ is true in all standard RL_J -models;
3. φ is RL_J -provable.

By the above theorem and Theorem 8.3.1, we have:

Theorem 8.3.3 (Relational Soundness and Completeness of J). *Let φ be a J-formula. Then for all object variables x and y , the following conditions are equivalent:*

1. φ is J-valid;
2. $x(\tau(\varphi))y$ is RL_J -provable.

Example. Consider the following J-formula φ :

$$\varphi = p \rightarrow \neg\neg p.$$

The translation of φ into an RL_J -term is:

$$\tau(\varphi) = -(R; (P \cap \neg\neg(R; (\neg(R; (P \cap \neg Q)) \cap \neg Q))))),$$

where $\tau(p) = P$. J-validity of φ is equivalent to RL_J -provability of the formula $x\tau(\varphi)y$. Figure 8.2 presents its RL_J -proof.

In the previous section we constructed an RL_{INT} -proof for the formula $\varphi = \neg p \rightarrow (p \rightarrow t)$ which showed its INT-validity. This formula is not J-valid. Its translation into RL_J -term is:

$$\tau(\varphi) = -(R; (\neg(R; (P \cap \neg Q)) \cap \neg\neg(R; (P \cap \neg T))))),$$

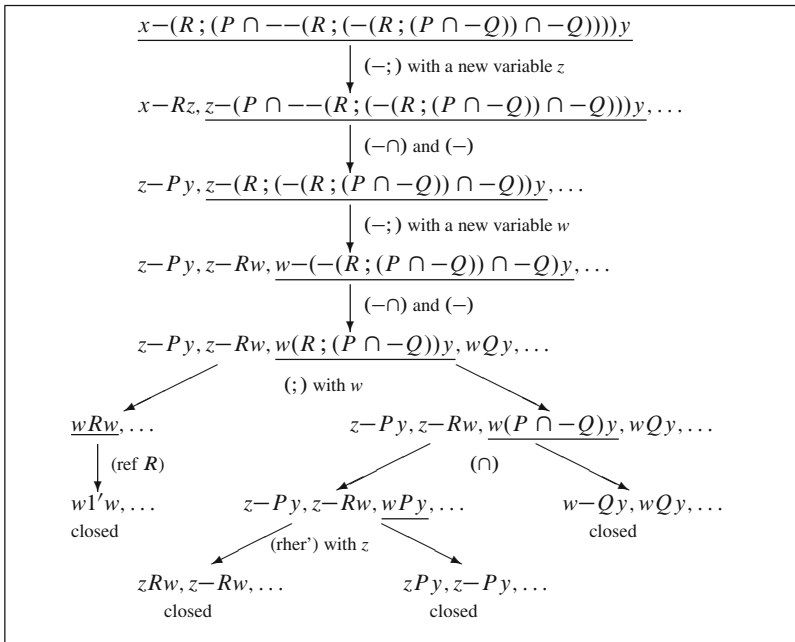


Fig. 8.2 An RL_J -proof of the formula $p \rightarrow \neg\neg p$

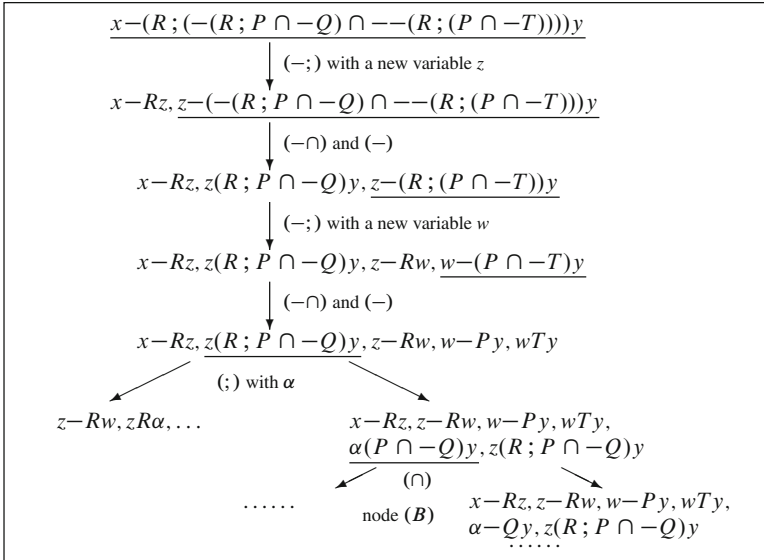


Fig. 8.3 A failed proof search of the formula $\neg p \rightarrow (p \rightarrow t)$

where $\tau(p) = P$ and $\tau(t) = T$. Since φ is not J-valid, any RL_J -proof tree must not close. Figure 8.3 presents an RL_J -proof tree of $x\tau(\varphi)y$ in which there is a branch that cannot be closed. Consider the node (B) in that tree which consists of the following formulas: $x-Rz, z-Rw, w-Py, wTy, \alpha-Qy, z(R; P \cap -Q)y$, where a variable α is arbitrary. The only decomposition rule that can be applied to node (B) is the rule (;), which generates, among others, a node consisting of the same formulas as (B) and a formula $\beta-Qy$, where β may be an arbitrary variable. We can observe that no application of the rule (;) would close the branch. Also, no application of a specific rule would close the branch. The rules (1'1) and (1'2) can be applied only to the formula wTy . As the result we obtain a node that consists of the same formulas as node (B) and the formula $y1'\beta$ or $w1'\beta$. Since node (B) does not contain a formula built with the complement of $1'$, the branch can be closed neither by an application of the rule (1'1) nor (1'2). The rules (ref R) and (tran R) cannot be applied to the node (B), because it does not contain a formula built with the atomic term R . Similarly, the rules (rQ1) and (rQ2) cannot be applied. The rules (rher') and (ideal) can be applied only to the formula wTy . However, observe that any application of the rule (rher') generates a node consisting of the same formulas as (B) together with the formula βTy . It is easy to see that it cannot close the branch. In a similar way, one can show that the same holds for the rule (ideal). Thus, the branch cannot be closed.

8.4 Relational Formalization of Some Intermediate Logics

Intermediate logics are the logics whose valid formulas include all the formulas that are valid in intuitionistic logic but not necessarily all the tautologies of classical propositional logic. In that sense, intermediate logics are between intuitionistic and classical logic. The common language of intermediate logics is the INT-language.

In this section we consider two intermediate logics INT_{L_1} and INT_{L_2} whose models are the INT-models satisfying the following conditions, respectively:

- (L₁) $\forall x \forall y \forall z \forall t [(x, y) \in R \wedge (x, z) \in R \wedge (y, t) \in R \rightarrow ((y, z) \in R \vee (z, y) \in R \vee (z, t) \in R)]$;
 (L₂) $\forall x \forall y \forall z [(x, y) \in R \wedge (x, z) \in R \rightarrow \exists t ((y, t) \in R \wedge (z, t) \in R)]$.

The specific axioms of the logics INT_{L_1} and INT_{L_2} in their Hilbert-style axiomatizations are:

- (A1) $[(\neg\neg\varphi \rightarrow \varphi) \rightarrow (\neg\varphi \vee \varphi)] \rightarrow (\neg\varphi \vee \neg\neg\varphi)$;
 (A2) $\neg\varphi \vee \neg\neg\varphi$.

INT_{L_1} is referred to as Scott's logic and INT_{L_2} is the logic of weak excluded middle.

The common language of a relational logic RL_{INT_L} for an intermediate logic $L \in \{L_1, L_2\}$ is the RL_{INT_L} -language. Models of RL_{INT_L} satisfy all the conditions assumed in INT_L -models for the accessibility relation R and (L₁) or (L₂), respectively. The translation of INT_L -formulas into relational terms is defined as for INT-formulas in Sect. 8.2. To obtain dual tableaux for these logics, we add to RL_{INT} -dual tableau the rules reflecting the specific semantic conditions that are assumed in their models. The specific rules reflecting the conditions (L₁) and (L₂) have the following forms:

For all object variables x, y, z , and t ,

$$(\text{rL}_1) \frac{yRz, zRy, zRt}{xRy, yRz, zRy, zRt \mid xRz, yRz, zRy, zRt \mid yRt, yRz, zRy, zRt}$$

$$(\text{rL}_2) \frac{}{xRy \mid xRz \mid y-(R; R^{-1})z}$$

The above rules have the property of preserving formulas built with atomic terms or their complements. Hence, the closed branch property holds. An alternative form of rule (rL₂) is discussed in Sect. 25.9.

Theorem 8.4.1 (Correspondence). *Let \mathcal{K} be a class of RL_{INT} -models and let $i \in \{1, 2\}$. Then, \mathcal{K} is a class of $\text{RL}_{\text{INT}_{L_i}}$ -models iff the rule (rL_i) is \mathcal{K} -correct.*

Proof.

(\Rightarrow) By way of example, we show it for $i=1$. Let \mathcal{K} be a class of $\text{RL}_{\text{INT}_{L_1}}$ -models, that is every model from \mathcal{K} satisfies the condition (L₁). Let X be a finite set of RL_{INT} -formulas. Clearly, if $X \cup \{yRz, zRy, zRt\}$ is a \mathcal{K} -set, then so are $X \cup \{xRy, yRz, zRy, zRt\}$, $X \cup \{xRz, yRz, zRy, zRt\}$, and $X \cup \{yRt, yRz, zRy, zRt\}$. Assume $X \cup \{xRy, yRz, zRy, zRt\}$, $X \cup \{xRz, yRz,$

zRy, zRt }, and $X \cup \{yRt, yRz, zRy, zRt\}$ are \mathcal{K} -sets. Suppose $X \cup \{yRz, zRy, zRt\}$ is not a \mathcal{K} -set. Then there exist an $\mathbf{RL}_{\text{INT}_{L_1}}$ -model $\mathcal{M} = (U, R, m)$ and a valuation v in \mathcal{M} such that for every $\varphi \in X \cup \{yRz, zRy, zRt\}$, $\mathcal{M}, v \not\models \varphi$, in particular, $(v(y), v(z)) \notin R$, $(v(z), v(y)) \notin R$, and $(v(z), v(t)) \notin R$. However, by the assumption, model \mathcal{M} and valuation v satisfy $(v(x), v(y)) \in R$, $(v(x), v(z)) \in R$, and $(v(y), v(t)) \in R$. By the condition (L_1) , it means that either $(v(y), v(z)) \in R$ or $(v(z), v(y)) \in R$ or $(v(z), v(t)) \in R$, a contradiction.

(\leftarrow) By way of example, we show it for $i = 2$. Let \mathcal{K} be a class of \mathbf{RL}_{INT} -models.

Assume the rule (rL_2) is \mathcal{K} -correct. Let $X \stackrel{\text{df}}{=} \{x-Ry, x-Rz, y(R; R^{-1})z\}$. Clearly, the sets $X \cup \{xRy\}$, $X \cup \{xRz\}$, and $X \cup \{y-(R; R^{-1})z\}$ are \mathcal{K} -sets. Thus, by the assumption, X must be a \mathcal{K} -set. Therefore, for every model $\mathcal{M} = (U, R, m)$ from \mathcal{K} and for all $x, y, z \in U$, if $(x, y) \in R$ and $(x, z) \in R$, then $(y, z) \in R; R^{-1}$. Since $(y, z) \in R; R^{-1}$ is equivalent to $\exists t((y, t) \in R \text{ and } (z, t) \in R)$, \mathcal{K} is a class of $\mathbf{RL}_{\text{INT}_{L_2}}$ -models. \square

The completion conditions determined by the rules (rL_1) and (rL_2) are:

For all object variables x, y, z , and t ,

$\text{Cpl}(rL_1)$ If the formulas yRz, zRy , and zRt are in b , then either $xRy \in b$ or $xRz \in b$ or $yRt \in b$;

$\text{Cpl}(rL_2)$ Either $xRy \in b$ or $xRz \in b$ or $y-(R; R^{-1})z \in b$.

In order to prove completeness, we need to show that the branch structure is an $\mathbf{RL}_{\text{INT}_L}$ -model, for $L \in \{L_1, L_2\}$.

Proposition 8.4.1 (Branch Model Property). *Let $L \in \{L_1, L_2\}$ and let b be an open branch of an $\mathbf{RL}_{\text{INT}_L}$ -proof tree. Then, \mathcal{M}^b is an $\mathbf{RL}_{\text{INT}_L}$ -model.*

Proof. It suffices to show that if $\mathbf{RL}_{\text{INT}_L}$ -dual tableau includes a rule (rL_i) , then \mathcal{M}^b satisfies the condition (L_i) , for $i \in \{1, 2\}$. By way of example, we show it for $i = 2$. Indeed, by the completion condition $\text{Cpl}(rL_2)$, for all $x, y, z \in U^b$, either $xRy \in b$ or $xRz \in b$ or $y-(R; R^{-1})z \in b$. Thus, by the completion conditions $\text{Cpl}(;)$ and $\text{Cpl}(-^{-1})$, for all $x, y, z \in U^b$, either $xRy \in b$ or $xRz \in b$ or there exists $t \in U^b$ such that $y-Rt \in b$ and $z-Rt \in b$. So $(x, y) \notin R^b$ or $(x, z) \notin R^b$ or there exists $t \in U^b$ such that $(y, t) \in R^b$ and $(z, t) \in R^b$. Therefore, for all $x, y, z \in U^b$, if $(x, y) \in R^b$ and $(x, z) \in R^b$, then there exists $t \in U^b$ such that $(y, t) \in R^b$ and $(z, t) \in R^b$, hence the condition (L_2) is satisfied. \square

Since the language of the logics $\mathbf{RL}_{\text{INT}_{L_1}}$ and $\mathbf{RL}_{\text{INT}_{L_2}}$ is the same as in \mathbf{RL}_{INT} -logic, the satisfaction in branch model property holds. Thus, the proof of completeness is similar as in \mathbf{RL}_{INT} -logic.

Theorem 8.4.2 (Soundness and Completeness of $\mathbf{RL}_{\text{INT}_L}$). *Let $L \in \{L_1, L_2\}$ and let φ be an $\mathbf{RL}_{\text{INT}_L}$ -formula. Then the following conditions are equivalent:*

1. φ is $\mathbf{RL}_{\text{INT}_L}$ -valid;
2. φ is true in all standard $\mathbf{RL}_{\text{INT}_L}$ -models;
3. φ is $\mathbf{RL}_{\text{INT}_L}$ -provable.

Theorem 8.4.3 (Relational Soundness and Completeness of INT_L). *Let $L \in \{L_1, L_2\}$ and let φ be an INT_L -formula. Then for all object variables x and y , the following conditions are equivalent:*

1. φ is INT_L -valid;
2. $x(\tau(\varphi))y$ is $\mathbf{RL}_{\text{INT}_L}$ -provable.

Example. Consider the following INT_{L_2} -formula:

$$\varphi = \neg p \vee \neg\neg p.$$

The translation of φ into $\mathbf{RL}_{\text{INT}_{L_2}}$ -term is:

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

where $\tau(p) = P$. INT_{L_2} -validity of φ is equivalent to $\mathbf{RL}_{\text{INT}_{L_2}}$ -provability of the formula $x\tau(\varphi)y$. Figure 8.4 presents its $\mathbf{RL}_{\text{INT}_{L_2}}$ -proof.

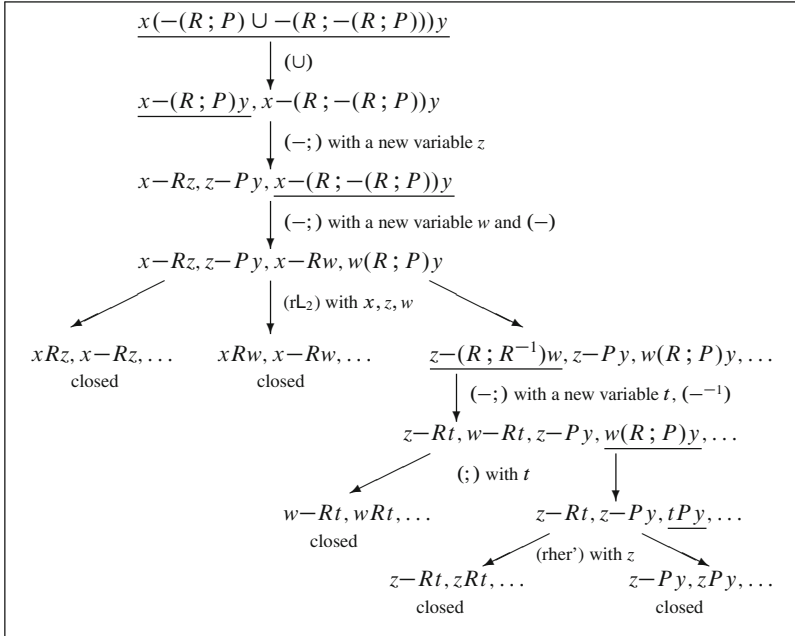


Fig. 8.4 An $\mathbf{RL}_{\text{INT}_{L_2}}$ -proof of the formula $\neg p \vee \neg\neg p$

8.5 Relational Formalization of a Logic for Hardware Verification

In this section we consider a propositional logic PLL which has been proposed as a tool for a formal verification of computer hardware (see [FM95, Men90, Men93]).

The logic PLL is obtained from the logic INT by endowing its language with a unary propositional operation \bigcirc that models a delay of propagation of signals. Signals are conceived as Boolean valued functions, where Boolean values are denoted by 1 and 0. To input and output signals of the gates we assign propositional variables. If p is such a variable, then truth of p in a model is interpreted as ‘ p is stable at 1’, truth of $\neg p$ means “ p is stable at 0’, truth of $\bigcirc p$ means ‘ p is going to stabilize to 1’, and truth of $\bigcirc \neg p$ means ‘ p is going to stabilize to 0’.

A PLL-model is a structure $\mathcal{M} = (U, R, S, m)$ such that (U, R, m) is an INT-model and S is a reflexive and transitive relation on U such that $S \subseteq R$. Then, the satisfaction of formulas of the form $\bigcirc \varphi$ is defined as follows:

$\mathcal{M}, s \models \bigcirc \varphi$ iff for every $s' \in U$, if $(s, s') \in R$, then there exists $t \in U$ such that $(s', t) \in S$ and $\mathcal{M}, t \models \varphi$.

The language of the relational logic RL_{PLL} is the language of RL_{INT} -logic endowed with the relational constant S .

An RL_{PLL} -model is a structure $\mathcal{M} = (U, R, S, m)$ such that (U, R, m) is an RL_{INT} -model and S is a reflexive and transitive relation on U satisfying $S \subseteq R$.

The translation τ from PLL-formulas into relational terms is defined as the translation in Sect. 8.2 with the following clause for formulas of the form $\bigcirc \varphi$:

$$\tau(\bigcirc \varphi) \stackrel{\text{df}}{=} \neg(R; \neg(S; (\tau(\varphi))))).$$

Much the same as we proved Theorem 8.2.1, we can show:

Theorem 8.5.1. *For every PLL-formula φ and for all object variables x and y , the following conditions are equivalent:*

1. φ is PLL-valid;
2. $x\tau(\varphi)y$ is RL_{PLL} -valid.

A dual tableau for the logic RL_{PLL} consists of the axiomatic sets and the rules of dual tableau for RL_{INT} -logic (see Sect. 8.2) adjusted to the language of RL_{PLL} , the rules (ref S) and (tran S) reflecting reflexivity and transitivity of S , respectively (see Sect. 6.6), and in addition the rule (SR) of the following form:

For all object variables x and y ,

$$(SR) \quad \frac{xRy}{xSy, xRy}$$

The above rule reflects the condition $S \subseteq R$. The rule (SR) preserves formulas built with atomic terms. Hence, the closed branch property holds. It is easy to see that the rule (SR) is RL_{PLL} -correct.

Proposition 8.5.1. *The RL_{PLL} -rules are RL_{PLL} -correct.*

The completion condition determined by the rule (SR) is:

For all object variables x and y ,

$\text{Cpl}(SR)$ If $xRy \in b$, then $xSy \in b$.

The branch model is defined in a standard way, that is $m^b(T) = \{(x, y) \in U^b \times U^b : xTy \notin b\}$, for every $T \in \mathbb{R} \vee_{\text{RL}_{\text{PLL}}} \cup \{R, S\}$.

Proposition 8.5.2 (Branch Model Property). *For every open branch b of an RL_{PLL} -proof tree, \mathcal{M}^b is an RL_{PLL} -model.*

Proof. It suffices to show that S^b satisfies all the conditions assumed in RL_{PLL} -models. Reflexivity and transitivity of S^b can be proved as in Theorem 6.6.1. Now, assume that $(x, y) \in S^b$, that is $xSy \notin b$. Suppose $(x, y) \notin R^b$, that is $xRy \in b$. Then, by the completion condition $\text{Cpl}(SR)$, we get $xSy \in b$, a contradiction. Hence, \mathcal{M}^b is an RL_{PLL} -model. \square

Since the branch model \mathcal{M}^b is defined in a standard way (see Sect. 2.6, p. 44), the satisfaction in branch model property can be proved as in $\text{RL}(1, 1')$ -logic (see Sects. 2.5 and 2.7). Hence, completeness of RL_{PLL} -dual tableau follows.

Theorem 8.5.2 (Soundness and Completeness of RL_{PLL}). *Let φ be an RL_{PLL} -formula. Then the following conditions are equivalent:*

1. φ is RL_{PLL} -valid;
2. φ is true in all standard RL_{PLL} -models;
3. φ is RL_{PLL} -provable.

By the above theorem and Theorem 8.5.1, we have:

Theorem 8.5.3 (Relational Soundness and Completeness of PLL). *Let φ be a PLL-formula. Then for all object variables x and y , the following conditions are equivalent:*

1. φ is PLL-valid;
2. $x(\tau(\varphi))y$ is RL_{PLL} -provable.

Example. Consider the following PLL-formula:

$$\varphi = p \rightarrow \bigcirc p.$$

The translation of φ into RL_{PLL} -term is:

$$\tau(\varphi) = \neg(R; (P \cap \neg\neg(R; \neg(S; P))))),$$

where $\tau(p) = P$. PLL-validity of φ is equivalent to RL_{PLL} -provability of the formula $x\tau(\varphi)y$. Figure 8.5 presents its RL_{PLL} -proof.

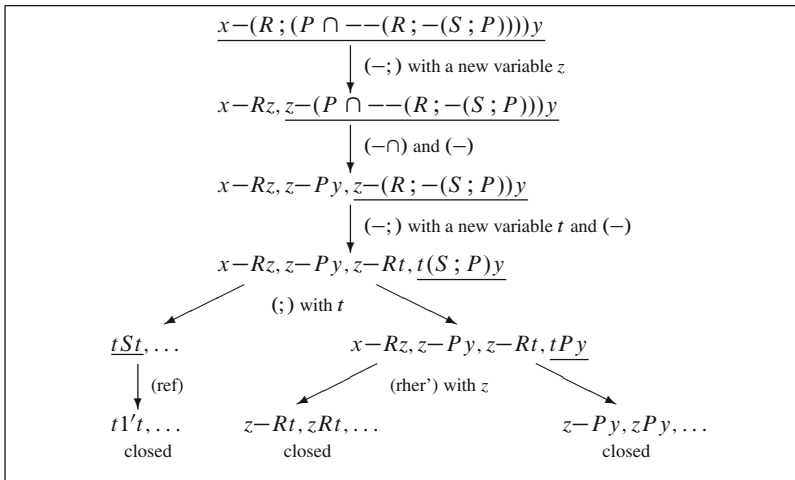


Fig. 8.5 An RL_{PLL} -proof of the formula $p \rightarrow \bigcirc p$