City University of New York (CUNY)
CUNY Academic Works

Computer Science Technical Reports | CUNY Academic Works

2005

TR-2005013: Typing in Reflective Combinatory Logic

Nikolai Krupski

How does access to this work benefit you? Let us know!

More information about this work at: https://academicworks.cuny.edu/gc_cs_tr/268
Discover additional works at: https://academicworks.cuny.edu

This work is made publicly available by the City University of New York (CUNY).
Contact: AcademicWorks@cuny.edu

Typing in Reflective Combinatory Logic

Nikolai Krupski

Department of Math. Logic and the Theory of Algorithms,

Faculty of Mechanics and Mathematics,

Moscow State University, Moscow 119992, Russia

nvk@lpcs.math.msu.su

Abstract

We study the syntax of Artemov's Reflective Combinatory Logic $\mathsf{RCL}_{\to}$. We provide the explicit definition of types for $\mathsf{RCL}_{\to}$ and prove that every well-formed term has a unique type. We establish that the typability testing and detailed type restoration can be done in polynomial time and that the derivability relation for $\mathsf{RCL}_{\to}$ is decidable and $PSPACE$-complete. These results also formalize the intended semantics of the type $t{:}F$ in $\mathsf{RCL}_{\to}$. Terms $\mathsf{RCL}_{\to}$ store the complete information about the judgment "t is a term of type F", and this information can be extracted by the type restoration algorithm.

1 Introduction.

The Reflective Combinatory Logic $\mathsf{RCL}_{\to}$ was introduced by S. Artemov in [4]. [1] $\mathsf{RCL}_{\to}$ is an extension of Typed Combinatory Logic $\mathsf{CL}_{\to}$ (see [12]) which admits the embedding of typing judgments of the form "t is a term of type F" into the types of the system via the additional type constructor $t : F$. A similar type constructor is available in Intuitionistic Type Theory (ITT, see [7, 10]) but there it is trivial in the following sense: any nonempty ITT-type

[1] The definitions appear a year before this publication in Artemov's public lectures on computational logic at Lomonosov Moscow State University and are available from the homepage of the course since March 2003. http://lpcs.math.msu.su/rus/cl.htm

$t : F$ contains a unique canonical element which is the same for all types of this form. Unlike the case of ITT a type $t : F$ in $\mathsf{RCL}_{\to}$ is nontrivial. It is inhabited by terms which store the information about the typing judgment. The term part t of the judgment can be restored by the reflection operation represented inside the system by special combinator $d : (t : F \to F)$. In this paper the restoration of the remaining part F is considered. We provide an algorithm for more general type restoration problem. Being applied to a member of type $t : F$ it restores the complete judgment $t : F$.

The notation for types in $\mathsf{RCL}_{\to}$ (also called well-formed formulas) allows terms inside types, so it is not well-defined unless the types of these terms are unique. We prove that the definition of terms and types in $\mathsf{RCL}_{\to}$ is correct by establishing that every well-formed term has a unique type and providing a detailed notation with explicit typing. It is shown that the typability testing and detailed type restoration can be done in polynomial time, so $\mathsf{RCL}_{\to}$ admits polynomial time well-formedness test. We also give a cut-free sequent formulation for $\mathsf{RCL}_{\to}$ and prove that the derivability relation defined by $\mathsf{RCL}_{\to}$ is $PSPACE$-complete.

2 Well-formedness definition

The formal system $\mathsf{RCL}_{\to}$ is introduced in [4] by complex inductive definition. The following two judgments are defined by simultaneous induction: "F is a well-formed formula (or type)" and "F is derivable from $F_1, \dots, F_n$". The last one has the supposed informal meaning "the type F is inhabited provided all types F_i are". The notion of derivation reflected in the second judgment has the sole inference rule Modus Ponens and it is postulated explicitly that Modus Ponens preserves well-formedness. The well-formedness precondition for an expression of certain kind to be an axiom is required too.

We try to decompose the definition from [4] and succeed in extracting its well-formedness brunch as a separate calculus RCL–wf (the implication is omitted in order to simplify the notation).

Let $p_0, p_1, \dots$ be propositional variables (or type variables), k, s, d, o, c and $x_0, x_1, \dots$ be two groups of identifiers.

Rules of RCL–wf:

$$1.\ p_i - \mathrm{wf} \qquad 2.\ \frac{F, G - \mathrm{wf}}{F \to G - \mathrm{wf}} \qquad 3.\ \frac{F - \mathrm{wf}}{x_i^F : F - \mathrm{wf}}$$

$$4.\ \frac{F,\, t\!:\!F - \mathrm{wf}}{d^{\,tF\to F}\!:\!(t\!:\!F \to F) - \mathrm{wf}} \qquad 5.\ \frac{F \to (G \to F) - \mathrm{wf}}{k^{\,F\to(G\to F)}\!:\!(F \to (G \to F)) - \mathrm{wf}}$$

$$6.\ \frac{(F \to (G \to H)) \to ((F \to G) \to (F \to H)) - \mathrm{wf}}{s^{(F\to(G\to H))\to((F\to G)\to(F\to H))}\!:\!((F \to (G \to H)) \to ((F \to G) \to (F \to H))) - \mathrm{wf}}$$

$$7.\ \frac{F,\, G,\, u\!:\!(F \to G),\, v\!:\!F - \mathrm{wf}}{(uv)\!:\!G - \mathrm{wf}}$$

$$8.\ \frac{F,\, G,\, u\!:\!(F \to G),\, v\!:\!F - \mathrm{wf}}{o^{u:(F\to G)\to(vF\to(uv)G)}\!:\!(u\!:\!(F \to G) \to (v\!:\!F \to (uv)\!:\!G)) - \mathrm{wf}}$$

$$9.\ \frac{F,\, t\!:\!F - \mathrm{wf}}{!t\!:\!(t\!:\!F) - \mathrm{wf}} \qquad 10.\ \frac{F,\, t\!:\!F - \mathrm{wf}}{c^{tF\to !t(tF)}\!:\!(t\!:\!F \to !t\!:\!(t\!:\!F)) - \mathrm{wf}}$$

$$11.\ \frac{F \to G,\, F - \mathrm{wf}}{G - \mathrm{wf}}$$

In [4] no explicit definition of terms is given. It is suggested to reread the judgments of the form "$t : F$ is a well-formed formula" as "t is a term of type F". Thus x_i^F is a variable term of type F (or, simply, variable of type F) provided the judgment "$x_i^F : F - \mathrm{wf}$" is derivable in RCL–wf. The same with the constants $k^{(\dots)}, s^{(\dots)}, d^{(\dots)}, o^{(\dots)}, c^{(\dots)}$, but $\mathsf{RCL}_{\to}$ postulates that they inhabit their types. The identifiers (without superscripts) have no special meaning at all.

Definition 2.1 The sets of all *terms* (Tm) and all *formulas* (Fm) are defined by the following grammar:

$$\begin{array}{lcl} Tm & ::= & x_i^{Fm} \mid k^{Fm} \mid s^{Fm} \mid o^{Fm} \mid d^{Fm} \mid c^{Fm} \mid !\,Tm \mid Tm \cdot Tm, \\ Fm & ::= & p_i \mid Fm \to Fm \mid Tm\!:\!Fm. \end{array}$$

Variables are terms of the form x_i^{Fm}. *Constants* are terms of the forms k^{Fm}, s^{Fm}, o^{Fm}, d^{Fm} or c^{Fm}. A formula F is called *well-formed* if the judgment "$F - \mathrm{wf}$" is provable in RCL–wf. A term t is called *typable* if the judgment "$t\!:\!F - \mathrm{wf}$" for some F is provable in RCL–wf.

Definition 2.2 For an expression $e \in Tm \cup Fm$ we define two sets — the set $Sub(e)$ of *subexpressions* and the set $ASub(e)$ of *associated subexpressions*. $Sub(e)$ consists of all terms and formulas that occur inside e but this occurrence is not inside a superscript. $ASub(e)$ consists of all terms and formulas that occur inside e including the occurrences inside superscripts, inside superscripts in superscripts, etc. A subexpression is called a subterm or a subformula when it is a term or a formula respectively. The same with associated subterms and subformulas. The *size* of an expression $e \in Tm \cup Fm$ will be measured as the cardinality of $ASub(e)$.

3 Typing of subterms

We simplify the formulation of the calculus RCL–wf.

Lemma 3.1 *If the judgment "X – wf" is derivable in RCL–wf then it is derivable in RCL–wf without Rule 11 too.*

Proof. Consider the shortest derivation of "X – wf" and the first occurrence of Rule 11 in it. Its premise "$F \to G$ – wf" can be derived by Rule 2 only. But this derivation can be reduced:

$$\dfrac{\dfrac{F - \mathrm{wf} \quad G - \mathrm{wf}}{F \to G - \mathrm{wf}}\,(2) \qquad F - \mathrm{wf}}{G - \mathrm{wf}}\,(11) \quad \longmapsto G - \mathrm{wf}.$$

Contradiction with the choice of the derivation. ∎

Definition 3.2 A RCL–wf-derivation without Rule 11 will be called *MP-free derivation.*

Corollary 3.3 *The formula $X \to Y$ is well-formed iff X and Y are well-formed.*

Corollary 3.4 *The replacement of the premises in Rules 5 and 6 of RCL–wf by "F, G – wf" and "F, G, H – wf" respectively does not change the set of derivable judgments.*

Lemma 3.5 *If both formulas $t{:}X$ and $t{:}Y$ are well-formed then X and Y coincide.*

Proof. Suppose the opposite. Let "$t\!:\!X$ – wf" and "$t\!:\!Y$ – wf" be derivable without Rule 11, $X \neq Y$ and the total length l of these MP-free derivations is minimal. Note that the last rules in both derivations should be the same. It cannot be Rule 1 or 2. If it is one of Rules 3,4,5,6,8,10 then the instances of the rule must coincide too, which contradicts with the condition "$X \neq Y$". The remaining cases are:

Rule 7 ($t = uv$).

$$\frac{A,\ X,\ u\!:\!(A \rightarrow X),\ v\!:\!A - \mathrm{wf}}{(uv)\!:\!X - \mathrm{wf}} \quad \text{and} \quad \frac{B,\ Y,\ u\!:\!(B \rightarrow Y),\ v\!:\!B - \mathrm{wf}}{(uv)\!:\!Y - \mathrm{wf}}$$

The total length of the derivations of "$u\!:\!(A \rightarrow X)$ – wf" and "$u\!:\!(B \rightarrow Y)$ – wf" is less than l and $(A \rightarrow X) \neq (B \rightarrow Y)$. Contradiction.

Rule 9. Similar to the previous one. ∎

Lemma 3.6 *If t is a subterm of a well-formed formula F then t is typable.*

Proof. Straightforward induction on the MP-free derivation of the formula "F – wf". ∎

Lemma 3.7 *If a formula "$t\!:\!X$" is well-formed then X is also well-formed.*

Proof. Straightforward induction on the MP-free derivation of the judgement "$t\!:\!X$ – wf". ∎

Corollary 3.8 *In Rules 4,7,8,9 and 10 of* RCL*–wf the premises "F – wf" and "G – wf" can be omitted without the change of the set of derivable judgments.*

Theorem 3.9 *1. If t is an associated subterm of some well-formed formula then t is typable. Moreover, the formula G such that $t\!:\!G$ is well-formed is unique and also well-formed.*

2. Every associated subformula of a well-formed formula is well-formed.

Proof. 1. Let t be an associated subterm of a well-formed formula F. We prove that t is typable by induction on the superscript height of the occurrence of t in F. Lemma 3.6 covers the case when t is a subterm of F. Now suppose that t occurs in a superscript H of a variable or a constant $\xi^H \in Sub(F)$. By the induction hypothesis, the formula $\xi^H\!:\!H_1$ is well-formed for some H_1. But there is no MP-free derivation of a formula of this form when H_1 is different from H. So $H = H_1$ and H is well-formed by Lemma 3.7. Now we can prove the typability of t by the induction hypothesis applied to the occurrence of t in H.

The uniqueness and well-formedness of corresponding formula G is proved in Lemmas 3.5, 3.7.

2. Let G be an associated subformula of a well-formed formula F. Induction on the superscript height of the occurrence of G in F. The well-formedness of G for $G \in Sub(F)$ follows from Corollary 3.3 and Lemma 3.7. Suppose that G occurs in a superscript H of a variable or a constant $\xi^H \in Sub(F)$. By part 1, the term ξ^H is typable, so $\xi^H : H$ and H are well-formed. It remains to apply the induction hypothesis to the occurrence of G in H. ∎

4 Types of $\mathsf{RCL}_{\to}$

By Theorem 3.9, the associated subterms of well-formed formulas can be labelled by the unique formulas denoting their types. We shall write these labels as superscripts and extend the labelling recursively in superscripts. This results in the explicit definition of types for $\mathsf{RCL}_{\to}$ whereas the well-formed formulas provide the short form notation for these types.

The well-formedness judgment for types will be written as "F type". It is defined by the following calculus RCLT–wf.

Rules of RCLT–wf:

$$1.\ p_i \text{ type} \qquad 2.\ \frac{F, G \text{ type}}{F \to G \text{ type}} \qquad 3.\ \frac{F \text{ type}}{x_i^F : F \text{ type}}$$

$$4.\ \frac{u : F \text{ type}}{d^{u:F\to F} : (u : F \to F) \text{ type}} \qquad 5.\ \frac{F,\ G \text{ type}}{k^{F\to(G\to F)} : (F \to (G \to F)) \text{ type}}$$

$$6.\ \frac{F,\ G,\ H \text{ type}}{s^{(F\to(G\to H))\to((F\to G)\to(F\to H))} : ((F \to (G \to H)) \to ((F \to G) \to (F \to H))) \text{ type}}$$

$$7.\ \frac{u : (F \to G),\ v : F \text{ type}}{(uv)^G : G \text{ type}}$$

8. $$\frac{u\!:\!(F \to G),\ v\!:\!F \text{ type}}{o^{u:(F\to G)\to(v F\to(uv)^G:G)}\!:\!(u\!:\!(F \to G) \to (v\!:\!F \to (uv)^G\!:\!G)) \text{ type}}$$

9. $$\frac{u\!:\!F \text{ type}}{(!u)^{u:F}\!:\!(u\!:\!F) \text{ type}}$$ 10. $$\frac{u\!:\!F \text{ type}}{c^{u:F\to(!u)^{u:F}:(u:F)}\!:\!(u\!:\!F \to (!u)^{u:F}\!:\!(u\!:\!F)) \text{ type}}$$

We adjust Definition 2.1 to this new syntax:

Definition 4.1 The sets of all *labelled terms* (LTm) and all *labelled formulas* (LFm) are defined by the following grammar:

$$\begin{array}{rcl} LTm & ::= & x_i^{LFm} \mid k^{LFm} \mid s^{LFm} \mid o^{LFm} \mid d^{LFm} \mid c^{LFm} \\ & & \mid (!\,LTm)^{LFm} \mid (LTm \cdot LTm)^{LFm}, \\ LFm & ::= & p_i \mid LFm \to LFm \mid LTm\!:\!LFm. \end{array}$$

A labelled formula F is called a *type* if the judgment "F type" is provable in RCLT–wf. A labelled term t^F is called a *typed term* if the judgment "$t^F : F$ type" is provable in RCLT–wf. The definitions of *subexpressions*, *associated subexpressions* and *size* for a labelled expression $e \in LTm \cup LFm$ should be modified in the similar way.

Let us consider the (partial) translation $(\cdot)^r$ which restores superscripts in expressions $e \in Tm \cup Fm$ when possible:

$(p_i)^r = p_i, \quad (F \to G)^r = F^r \to G^r,$

$(\xi^F)^r = \xi^{F^r}$ where ξ is an identifier and $\mathsf{RCL}\text{–wf} \vdash\ \xi^F\!:\!F$ – wf,

$(uv)^r = (u^r v^r)^{G^r}$ where $\mathsf{RCL}\text{–wf} \vdash\ uv : G$ – wf,

$(!t)^r = (!t^r)^{G^r}$ where $\mathsf{RCL}\text{–wf} \vdash\, !t : G$ – wf,

$(t\!:\!F)^r = t^r\!:\!F^r$ when $\mathsf{RCL}\text{–wf} \vdash t\!:\!F$ – wf.

The reverse translation $(\cdot)^0$ is the forgetful projection. It deletes the superscripts of associated labelled subterms of the forms $(u \cdot v)^F$ and $(!t)^G$. It is easy to see that $(e^r)^0 = e$ whenever e^r is defined.

Lemma 4.2 *For every well-formed formula X its translation X^r is defined.*

Proof. Consider the shortest MP-free derivation of "X – wf" with undefined X^r and the last rule in it. It is easy to see that the last rule cannot be one of Rules 1, 2, 3, 4, 5, 6, 9, 10.

Case of Rule 7. For its premises the values $(u:(F \to G))^r$ and $(v:F)^r$ are defined. So u^r, v^r and G^r are defined too. But "$(uv):G$ – wf" (the conclusion of the rule) is derivable, so $(uv)^r = (u^r v^r)^{G^r}$ is also defined. Then $X^r = (uv)^r : G^r$ is defined. Contradiction.

Case of Rule 8. Let us replace the last Rule 8 with Rule 7 with the same premises. The resulting derivation is shorter than the initial one, so the value of $((uv):G)^r$ for its conclusion is defined. By the same reason the values $(u:(F \to G))^r$ and $(v:F)^r$ for the premises are defined too. Thus,

$$Y = (u:(F \to G) \to (v:F \to (uv):G))^r$$

is defined. But the judgment "X – wf" is derivable, so $(o^{(\dots)})^r = o^Y$ and $X^r = o^Y : Y$. Contradiction. ∎

Below we assume that all expressions and labelled expressions are represented by appropriate DAGs (labelled Directed Acyclic Graphs; see [5] for examples), so the identical parts are never stores twice. The number of nodes of the DAG representing an expression (labelled expression) e coincides with the size of e.

Theorem 4.3 *1.* RCL*–wf* $\vdash$ *X – wf* *implies* RCLT*–wf* $\vdash$ *X^r type. Moreover, there exists a polynomial time algorithm that transforms any MP-free derivation of "X – wf" into a derivation of "X^r type".*

2. RCLT*–wf* $\vdash$ *X type* *implies* RCL*–wf* $\vdash$ *X^0 – wf.*

3. $(X^0)^r = X$ holds for every type X.

Proof. 1. Let an MP-free derivation of"X – wf" be given. First simplify the rules 4, 5, 6, 7, 8, 9 and 10 in it as it is recommended by Corollaries 3.4, 3.8. Consider an instance of a rule that occurs in the simplified derivation:

$$\frac{Y_1, \dots, Y_k \text{ – wf}}{Y \text{ – wf}} \tag{1}$$

Its premises and conclusion are derivable, so $(Y_i)^r$ and $(Y)^r$ are defined by Lemma 4.2. The translation $(\cdot)^r$ converts (1) into

$$\frac{Y_1^r, \dots, Y_k^r \text{ type}}{Y^r \text{ type}}. \tag{2}$$

The inspection of the rules shows that (2) is an instance of corresponding rule of RCLT–wf.

So in order to derive "X^r type" one should apply the rules of RCLT–wf with the same numbers in the same order as in the simplified MP-free derivation of "X – wf". But for every rule of RCLT–wf there exists a polynomial time algorithm that restores the conclusion given the premise(s) of the rule. One can implement the application of rules by application of corresponding algorithms. It results in the polynomial time proof conversion method.

2. The forgetful projections of the rules of RCLT–wf are admissible in RCL–wf.

3. It is sufficient to prove that $X^0 = Y^0$ implies $X = Y$ when X, Y are types. Induction on the total length of the derivations of "X type" and "Y type". The last rule in both derivations should be the same. For example, consider the case of Rule 7:

$$\frac{u\!:\!(F \to G),\ v\!:\!F \text{ type}}{\underbrace{(uv)^G\!:\!G}_{X} \text{ type}} \qquad\qquad \frac{u_1\!:\!(F_1 \to G_1),\ v_1\!:\!F_1 \text{ type}}{\underbrace{(u_1v_1)^{G_1}\!:\!G_1}_{Y} \text{ type}}$$

Suppose $X^0 = Y^0$. Then $u^0 = u_1^0$. By part 2, the formulas $u^0\!:\!(F^0 \to G^0)$ and $u_1^0\!:\!(F_1^0 \to G_1^0)$ are well-formed. Then $F^0 = F_1^0$ and $G^0 = G_1^0$ (Lemma 3.5). By the induction hypothesis, $u\!:\!(F \to G)$ coincides with $u_1\!:\!(F_1 \to G_1)$. Similarly we prove that $v\!:\!F$ coincides with $v_1\!:\!F_1$. Thus, $X = Y$.

Other rules can be treated in the same way.

■

Corollary 4.4 *An associated subformula of a type is a type.*

Comment. The proof of Theorem 4.3 (part 1) gives a polynomial time algorithm which converts a RCL–wf-derivations of "X – wf" into RCLT–wf-derivation of "X^r type". It can be used as a part of the following type restoration method: given X search for a RCL–wf-derivation of "X – wf", convert it into RCLT–wf-derivation of "X^r type" and extract X^r from it. Below we show that the proof search can be done in polynomial time too so the method gives a polynomial time solution for the type restoration problem concerning $\mathsf{RCL}_{\to}$.

5 Type restoration

In this section we prove that the following two problems can be solved in polynomial time:

Typing. Given a term t to test its typability and to find t^r when it is typable.

Type restoration. Given a formula F to test its well-formedness and to find F^r when it is well-formed.

In algorithms we use the proof search for more primitive calculus $\mathsf{RLP}_{\rightarrow}$ (see [9]). The language of $\mathsf{RLP}_{\rightarrow}$ is a fragment of the language of Logic of Proofs (LP language, see [1, 2, 3, 4]). The main difference from $\mathsf{RCL}_{\rightarrow}$ is that unlike Definition 2.1 the language of LP admits any names for variables and constants and treats them as atoms without analyzing the structure of the name. In this fragment terms (proof polynomials) are build from such variables and constants using the same operations "!" and "$\cdot$". Formulas are build from propositional variables p_i and quasiatomic formulas using "$\rightarrow$" only and quasiatomic formulas have the form $t\!:\!F$ where t is a term and F is a formula.

Axioms of $\mathsf{RLP}_{\rightarrow}$: all formulas of the form $a : A$ where a is a constant and A has one of the forms:

$$\begin{array}{l} F \rightarrow (G \rightarrow F), \\ (F \rightarrow (G \rightarrow H)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow H)), \\ u\!:\!F \rightarrow F, \\ u\!:\!(F \rightarrow G) \rightarrow (v\!:\!F \rightarrow uv\!:\!G), \\ u\!:\!F \rightarrow !u\!:\!(u\!:\!F). \end{array} \qquad (3)$$

Rules of $\mathsf{RLP}_{\rightarrow}$:

$$(\mathbf{C1}) \quad \frac{u\!:\!(F \rightarrow G) \quad v\!:\!F}{(uv)\!:\!G} \qquad\qquad (\mathbf{C2}) \quad \frac{t\!:\!F}{!t\!:\!(t\!:\!F)}$$

Definition 5.1 We use variable terms and constant terms from the language of $\mathsf{RCL}_{\rightarrow}$ as atomic names for variables and constants in $\mathsf{RLP}_{\rightarrow}$, so the sets of all $\mathsf{RLP}_{\rightarrow}$-terms and all $\mathsf{RLP}_{\rightarrow}$-formulas coincide with Tm and Fm respectively (see Definition 2.1). The formulas from (3) will be called *patterns*: k-pattern, s-pattern, … , c-pattern respectively. We shall say that an $\mathsf{RLP}_{\rightarrow}$-derivation *respects patterns* if the following holds: whenever an axiom $a\!:\!A$ is involved then a has the form i^A, $i \in \{k, s, d, o, c\}$ and A is the i-pattern.

(This means that $a : A$ can be derived from A by one of Rules 4, 5, 6, 8 or 10.)

Let t be a term (considered as $\mathsf{RCL}_{\rightarrow}$-term). Let $Var(t)$ and $Cnst(t)$ be the sets of all variable subterms and all constant subterms of t respectively, $Hyp(t) = \{x^G : G \mid x^G \in Var(t)\}$.

Lemma 5.2 *Let t be a term. It is typable iff for every $\xi^G \in Var(t) \cup Cnst(t)$ the formula G is well-formed and $Hyp(t) \vdash_{\mathsf{RLP}_{\rightarrow}} t : F$ for some F and this derivation respects patterns. Moreover, if all these tests succeed then the derivation $Hyp(t) \vdash_{\mathsf{RLP}_{\rightarrow}} t : F$ is also an $\mathsf{RCL}_{\rightarrow}$-derivation of "$t : F$ – wf" from hypotheses "$\xi^G : G$ – wf", $\xi^G \in Var(t) \cup Cnst(t)$.*

Proof. Suppose that t is typable. The judgment "$t : F$ – wf" for some F has an MP-free derivation. By Corollary 3.8, we can assume that Rules 7, 9 in the derivation are in reduced forms, i.e. they coincide with Rules **C1** and **C2** of $\mathsf{RCL}_{\rightarrow}$ respectively.

The derivation tree already contains the subtrees that derive all the judgments "$\xi^G : G$ – wf" where $\xi^G \in Var(t) \cup Cnst(t)$. Let us remove them. The remaining part is the $\mathsf{RCL}_{\rightarrow}$-derivation $Hyp(t) \vdash_{\mathsf{RLP}_{\rightarrow}} t : F$ which respects patterns. All the formulas G with $\xi^G \in Var(t) \cup Cnst(t)$ are well-formed by Lemma 3.7.

Now suppose that the $\mathsf{RLP}_{\rightarrow}$-derivation $Hyp(t) \vdash_{\mathsf{RLP}_{\rightarrow}} t : F$ is given, it respect patterns and every G with $\xi^G \in Var(t) \cup Cnst(t)$ is well-formed. One can derive "$x^G : G$ – wf" for $x^G \in Var(t)$ by Rule 3. The requirement to respect patterns guarantees the derivability of the judgments "$i^G : G$ – wf" for $i^G \in Cnst(t)$ by one of Rules 4, 5, 6, 8 or 10. Thus, all axioms and hypotheses involved in the $\mathsf{RLP}_{\rightarrow}$-derivation are provable in $\mathsf{RCL}_{\rightarrow}$ and all $\mathsf{RLP}_{\rightarrow}$-rules are admissible in $\mathsf{RCL}_{\rightarrow}$, so $\mathsf{RCL}_{\rightarrow} \vdash t : F$ – wf. ∎

Lemma 5.3 *For every term t there exists at most one formula F such that $Hyp(t) \vdash_{\mathsf{RLP}_{\rightarrow}} t : F$ and this derivation respects patterns. There exists a polynomial time algorithm that tests the condition above and restores the corresponding derivation (if it exists) given t.*

Proof. We adopt the proof search method from [9]. By induction on the complexity of t we prove that F is unique and fix some standard form of $\mathsf{RLP}_{\rightarrow}$-derivation of $t : F$ from $Hyp(t)$. Cases when t is a variable term or a constant term are trivial. Let t be a non-atomic term. Consider a minimal derivation of the form $Hyp(t) \vdash_{\mathsf{RLP}_\to} t : F$ that respects patterns (when it exists). It is a sequence of formulas

$$u_1 : X_1, \ldots, u_n : X_n \tag{4}$$

where $u_1, \ldots, u_n$ is the list of all subterms of t and the order satisfies the condition:

$$u_i \in Sub(u_j) \ \Rightarrow \ i \leq j. \tag{5}$$

We may fix some standard ordering of subterms because any permutation that preserves the condition (5) does not break the derivation. The induction hypothesis and minimality condition imply that $u_i \neq u_j$ when $i \neq j$.

The following equalities expresses a relation between premise(s) and the conclusion of the inference rule involved:

$$\begin{array}{ll} X_i = G & \text{for } u_i = \xi^G \in Var(t) \cup Cnst(t); \\ X_k \ = \ X_l \to X_i & \text{for } u_i = u_k u_l; \\ X_i = u_j : X_j & \text{for } u_i = !u_j. \end{array} \tag{6}$$

They determine the formulas X_i uniquely for given t. Thus, the formula $F = X_n$ is unique if exists.

Now consider (6) as a unification problem, i.e. a set of equations on syntactical variables X_k. It can be restored in polynomial time from t for every $t \in Tm$. It is unifiable iff $Hyp(t) \vdash_{\mathsf{RLP}_\to} t : F$. The corresponding derivation can be obtained by substituting the values of X_i in (4). One can test the unifiability of (6) and find the solution by polynomial time unification algorithm (see [5]). ∎

Theorem 5.4 *Typing and Type restoration problems for* $\mathsf{RCL}_\to$ *can be solved in polynomial time.*

Proof. We have seen (Theorem 4.3) that Type restoration problem is polynomial time reducible to the following one: given a formula X to test its well-formedness and restore the MP-free derivation of "X – wf" when it exists. Here we give the polynomial time solution to the latter one:

1. Construct the derivation of "X – wf" from the hypotheses "$t : F$ – wf" where $t : F \in Sub(X)$.

2. For every formula $t : F \in Sub(X)$ apply Lemmas 5.2, 5.3 to t and restore the $\mathsf{RCL}_{\to}$-derivation of "$t : F'$ – wf" from hypotheses "$\xi^G : G$ – wf", $\xi^G \in Var(t) \cup Cnst(t)$. If there is no such derivation or $F' \neq F$ then fail.

3. The judgments "$\xi^G : G$ – wf" can be derived from "G – wf" by one of Rules 3, 4, 5, 6, 8 or 10. Apply step 1 to all G's.

This algorithm constructs a valid derivation of "X – wf" or fails. It follows from Lemmas 3.3, 5.2, 5.3 that its failure means that X is not well-formed.

The polynomial time bound for step 2 is proved in Lemma 5.3, for steps 1 and 3 it is trivial. The number of iterations of 1,2,3-block is bound by the cardinality of $ASub(X)$ which is the size of X.

Typing problem has a similar polynomial time solution. Given a term t apply Lemma 5.2 and restore the formula F such that $Hyp(t) \vdash_{\mathsf{RLP}_{\to}} t : F$. It is the unique candidate for a formula denoting the type of t. Then try to calculate $(t : F)^r = t^r : F^r$ by the previous method and extract t^r from it. If something goes wrong then t is not typable. ■

6 Derivability in $\mathsf{RCL}_{\to}$

Now we consider the $\mathsf{RCL}_{\to}$-judgement "F is derivable from $F_1, \ldots, F_n$". It is defined in [4] for well-formed formulas $F_1, \ldots, F_n, F$ as the derivability from hypotheses in the following calculus.

$\mathsf{RCL}_{\to}$ (all axioms **(A1)** – **(A6)** are well-formed):

(A1) $t : F \to F$,

(A2) $d^{t:F\to F} : (t : F \to F)$,

(A3) $k^{F\to(G\to F)} : (F \to (G \to F))$,

(A4) $s^{(\ldots)} : ((F \to (G \to H)) \to ((F \to G) \to (F \to H)))$,

(A5) $o^{(\ldots)} : (u : (F \to G) \to (v : F \to (uv) : G))$,

(A6) $c^{(\ldots)} : (t : F \to !t : (t : F))$,

(MP) $\dfrac{F \to G \quad F}{G}$.

Comment. The well-formedness of axioms is a precondition: a formula can be used as an axiom when it is well-formed and has one of the forms **(A1)** –

(A6). We have seen that **(MP)** preserves well-formedness, so every formula in a valid derivation must be well-formed.

Our goal is to prove the decidability of the relation $F_1, \ldots, F_n \vdash_{\mathsf{RCL}_\rightarrow} F$. The first step is to reformulate $\mathsf{RCL}_\rightarrow$ in a sequent form. We give the sequent formulation for the case of types (i.e. well-formed formulas with explicit labelling of all subterms).

Definition 6.1 A sequent has the form $\Gamma \Rightarrow F$ where $F \in LFm$ and Γ is a finite multiset of labelled formulas. It is called well-formed when F and all members of Γ are types. A sequent derivation is called well-formed when all sequents in it are well-formed. The sequent calculus $\mathsf{RCLT_G}$ is defined as follows:

Axioms are all sequents of the form $P, \Gamma \Rightarrow P$ where P is a type variable p_i or a labelled formula of the form $t^G\!:\!G$.

Rules:

$$\frac{\Gamma \Rightarrow F \quad G, \Gamma \Rightarrow H}{F \rightarrow G, \Gamma \Rightarrow H}\,(L \rightarrow) \qquad \frac{F, \Gamma \Rightarrow G}{\Gamma \Rightarrow F \rightarrow G}\,(R \rightarrow)$$

$$\frac{F, F, \Gamma \Rightarrow G}{F, \Gamma \Rightarrow G}\,(L\,C) \qquad \frac{\Gamma \Rightarrow F \quad F, \Gamma' \Rightarrow G}{\Gamma, \Gamma' \Rightarrow G}\,(Cut)$$

$$\frac{F, \Gamma \Rightarrow G}{t^F\!:\!F, \Gamma \Rightarrow G}\,(L\ :) \qquad \frac{\Gamma \Rightarrow t^F\!:\!F}{\Gamma \Rightarrow (!t^F)^{t^F:F}\!:\!(t^F\!:\!F)}\,(R\,!)$$

$$\frac{\Gamma \Rightarrow s^{F\rightarrow G}\!:\!(F \rightarrow G) \quad \Gamma \Rightarrow t^F\!:\!F}{\Gamma \Rightarrow (s^{F\rightarrow G} \cdot t^F)^G\!:\!G}\,(R\,\cdot) \qquad \frac{\dfrac{\mathcal{D}}{\Gamma \Rightarrow A}}{\Gamma \Rightarrow i^A\!:\!A}\,(R\,i)$$

Rule $(R\ i)$ has a special form. Letter i in it denotes one of the identifiers k, s, d, o, c and A is an i-pattern. $\mathcal{D}$ depends on Γ, A and is the standard cut-free derivation of the sequent $\Gamma \Rightarrow A$ that is given below. The rule states that this particular derivation can be extended (in one step) to a derivation of the sequent $\Gamma \Rightarrow i^A\!:\!A$.

Comment. The prototype for this definition is the sequent formulation of Intuitionistic Logic of Proofs ILPG from [1, 3]. $\mathsf{RCLT_G}$ corresponds to its minimal implicative fragment. The latter can be considered as a completely forgetful projection of $\mathsf{RCLT_G}$ in which the superscripts and the correspondence between identifiers and patterns in $(R\ i)$ are omitted.

The cut-free derivations of k-patterns and s-patterns are well known. Here are the variants of $\mathcal{D}$ for d-, c- and o-patterns:

$$\dfrac{\dfrac{\dfrac{\cdots}{F,\, \Gamma \Rightarrow F}}{t^F\!:\!F,\, \Gamma \Rightarrow F}}{\Gamma \Rightarrow (t^F\!:\!F \to F)} \qquad \dfrac{\dfrac{\dfrac{\cdots}{t^F\!:\!F,\, \Gamma \Rightarrow t^F\!:\!F}}{t^F\!:\!F,\, \Gamma \Rightarrow (!t^F)^{t^F:F}\!:\!(t^F\!:\!F)}}{\Gamma \Rightarrow (t^F\!:\!F \to (!t^F)^{t^F:F}\!:\!(t^F\!:\!F))}$$

$$\dfrac{\dfrac{\dfrac{\cdots}{t^F\!:\!F,\, s^{F\to G}\!:\!(F\to G),\, \Gamma \Rightarrow s^{F\to G}\!:\!(F\to G)} \qquad \dfrac{\cdots}{t^F\!:\!F,\, s^{F\to G}\!:\!(F\to G),\, \Gamma \Rightarrow t^F\!:\!F}}{t^F\!:\!F,\, s^{F\to G}\!:\!(F\to G),\, \Gamma \Rightarrow (s^{F\to G}t^F)^G\!:\!G}}{\Gamma \Rightarrow (s^{F\to G}\!:\!(F\to G) \to (t^F\!:\!F \to (s^{F\to G}t^F)^G\!:\!G))}$$

Lemma 6.2 *A cut-free derivation of a well-formed sequent is well-formed.*

Proof. For every rule except (Cut) if the conclusion is well-formed then the premise(s) are well formed too. ∎

In particular, any well formed sequent of the form $F,\Gamma \Rightarrow F$ has a well-formed cut-free derivation.

Theorem 6.3 *1. If $\Gamma \vdash_{\mathsf{RCL}_\to} F$ then the sequent $\Gamma^r \Rightarrow F^r$ has a well-formed $\mathsf{RCLT_G}$-derivation.*
2. If a sequent $\Gamma \Rightarrow F$ has a well-formed $\mathsf{RCLT_G}$-derivation then $\Gamma^0 \vdash_{\mathsf{RCL}_\to} F^0$.

Proof. 1. Induction on the derivation $\Gamma \vdash_{\mathsf{RCL}_\to} F$. By the definition of $\vdash_{\mathsf{RCL}_\to}$, the formula F and all formulas from Γ are well-formed. So the sequent $\Gamma^r \Rightarrow F^r$ is well-formed by Theorem 4.3.

Case of **(A1)**. The required derivation is the cut-free derivation $\mathcal{D}$ involved in Rule $(R\ i)$ for $i = d$. It is well-formed by Lemma 6.2.

Cases of **(A2)** – **(A6)**. The sequent $\Gamma \Rightarrow F$ can be derived by Rule $(R\ i)$. This derivation is cut-free, so it is well-formed (Lemma 6.2).

Case of **(MP)**. By the induction hypothesis, the sequents $\Gamma^r \Rightarrow X^r \to Y^r$ and $\Gamma^r \Rightarrow X^r$ already have well-formed derivations. Refine these derivations as follows:

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma^r \Rightarrow X^r\end{array} \qquad \dfrac{\begin{array}{c}\vdots\\ \Gamma^r \Rightarrow X^r \to Y^r\end{array} \qquad \dfrac{X^r \Rightarrow X^r \quad Y^r, X^r \Rightarrow Y^r}{X^r \to Y^r, X^r \Rightarrow Y^r}}{X^r, \Gamma^r \Rightarrow Y^r}\,(Cut)}{\dfrac{\Gamma^r, \Gamma^r \Rightarrow Y^r}{\Gamma^r \Rightarrow Y^r}}\,(Cut)$$

2. Straightforward induction on well-formed derivation of a sequent $\Gamma \Rightarrow F$. Case $(R \to)$ is the standard deduction theorem which holds for $\mathsf{RCL}_\to$ because k-patterns and s-patterns are derivable in $\mathsf{RCL}_\to$ whenever they are well-formed. Other cases are trivial.

■

7 The relation $\vdash_{\mathsf{RCL}_\to}$ is decidable

Let a set $\Gamma \subset Fm$ be finite and $F \in Fm$. Note that the sequent $\Gamma^r \Rightarrow F^r$ exists and is well-formed iff all members of $\Gamma \cup \{F\}$ are well-formed. Theorem 6.3 gives the following reduction: $\Gamma \vdash_{\mathsf{RCL}_\to} F$ iff

(i) all members of $\Gamma \cup \{F\}$ are well-formed and

(ii) the sequent $\Gamma^r \Rightarrow F^r$ has a well-formed derivation.

Theorem 5.4 provides a polynomial time decision procedure for condition (i). Below we establish that (ii) is also decidable and prove the upper complexity bound PSPACE for it.

Lemma 7.1 *If a sequent has a well-formed derivation then it has a cut-free well-formed derivation too.*

Proof. (Sketch.) In [1] the syntactical cut-elimination method for Intuitionistic Logic of Proofs ILPG is proposed. A derivation of a sequent in $\mathsf{RCLT_G}$ can be considered as an ILPG-derivation in which all terms have additional labels (superscripts), so the method can be applied to an $\mathsf{RCLT_G}$-derivation as well. It can be shown that every step of the cut-elimination procedure preserves the property "to be a valid $\mathsf{RCLT_G}$-derivation". Thus, being applied to a well-formed $\mathsf{RCLT_G}$-derivation it returns a cut-free $\mathsf{RCLT_G}$-derivation of the same sequent. By Lemma 6.2, the resulting derivation is well-formed.

■

Definition 7.2 Let $\mathsf{RCLT'_G}$ denote the variant of the calculus $\mathsf{RCLT_G}$ without Rules (Cut) and $(R\,i)$ extended by the following additional axioms and rules:

$$(CS) \quad \Gamma \Rightarrow i^A : A, \quad i \in \{k, s, d, o, c\}, \ A \text{ is an } i\text{-pattern},$$

$$\frac{F \to G, \Gamma \Rightarrow F \quad G, F \to G, \Gamma \Rightarrow H}{F \to G, \Gamma \Rightarrow H}\,(L \to^C), \quad \frac{F, t^F : F, \Gamma \Rightarrow G}{t^F : F, \Gamma \Rightarrow G}\,(L :^C),$$

$$\frac{F, \Gamma \Rightarrow G}{F, \Gamma \Rightarrow F \to G}\,({}^W R \to).$$

Comment. The additional rules are admissible in $\mathsf{RCLT_G}$. The new axioms (CS) are derivable in $\mathsf{RCLT_G}$ and can emulate the applications of $(R\,i)$ rule. The cut elimination property for $\mathsf{RCLT_G}$ is established in Lemma 7.1. So the calculus $\mathsf{RCLT'_G}$ derives the same sequents as $\mathsf{RCLT_G}$.

Definition 7.3 For a multiset Γ let $set(\Gamma)$ denote the set of its members. An instance of a rule

$$\frac{\Gamma_1 \Rightarrow F_1 \ \ldots \ \Gamma_n \Rightarrow F_n}{\Gamma \Rightarrow F}$$

is called *monotone* if $set(\Gamma) \subseteq \cap_i set(\Gamma_i)$. A *monotone derivation* is a sequent derivation that uses monotone instances of inference rules only.

Lemma 7.4 *A sequent $\Gamma \Rightarrow F$ is provable in $\mathsf{RCLT_G}$ iff the sequent $set(\Gamma) \Rightarrow F$ has a monotone $\mathsf{RCLT'_G}$-derivation $\Gamma_1 \Rightarrow F_1, \ldots, \Gamma_n \Rightarrow F_n$ such that $\Gamma_i = set(\Gamma_i)$ (i.e. no labelled formula occurs in Γ_i twice).*

Proof. $\mathsf{RCLT_G} \vdash \Gamma \Rightarrow F$ iff $\mathsf{RCLT_G} \vdash set(\Gamma) \Rightarrow F$, so it is sufficient to consider the case when $\Gamma = set(\Gamma)$. The part "only if" is trivial ($\mathsf{RCLT_G}$ and $\mathsf{RCLT'_G}$ have the same provable sequences). Let us prove the remaining part "if". Consider a cut-free $\mathsf{RCLT_G}$-derivation of the sequent $\Gamma \Rightarrow F$ with $\Gamma = set(\Gamma)$.

Conversion to monotone derivation. Apply to the derivation the following transformation repeatedly, whenever it is possible. Find the last rule that adds a "new" labelled formula A to the antecedent of its conclusion. ("New" means that A is not a member of the antecedent of a premise of the rule.) Add the second copy of A to the antecedent of the conclusion. Also add a copy of A to the antecedents of all sequents above and insert $(L\,C)$ rule below:

$$\dfrac{\vdots}{A,\,\Gamma \Rightarrow F} \quad\longmapsto\quad \dfrac{\dfrac{\vdots}{A,\,A,\,\Gamma \Rightarrow F}}{A,\,\Gamma \Rightarrow F}\,(L\,C)$$

The result will be a monotone $\mathsf{RCLT_G}$-derivation of the same sequent. Now replace the occurrences of $(R\,i)$ rule by corresponding (CS) axioms. We shall also replace the consequent applications of Rules $(L \rightarrow)(L\,C)$ with the same main formula by $(L \rightarrow^C)$ and $(L\,:)(L\,C)$ with the same main formula – by $(L\,:^C)$. This will give a monotone $\mathsf{RCLT'_G}$-derivation of the initial sequent.

Conversion to sets in antecedents. For every labelled formula that occurs twice in the antecedent of a sequent from the derivation let us remove all but one copy of it from the antecedent of every sequent in the derivation. This transformation converts an axiom into some other axiom of $\mathsf{RCLT'_G}$. Let us see that an instance of a rule will be converted either into some other instance of some rule of $\mathsf{RCLT'_G}$ or into the the trivial rule that can be removed from the derivation (its conclusion coincides with one of the premises). Thus, the result will be a valid monotone $\mathsf{RCLT'_G}$-derivation and the condition $\Gamma' = set(\Gamma')$ holds for every sequent $\Gamma' \Rightarrow F'$ in it.

Indeed, the instances of $(R\,!)$, $(R\cdot)$ will be converted into some other instances of the same rule. An instance of $(L\,C)$ will be converted into the trivial rule:

$$\dfrac{F^{k+1},\,\Gamma \Rightarrow G}{F^k,\,\Gamma \Rightarrow G}\,(L\,C) \quad\longmapsto\quad \dfrac{F,\,\Delta \Rightarrow G}{F,\,\Delta \Rightarrow G} \quad\longmapsto\quad \text{remove.}$$

Rule $(L\,:)$. Let $\Gamma = (t^F\!:F)^n, \Gamma'$. If $F \notin \Gamma$ then

$$\dfrac{F,\,(t^F\!:F)^n,\,\Gamma' \Rightarrow G}{(t^F\!:F)^{n+1},\,\Gamma' \Rightarrow G} \quad\longmapsto\quad \dfrac{F,\,t^F\!:F,\,\Delta \Rightarrow G}{t^F\!:F,\,\Delta \Rightarrow G}\,(L\,:^C)$$

Otherwise, $\Gamma = F^k, (t^F : F)^n, \Gamma''$ and the result is trivial:

$$\frac{F^{k+1},\ (t^F : F)^n,\ \Gamma'' \Rightarrow G}{F^k,\ (t^F : F)^{n+1},\ \Gamma'' \Rightarrow G} \quad \longmapsto \quad \frac{F,\ t^F : F,\ \Delta \Rightarrow G}{F,\ t^F : F,\ \Delta \Rightarrow G} \quad \longmapsto \quad \text{remove.}$$

Rule $(L :^C)$.

$$\frac{F^{k+1},\ (t^F : F)^{n+1},\ \Gamma' \Rightarrow G}{F^k,\ (t^F : F)^{n+1},\ \Gamma' \Rightarrow G} \quad \longmapsto \quad \begin{cases} \dfrac{F,\ t^F : F,\ \Delta \Rightarrow G}{t^F : F,\ \Delta \Rightarrow G}\,(L :^C), & \text{if } k = 0, \\ \text{remove}, & \text{if } k > 0. \end{cases}$$

Rule $(R \to)$.

$$\frac{F^{k+1},\ \Gamma' \Rightarrow G}{F^k,\ \Gamma' \Rightarrow F \to G} \quad \longmapsto \quad \begin{cases} \dfrac{F,\ \Delta \Rightarrow G}{F,\ \Delta \Rightarrow F \to G}\,({}^W R \to), & \text{if } k > 0, \\ \dfrac{F,\ \Delta \Rightarrow G}{\Delta \Rightarrow F \to G}\,(R \to), & \text{if } k = 0. \end{cases}$$

Rule $(L \to)$. A monotone instance of this rule has the form

$$\frac{G^l,\ (F \to G)^{k+1},\ \Gamma' \Rightarrow F \quad G^{l+1},\ (F \to G)^{k+1},\ \Gamma' \Rightarrow H}{G^l,\ (F \to G)^{k+2},\ \Gamma' \Rightarrow H}.$$

If $l > 0$ then the transformation gives a trivial rule (its second premise coincides with the conclusion). It can be removed from the resulting derivation together with the derivation of the first premise. If $l = 0$ then the result will be an instance of $(L \to^C)$:

$$\frac{(F \to G)^{k+1},\ \Gamma' \Rightarrow F \quad G,\ (F \to G)^{k+1},\ \Gamma' \Rightarrow H}{(F \to G)^{k+2},\ \Gamma' \Rightarrow H} \quad \longmapsto$$

$$\longmapsto \quad \frac{F \to G,\ \Delta \Rightarrow F \quad G,\ F \to G,\ \Delta \Rightarrow H}{F \to G,\ \Delta \Rightarrow H}\,(L \to^C).$$

Similarly with rule $(L \to^C)$:

$$\frac{G^l,\ (F \to G)^{k+1},\ \Gamma' \Rightarrow F \quad G^{l+1},\ (F \to G)^{k+1},\ \Gamma' \Rightarrow H}{G^l,\ (F \to G)^{k+1},\ \Gamma' \Rightarrow H}$$

The result is either a trivial rule (case $l > 0$) or an instance of $(L \rightarrow^C)$ rule (case $i = 0$).

■

Definition 7.5 A labelled formula F is a *weak subformula* of a labelled formula G if $F \in ASub(G)$ or $F = t^H : H$ for some labelled term $t^H \in ASub(G)$.

Lemma 7.6 *Any labelled formula in an* $\mathsf{RCLT}'_\mathsf{G}$*-derivation of a sequent* $\Gamma \Rightarrow F$ *is a weak subformula of some formula* $G \in set(\Gamma) \cup \{F\}$*.*

Proof. For every rule of $\mathsf{RCLT}'_\mathsf{G}$ holds: if a labelled formula occurs in the premise(s) of the rule then it is a weak subformula of some formula from the conclusion. ■

Theorem 7.7 *For the calculus* RCLT_G *the set of all derivable sequents is decidable. The derivability relation* $\vdash_{\mathsf{RCL}_\rightarrow}$ *is decidable.*

Proof. Decidability of RCLT_G. By Lemmas 7.4, 7.6, $\mathsf{RCLT}_\mathsf{G} \vdash \Gamma \Rightarrow F$ iff there exists a finite sequence of sequents $\Gamma_1 \Rightarrow F_1, \ldots, \Gamma_n \Rightarrow F_n$ with the following properties:

1. All sequents are distinct from each other.
2. For every i, the sequent $\Gamma_i \Rightarrow F_i$ consists of weak subformulas of some formulas from $\Gamma \cup \{F\}$.
3. For every i, no labelled formula occurs in Γ_i twice.
4. It is a monotone $\mathsf{RCLT}'_\mathsf{G}$-derivation of the sequent $set(\Gamma) \Rightarrow F$.

There are finitely many sequences satisfying the conditions 1–3. One can try them all and check the condition 4.

Decidability of $\vdash_{\mathsf{RCL}_\rightarrow}$. By Theorem 6.3, $\Gamma \vdash_{\mathsf{RCL}_\rightarrow} F$ iff (i) all members of $\Gamma \cup \{F\}$ are well-formed and (ii) the sequent $\Gamma^r \Rightarrow F^r$ has a well-formed RCLT_G-derivation. By Theorem 5.4, the well-formedness condition (i) is decidable. It implies the existence and well-formedness of the sequent $\Gamma^r \Rightarrow F^r$. But if a well-formed sequent is derivable in RCLT_G then it has a well-formed RCLT_G-derivation too (Lemmas 7.1, 6.2). Thus, the condition (ii) can be replaced by $\mathsf{RCLT}_\mathsf{G} \vdash \Gamma^r \Rightarrow F^r$ which is decidable. ■

Comment. The decidability results (Theorem 7.7) can be proved by essentially the same method based on a more simple form of Lemma 7.4 with monotonicity condition omitted. The full-scale Lemma 7.4 is used in the proof of the complexity bound below.

8 Complexity bound

Both problems considered in Theorem 7.7, i.e. derivability of a sequent in $\mathsf{RCLT_G}$ and derivability of a formula from a set of hypotheses in $\mathsf{RCL}_{\rightarrow}$, are $PSPACE$-complete. The lower bound ($PSPACE$-hardness) follows from $PSPACE$-completeness result for implicational fragment of intuitionistic propositional logic proved in [11]. We prove the upper bound.

Definition 8.1 The *size of a sequent* $F_1, \ldots, F_n \Rightarrow F$ is the total sum of the sizes of all formulas F_i and F.

Lemma 8.2 *Let M_n be the set of all minimal monotone $\mathsf{RCLT}'_\mathsf{G}$-derivations $\mathcal{D}$ of sequents of size n with additional property: $\Gamma = set(\Gamma)$ holds for every sequent $\Gamma \Rightarrow F$ in $\mathcal{D}$. There exist two polynomials q_1 and q_2 such that the depth of every derivation $\mathcal{D} \in M_n$ is bound by $q_1(n)$ and the sizes of all sequents from $\mathcal{D}$ are bound by $q_2(n)$.*

Proof. Consider the proof tree for some $\mathcal{D} \in M_n$ and a path from the root to some leaf in it:

$$\Gamma_0 \Rightarrow F_0, \ldots, \Gamma_l \Rightarrow F_l.$$

All sequents in the path are distinct from each other. Let S be the set of all weak subformulas of formulas $G \in \Gamma_0 \cup \{F_0\}$. Every sequent in $\mathcal{D}$ consists of some formulas from S (Lemma 7.6). The number of elements in S is bound by n. Indeed, every weak subformula $G = t^H : H \notin ASub(F)$ of a formula F corresponds to the unique term $t^H \in ASub(F)$. Thus, $size(F)$ which is the total number of all associated subformulas and all associated subterms of F bounds the number of weak subformulas of F.

We split the path into maximal segments with Γ_i be constant inside a segment. All sequents in a segment have the same antecedent and various succedents $F \in S$. Thus, the length of a segment is bound by n. The number of segments is bound by n too because $\Gamma_0 \subseteq \Gamma_1 \ldots \subseteq \Gamma_l \subseteq S$. So, the length l of the path is bound by $q_1(n) = n^2$.

Consider a sequent $F_1, \ldots, F_m \Rightarrow F$ from $\mathcal{D}$. All formulas F_i and F belong to S, so their sizes are bound by n. But no formula occurs in the antecedent of the sequent twice, so $m \leq n$. Thus, the size of the sequent is bound by $q_2(n) = n(n+1)$.

■

Theorem 8.3 *For the calculus* $\mathsf{RCLT_G}$ *the set of all derivable sequents belongs to* $PSPACE$*.*

Proof. Let q_1, q_2 be polynomials from Lemma 8.2. Consider the following alternative two-person game. The initial configuration of the game (b_0) is a sequent $\Gamma \Rightarrow F$ of size n. The first player (I) writes down one or two sequents of sizes less than $q_2(n)$ and his opponent (II) chooses one of them, and so on. The game is over after $q_1(n)$ moves of (II) or when (II) chooses a sequent that is an axiom of $\mathsf{RCLT'_G}$.

Let w_i and b_i denote the moves of the players (I) and (II) respectively, so $b_0, w_1, b_1, b_2, w_2, ...$ is the protocol of the game. The player (I) wins if the following conditions are satisfied:

1. For every move of (I) the figure $\dfrac{w_i}{b_{i-1}}$ is a monotone instance of some inference rule of $\mathsf{RCLT'_G}$.
2. $\Delta = set(\Delta)$ holds for every sequent $\Delta \Rightarrow G$ from the protocol.
3. The last move is an axiom of $\mathsf{RCLT'_G}$.

The winning condition for this game is decidable in polynomial time. The number of moves and the size of a move are bound by polynomials too. For any such game the set

$$W = \{b_0 \mid \text{(I) has a winning strategy in a game started from } b_0\}$$

belongs to $PSPACE$ (see [6], [8]).

A sequent of size n belongs to W iff it has a monotone $\mathsf{RCLT'_G}$-derivation $\mathcal{D}$ such that the depth of $\mathcal{D}$ is bound by $q_1(n)$, for every sequent $\Delta \Rightarrow G$ in $\mathcal{D}$ holds $\Delta = set(\Delta)$ and the sizes of all these sequents are bound by $q_2(n)$. By Lemmas 7.4, 8.2, every sequent that is derivable in $\mathsf{RCLT_G}$ has an $\mathsf{RCLT'_G}$-derivation of this sort. But $\mathsf{RCLT_G}$ and $\mathsf{RCLT'_G}$ have the same derivable sequents. So $\mathsf{RCLT_G} \vdash \Gamma \Rightarrow F$ iff $(\Gamma \Rightarrow F) \in W$.

■

Corollary 8.4 *The derivability problem for the calculus* $\mathsf{RCLT_G}$ *and the relation* $\vdash_{\mathsf{RCL}_\to}$ *(derivability from hypotheses in* $\mathsf{RCL}_\to$*) are* $PSPACE$*-complete.*

Proof. The proof of Theorem 7.7 provides a polynomial time reduction of the second problem to the first one. So the relation $\vdash_{\mathsf{RCL}_\to}$ belongs to $PSPACE$ too. It remains to prove the lower complexity bound: $\vdash_{\mathsf{RCL}_\to}$ is $PSPACE$-hard.

Consider a term-free fragment of $\mathsf{RCL}_\to$, i.e. the set of all propositional formulas that are provable in $\mathsf{RCL}_\to$. It coincides with the implicational fragment $\mathsf{Int}_\to$ of intuitionistic propositional logic. Indeed, every propositional formula F is well-formed and $F^r = F$, so $\vdash_{\mathsf{RCL}_\to} F$ iff the sequent $\Rightarrow F$ has a cut-free $\mathsf{RCLT_G}$-derivation. This derivation is nothing but a derivation in the cut-free sequent formulation of $\mathsf{Int}_\to$. The $PSPACE$-hardness of $\mathsf{Int}_\to$ is proved in [11].

■

Acknowledgments

I would like to thank professor S. Artëmov, who advised me on this work, for his support and helpful discussions on the subject.

The research described in this paper was partially supported by Russian Foundation for Basic Research.

References

[1] S. Artemov. Logic of proofs: a unified semantics for modality and λ-terms. Technical Report CFIS 98-06, Cornell University, 1998

[2] S. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1):1–36, 2001.

[3] S. Artemov. Unified semantics for modality and lambda-terms via proof polynomials. In /Kees Vermeulen and Ann Copestake/ eds. Algebras, Diagrams and Decisions in Language, Logic and Computation, CSLI Publications, Stanford University, 2002.

[4] S. Artemov. Kolmogorov and Gödel's approach to intuitionistic logic: current developments. *Russian Mathematical Surveys*, 59(2):203–229, 2004.

[5] M. Bidoit, J. Corbin. A Rehabilitation of Robinson's Unification Algorithm. *Information Processing*, 83:909:-914, 1983.

[6] A.K. Chandra, D.C. Kozen, L.J. Stockmeyer. Alternation. J. Assoc. Comput. Mach., v. 28, 1981, pp 114-133.

[7] R.L. Constable. Types in logic, mathematics and programming. In S.R. Buss, ed., *Handbook of proof theory*, chapter X, Elsevier Science B.V., 1998, pp. 684-786

[8] A. Kitaev, A. Shen, M. Vyalyi. Classical and quantum computations. Moscow: MCCME, CheRo, 1999 (in Russian).

[9] N. Krupski. On the complexity of the reflected logic of proofs. CUNY Ph.D Program in CS Technical Reports, TR-2003007, 2003.

[10] P. Martin-Löf. An intuitionistic theory of types. In G. Sambin, J. Smith, eds., *Twenty-five years of Constructive Type Theory*, Oxford Logic Guides, v. 36, 1998, pp. 127-172

[11] R. Statman. Intuitionistic propositional logic is polynomial-space complete. Theoretical Comput. Sci. 9, 1979, pp. 67-72

[12] A. S. Troelstra, H. Schwichtenberg. *Basic proof theory*. Cambridge: Cambridge Univ. Press, 1996.