*Journal of Mathematics and Statistics*

Original Research Paper

# Formalizing Relations in Type Theory

**Farida Kachapova**

*Department of Mathematical Sciences, Auckland University of Technology, New Zealand*

*Article history*
Received: 02-03-2022
Revised: 21-07-2022
Accepted: 01-09-2022

Email: farida.kachapova@aut.ac.nz

**Abstract:** Type theory plays an important role in the foundations of mathematics as a framework for formalizing mathematics and a base for proof assistants providing semi-automatic proof checking and construction. Derivation of each theorem in type theory results in a formal term encapsulating the whole proof process. This study uses a variant of type theory, namely the Calculus of Constructions with Definitions, to formalize the standard theory of binary relations. This includes basic operations on relations, criteria for special properties of relations, invariance of these properties under the basic operations, equivalence relation, well-ordering, and transfinite induction. Definitions and proofs are presented as flag-style derivations.

**Keywords:** Type Theory, Calculus of Constructions, Binary Relation, Transfinite Induction, Flag-Style Derivation

## 1. Introduction

Type theories were developed as alternatives to set theory for the foundation of mathematics. Important type theories were introduced by A. Church and P. Martin-Lof; they are typed λ-calculus (see, for example, Barendregt (2012)) and intuitionistic type theory (see, for example, Granstrom (2011)). There are several higher-order variants of typed λ-calculus, such as Calculus of Constructions (CoC) and Calculus of Inductive Constructions (CIC) (Bertot and Casteran, 2013). These variants make formal bases of proof assistants, which are computer tools for formalizing and developing mathematics. In particular, the well-known proof assistant Coq (Coq Development Team, 2021) is based on the CIC.

This study uses the variant λ*D* of CoC developed by Nederpelt and Geuvers (2014); λ*D* is called the Calculus of Constructions with Definitions. λ*D* is chosen because of its following useful properties described in their book.

– In λ*D*, as in other variants of CoC, proofs are expressed as formal terms and thus are incorporated into the system.
– In λ*D* type checking is decidable and therefore proof checking is decidable. So the correctness of a proof can be checked by an algorithm.
– λ*D* is strongly normalizing, which implies the logical consistency of this theory, even with classical logic (when no extra axioms are added).

The theory λ*D* is weaker than CIC because λ*D* does not have inductive types. This does not limit its capability for formalizing mathematics because in λ*D* one can use an axiomatic approach and higher-order logic to express the objects that CIC defines with inductive types.

In these formalizations, the author aims to keep the language and theorems as close as possible to the ones of standard mathematics. Definitions and proofs in this study use the flag-style derivation described in Nederpelt and Geuvers (2014). Long formal derivations are moved from the main text to Appendices for better readability.

## 2. Type Theory λD

Nederpelt and Geuvers (2014) developed a formal theory λ*D* and formalized some parts of logic and mathematics in it. The main features of λ*D* are briefly described below.

### 2.1. Type Theory λD

The language of λ*D* described in Nederpelt and Geuvers (2014) has an infinite set of variables, *V*, and an infinite set of constants, *C*; these two sets are disjoint. There are also special symbols □ and ∗.

*Definition 2.1*

**Expressions** of the language are defined recursively as follows.

(1) Each variable is an expression.
(2) Each constant is an expression.
(3) The constant * is an expression.
(4) The constant □ is an expression.
(5) (Application) If A and B are expressions, then AB is an expression.

**DOI: 10.3844/jmssp.2022.148.162**

(6) (Abstraction) If A and B are expressions and x is a variable, then λx: A.B is an expression.
(7) (Dependent Product) If A and B are expressions and x is a variable, then Πx: A.B is an expression.
(8) If $A_1, A_2, ..., A_n$ are expressions and c is a constant, then $c(A_1, A_2, ..., A_n)$ is an expression.

An expression $A \to B$ is introduced as a particular type of Dependent Product from (7) when $x$ is not a free variable in $B$.

*Definition 2.2*

(1) *A **statement** is of the form M: N, where M and N are expressions.*
(2) *A **declaration** is of form x: N, where x is a variable and N is an expression.*
(3) *A **descriptive definition** is of the form:*

$$\bar{x} : \bar{A} \rhd c(\bar{x}) : = M : N,$$

*where, $\bar{x}$ is a list $x_1, x_2, ..., x_n$ of variables, $\bar{A}$ is a list $A_1, A_2, ..., A_n$ of expressions, c is a constant, and M and N are expressions.*
(4) *A **primitive definition** is of the form:*

$$\bar{x} : \bar{A} \rhd c(\bar{x}) := \perp\!\!\!\perp : N,$$

where, $\bar{x}$, $\bar{A}$, and $c$ are described the same way as in (3), and $N$ is an expression. The symbol $\perp\!\!\!\perp$ denotes the non-existing definiens. Primitive definitions are used for introducing axioms where no proof terms are needed.

(5) *A **definition** is a descriptive definition or a primitive definition.*
(6) *A **judgment** is of the form:*

$$\Delta; \Gamma \vdash M : N,$$

where, $M$ and $N$ are expressions of the language, Δ is an environment (a properly constructed sequence of definitions) and Γ is a context (a properly constructed sequence of declarations).

For brevity, most definitions use implicit variables by omitting the previously declared variables $\bar{x}$ in $c(\bar{x})$ in (3) and (4).

The following informally explains the meaning of expressions.

(1) If an expression $M$ appears in a derived statement of the form $M$: $*$, then $M$ is interpreted as a **type**, which represents a set or a proposition.

*Note*: There is only one type $*$ in λ*D*. But informally $*_p$ is often used for propositions and $*_s$ for sets to make proofs more readable.

(2) If an expression $M$ appears in a derived statement of the form $M$: $N$, where $N$ is a type, then $M$ is interpreted as an object at the lowest level.
When $N$ is interpreted as a set, then $M$ is regarded as an element of this set.
When $N$ is interpreted as a proposition, then $M$ is regarded as a proof (or a proof term) of this proposition.
(3) The symbol □ represents the highest level.
(4) **Sort** is $*$ or. Letters $s, s_1, s_2, ...$ are used as variables for sorts.
(5) If an expression $M$ appears in a statement of the form $M$: □, then $M$ is called a **kind.**

λ*D* contains the derivation rule:

$$\phi; \phi \ \vdash *: \square,$$

It is an axiom (the only axiom) of λ*D*, because it has an empty environment and an empty context.

Further details of the language and derivation rules of the theory λ*D* can be found in Nederpelt and Geuvers (2014). Judgments are formally derived in λ*D* using the derivation rules.

## *2.2. Flag Format of Derivations*

The flag-style deduction was introduced by Jaskowski and Fitch; it is described in detail by Nederpelt and Kamareddine (2004), and Nederpelt and Geuvers (2014). In short, a derivation in the flag format is a linear deduction. Each "flag" (a rectangular box) contains a declaration that introduces a variable or an assumption; a collection of already introduced variables and assumptions makes the current context. The scope of the variable or assumption is established by the "flag pole". In the scope, one constructs definitions and proof terms for proving statements/theorems in λ*D*. Each new flag extends the context and at the end of each flag pole, the context is reduced by the corresponding declaration. For brevity, several declarations can be combined in one flag.

## *2.3. Logic in λD*

The rules of intuitionistic logic are derived from the theory λ*D* as shown in Nederpelt and Geuvers (2014). These are briefly described below by showing the introduction and elimination rules for logical connectives and quantifiers.

### *Implication*

The logical implication $A \Rightarrow B$ is identified with the arrow type $A \to B$. The rules for implication follow the following general rules for the arrow type (here they are written in the flag format):

$$\begin{array}{l}
\boxed{\mathbf{var}\ A : s_1 \mid B : s_2} \\
\quad A \to B : s_2 \\
\quad \boxed{u : A \to B \mid v : A} \\
\quad\quad uv : B \\
\quad \boxed{\mathbf{var}\ x : A} \\
\quad\quad \dots \\
\quad\quad M : B \\
\quad \lambda x : A.M \ : \ A \to B
\end{array}$$

Here $x$ is not a free variable in $B$.

In $\lambda D$ arrows are right-associative, that is $A \to B \to C$ is a shorthand for $A \to (B \to C)$.

### *Falsity and Negation*

Falsity $\perp$ is introduced in $\lambda D$ by: $\perp := \Pi A : *_p.A : *_p$.

From this definition one gets a rule for falsity:

$$\begin{array}{l}
\boxed{\mathbf{var}\ B : *_p} \\
\quad \dots \\
\quad u : \perp \\
\quad u : \Pi A : *_p.A \\
\quad uB : B
\end{array}$$

The rule states that falsity implies any proposition.

As usual, negation is defined by: $\neg A: = A \to \perp$.

Other logical connectives and quantifiers are also defined using second-order encoding. Here only a list of their derived rules and names of the corresponding terms are provided, without details of their construction. The exact values of the terms can be found in Nederpelt and Geuvers (2014).

Some of our flag derivations contain the proof terms that will be re-used in other proofs; such proof terms are written in bold font, e.g., **∧-in** in the first derived rule for conjunction as follows.

### *Conjunction*

These are derived rules for conjunction ∧:

$$\begin{array}{l}
\boxed{\mathbf{var}\ A, B : *_p} \\
\quad \boxed{u : A \mid v : B} \\
\quad\quad \wedge\text{-}\mathbf{in}(A, B, u, v) \ : \ A \wedge B \\
\quad \boxed{w : A \wedge B} \\
\quad\quad \wedge\text{-}\mathbf{el}_1(A, B, w) \ : \ A \\
\quad\quad \wedge\text{-}\mathbf{el}_2(A, B, w) \ : \ B
\end{array}$$

### *Disjunction*

These are derived rules for disjunction ∨:

$$\begin{array}{l}
\boxed{\mathbf{var}\ A, B : *_p} \\
\quad \boxed{u : A} \\
\quad\quad \vee\text{-}\mathbf{in}_1(A, B, u) \ : \ A \vee B \\
\quad \boxed{u : B} \\
\quad\quad \vee\text{-}\mathbf{in}_2(A, B, u) \ : \ A \vee B \\
\quad \boxed{C : *_p} \\
\quad\quad \boxed{u : A \vee B \mid v : A \Rightarrow C \mid w : B \Rightarrow C} \\
\quad\quad\quad \vee\text{-}\mathbf{el}(A, B, C, u, v, w) \ : \ C
\end{array}$$

### *Bi-Implication*

Bi-implication ⇔ has the standard definition:

$$(A \Leftrightarrow B) := (A \Rightarrow B) \wedge (B \Rightarrow A).$$

### *Lemma 2.3.*

This lemma will be often used to prove bi-implication $A \Leftrightarrow B$.

$$\begin{array}{l}
\boxed{\mathit{var}\ A, B : *_p} \\
\quad \boxed{u : A \Rightarrow B \mid v : B \Rightarrow A} \\
\quad\quad \boldsymbol{\mathit{bi\text{-}impl}}(A, B, u, v) := \wedge\text{-}\mathit{in}(A \Rightarrow B, B \Rightarrow A, u, v) \ : \ A \Leftrightarrow B
\end{array}$$

### *Universal Quantifier*

The universal quantifier ∀ is defined through the dependent product:

$$\begin{array}{l}
\boxed{\mathbf{var}\ S : *_s \mid P : S \to *_p} \\
\quad \text{Definition } \forall(S, P) := \Pi x : S.Px \ : \ *_p \\
\quad \text{Notation} : (\forall \mathbf{x} : \mathbf{S}.\mathbf{Px}) \text{ for } \forall(S, P)
\end{array}$$

### *Existential Quantifier*

These are derived rules for the existential quantifier ∃:

$$\begin{array}{l}
\boxed{\mathbf{var}\ S : *_s \mid P : S \to *_p} \\
\quad \boxed{\mathbf{var}\ y : S \mid u : Py} \\
\quad\quad \exists\text{-}\mathbf{in}(S, P, y, u) \ : \ (\exists x : S.Px) \\
\quad \boxed{C : *_p} \\
\quad\quad \boxed{u : (\exists x : S.Px) \mid v : (\forall x : S.(Px \Rightarrow C))} \\
\quad\quad\quad \exists\text{-}\mathbf{el}(S, P, u, C, v) \ : \ C
\end{array}$$

Here $x$ is not a free variable in $C$.

### *Classical Logic*

This study uses mostly intuitionistic logic. But sometimes classical logic is needed; in these cases, the following Axiom of Excluded Third is added:

$$\begin{array}{l}
\boxed{\mathbf{var}\ A : *_p} \\
\quad \mathbf{exc\text{-}thrd}(A) := \perp\!\!\!\perp \ : \ A \vee \neg A
\end{array}$$

**DOI: 10.3844/jmssp.2022.148.162**

This axiom implies the Double Negation theorem:

$$\textbf{var } A : *_p$$
$$\textbf{doub-neg}(A) : (\neg\neg A \Rightarrow A)$$

## 3. Intensional Equality in λD

This section introduces intensional equality for elements of any type; it is called just equality. The next section will introduce extensional equality and the axiom of extensionality relating to the two types of equality.

$$\textbf{var } S : *$$
$$\textbf{var } x, y : S$$
$$eq(S, x, y) := \Pi P : S \to *_p . (Px \Rightarrow Py) : *_p$$
$$\text{Notation} : x =_S y \text{ for } eq(S, x, y) \qquad \textbf{Intensional equality}$$

### *3.1. Properties of Equality*

*Reflexivity*

The following diagram proves the reflexivity property of equality in λ*D*:

$$\textbf{var } S : * \mid x : S$$
$$\textbf{var } P : S \to *_p$$
$$Px : *_p$$
$$a_1 := \lambda u : Px . u : Px \Rightarrow Px$$
$$eq\text{-}refl(S, x) = \lambda P : S \to *_p . a_1 : (\Pi P : S \to *_p . (Px \Rightarrow Px))$$
$$\boldsymbol{eq\text{-}refl}(S, x) : x =_S x$$

Proof terms are constructed similarly for the following properties of Substitutivity, Congruence, Symmetry, and Transitivity (see Nederpelt and Geuvers (2014)).

*Substitutivity*

Substitutivity means that equality is consistent with predicates of corresponding types.

$$\textbf{var } S : *$$
$$\textbf{var } P : S \to *_p$$
$$\textbf{var } x, y : S \mid u : x =_S y \mid v : Px$$
$$\boldsymbol{eq\text{-}subs}(S, P, x, y, u, v) : Py$$

*Congruence*

Congruence means that equality is consistent with functions of corresponding types.

$$\textbf{var } Q, S : *$$
$$\textbf{var } f : Q \to S$$
$$\textbf{var } x, y : Q \mid u : x =_Q y$$
$$\boldsymbol{eq\text{-}cong}(Q, S, f, x, y, u) : fx =_S fy$$

*Symmetry*

The following diagram expresses the symmetry property of equality in λ*D*.

$$\textbf{var } S : *$$
$$\textbf{var } x, y : S \mid u : x =_S y$$
$$\boldsymbol{eq\text{-}sym}(S, x, y, u) : y =_S x$$

*Transitivity*

The following diagram expresses the transitivity property of equality in λ*D*.

$$\textbf{var } S : *$$
$$\textbf{var } x, y, z : S \mid u : x =_S y \mid v : y =_S z$$
$$\boldsymbol{eq\text{-}trans}(S, x, y, z, u, v) : x =_S z$$

## 4. Relations in Type Theory

### *4.1. Sets in λD*

Below are some definitions from Nederpelt and Geuvers (2014) relating to sets, in particular, subsets of type *S*:

$$\textbf{var } S : *_s$$
$$\boldsymbol{ps}(S) := S \to *_p \qquad \textbf{Power set of S}$$
$$\textbf{var } V : ps(S)$$
$$\text{Notation: } \{x : S \mid x \varepsilon V\} \text{ for } \lambda x : S . Vx$$
$$\textbf{var } x : S$$
$$\boldsymbol{element}(S, x, V) := Vx : *_p$$
$$\text{Notation: } x \varepsilon_S V \text{ or } x \varepsilon V \text{ for } element(S, x, V)$$

Thus, a subset *V* of *S* is regarded as a predicate on *S* and *x*ε*V* means *x* satisfies the predicate *V*.

### *4.2. Defining Binary Relations in λD*

Binary relations are introduced in Nederpelt and Geuvers (2014), together with the properties of reflexivity, symmetry, antisymmetry, and transitivity, and definitions of equivalence relation and partial order. These are used here as a starting point for formalizing the theory of binary relations in λ*D*.

A relation on *S* is a binary predicate on *S*, which is regarded in λ*D* as a composition of unary predicates. The type *br*(*S*) of all binary relations on *S* is introduced below, for brevity:

$$\textbf{var } S : *_s$$
$$\text{Definition } \boldsymbol{br}(\boldsymbol{S}) := S \to S \to *_p \ : \ \square$$

In the rest of the article, binary relations are called just relations. The equality of relations and operations on relations are defined similarly to the set equality and set operations.

Next, the extensional equality of relations is defined vs the intentional equality introduced in the previous section.

```
var S : *_s
  var R, Q : br(S)
    Definition ⊆(S, R, Q) := (∀x, y : S.(Rxy ⇒ Qxy)) : *_p
    Notation : R ⊆ Q for ⊆(S, R, Q)
    Definition Ex-eq(S, R, Q) := R ⊆ Q ∧ Q ⊆ R : *_p
    Notation : R = Q for Ex-eq(S, R, Q)                    Extensional equality
```

The following axiom of extensionality for relations is added to the theory $\lambda D$.

```
var S : *_s
  var R, Q : br(S)
    u : R = Q
      ext-axiom(S, R, Q, u) := ⫫ : R =_br(S) Q    Extensionality Axiom
```

The axiom is introduced in the last line by a primitive definition with the symbol ⫫ replacing a non-existing proof term. The Extensionality Axiom states that the two types of equality are the same for binary relations. So the symbol = will be used for both without elaborating on details of applying the axiom of extensionality, when converting one type of equality to the other.

### *4.3. Operations on Binary Relations*

The flag format is used to introduce the identity relation $id_S$ on type $S$ and converse $R^{-1}$ of a relation $R$:

```
var S : *_s
  Definition id_S := λx, y : S.(x =_S y) : br(S)        Identity relation
  var R : br(S)
    Definition conv(S, R) := λx, y : S.(Ryx) : br(S)
    Notation : R^-1 for conv(S, R)                      Converse relation
```

Next, the operations of union ∪, intersection ∩, and composition ∘ of relations are introduced:

```
var S : *_s
  var R, Q : br(S)
    Definition ∪(S, R, Q) := λx, y : S.(Rxy ∨ Qxy) : br(S)
    Notation : R ∪ Q for ∪(S, R, Q)                                Union
    Definition ∩(S, R, Q) := λx, y : S.(Rxy ∧ Qxy) : br(S)
    Notation : R ∩ Q for ∩(S, R, Q)                                Intersection
    Definition ∘(S, R, Q) := λx, y : S.(∃z : S.(Rxz ∧ Qzy)) : br(S)
    Notation : R ∘ Q for ∘(S, R, Q)                                Composition
```

### *4.4. Properties of Operations*

The following two technical lemmas will be used in some future proofs.

*Lemma 4.1.*

This lemma gives a shortcut for constructing an element of a composite relation.

```
var S : *_s | R, Q : br(S) | x, y, z : S
  u : Rxy | v : Qyz
    a := ∧-in (Rxy, Qyz, u, v) : Rxy ∧ Qyz
    prod-term (S, R, Q, x, y, z, u, v) := ∃-in (S, λt.Rxt ∧ Qtz, y, a) : (R ∘ Q)xz
```

*Lemma 4.2.*

This lemma gives a shortcut for proving the equality of two relations:

```
var S : *_s | R, Q : br(S)
  u : R ⊆ Q | v : Q ⊆ R
    rel-equal(S, R, Q, u, v) := ∧-in (R ⊆ Q, Q ⊆ R, u, v) : R = Q
```

*Theorem 4.3.*

For relations $R$, $P$, and $Q$ on type $S$ the following hold:

1) $(R^{-1})^{-1} = R$
2) $(R \circ Q)^{-1} = R$
3) $(R \cap Q)^{-1} = R^{-1} \cap Q^{-1}$
4) $(R \cup)^{-1} = R^{-1} \cup Q^{-1}$
5) $R \circ (P \cup Q) = R \circ P \cup R \circ Q$
6) $(P \cup Q) \circ R = P \circ R \cup Q \circ R$
7) $R \circ (P \cap Q) \subseteq R \circ P \cap R \circ Q$
8) $(P \cap Q) \circ R \subseteq P \circ R \cap Q \circ R$
9) $(R \circ Q) \circ Q = R \circ (P \circ Q)$

The formal proof is in Appendix A. The proof of part 2) has the form:

```
var S : *_s | R, Q : br(S)
  ...
  conv-prod(S, R, Q) := ... : (R ∘ Q)^-1 = Q^-1 ∘ R^-1
```

Its proof term *conv-prod* ($S$, $R$, $Q$) will be re-used later in the paper.

## 5. Properties of Binary Relations

The properties of reflexivity, symmetry, antisymmetry, transitivity and the relations of equivalence and partial order are defined in Nederpelt and Geuvers (2014) as follows.

```
var S : *_s | R : br(S)
  Definition refl(S, R) := ∀x : S.(Rxx) : *_p
  Definition sym(S, R) := ∀x, y : S.(Rxy ⇒ Ryx) : *_p
  Definition antisym(S, R) := ∀x, y : S.(Rxy ⇒ Ryx ⇒ x =_S y) : *_p
  Definition trans(S, R) := ∀x, y, z : S.(Rxy ⇒ Ryz ⇒ Rxz) : *_p
  Definition equiv-relation(S, R) := refl(S, R) ∧ sym(S, R) ∧ trans(S, R) : *_p
  Definition part-ord(S, R) := refl(S, R) ∧ antisym(S, R) ∧ trans(S, R) : *_p
```

*Theorem 5.1.*

Suppose $R$ is a relation on type $S$. Then the following hold.

1) **Criterion of reflexivity**. $R$ is reflexive ⇔ $id_S \subseteq R$.

2) **First criterion of symmetry**. $R$ is symmetric ⇔ $R^{-1} \subseteq R$.
3) **Second criterion of symmetry**. $R$ is symmetric ⇔ $R^{-1} = R$.
4) **Criterion of antisymmetry**. $R$ is antisymmetric ⇔ $R^{-1} \cap R \subseteq id_S$.
5) **Criterion of transitivity**. $R$ is transitive ⇔ $R \circ R \subseteq R$.

The formal proof is in Appendix B. The proof of part 3) has the form:

$\mathbf{var}\ S : *_s \mid R : br(S)$

$\quad \ldots$

$\quad \boldsymbol{sym\text{-}criterion}(S, R) := \ldots \ : \ sym(S, R) \Leftrightarrow R^{-1} = R$

Its proof term *sym-criterion* ($S$, $R$) will be re-used later in the paper.

*Theorem 5.2.*

Relation $R$ on $S$ is reflexive, symmetric, and antisymmetric ⇒ $R = id_S$.

*Proof*. The formal proof is in the following flag diagram.

$\mathbf{var}\ S : *_s \mid R : br(S)$

$\quad u_1 : refl(S, R) \mid u_2 : sym(S, R) \mid u_3 : antisym(S, R)$

$\quad\quad \mathbf{var}\ x, y : S \mid v : Rxy$

$\quad\quad\quad a_1 = u_2xyv : Ryx$

$\quad\quad\quad a_2 = u_3xyva_1 : x =_S y$

$\quad\quad a_3 := \lambda x, y : S.\lambda v : Rxy.a_2 \ : \ (R \subseteq id_S)$

$\quad\quad \mathbf{var}\ x, y : S \mid v : (id_S)xy$

$\quad\quad\quad v : x =_S y$

$\quad\quad\quad \text{Notation } P := \lambda z : S.Rxz \ : \ S \to *_p$

$\quad\quad\quad a_4 = u_1x : Rxx$

$\quad\quad\quad a_4 : Px$

$\quad\quad\quad a_5 := eq\text{-}subs(S, P, x, y, v, a_4) : Py$

$\quad\quad\quad a_5 : Rxy$

$\quad\quad a_6 := \lambda x, y : S.\lambda v : (id_S)xy.a_5 \ : \ (id_S \subseteq R)$

$\quad\quad a_7 := rel\text{-}equal(S, R, id_S, a_3, a_6) : R = id_S$

□

*Theorem 5.3.* **Invariance under converse operation.**

Suppose $R$ is a relation on type $S$. Then the following hold:

1) *$R$ is reflexive ⇒ $R^{-1}$ is reflexive*
2) *$R$ is symmetric ⇒ $R^{-1}$ is symmetric*
3) *$R$ is antisymmetric ⇒ $R^{-1}$ is antisymmetric*
4) *$R$ is transitive ⇒ $R^{-1}$ is transitive*

*Proof*. 1)

$\mathbf{var}\ S : *_s \mid R : br(S)$

$\quad u : refl(S, R)$

$\quad\quad \mathbf{var}\ x : S$

$\quad\quad\quad ux : Rxx$

$\quad\quad\quad ux : R^{-1}xx$

$\quad\quad a := \lambda x : S.ux \ : \ refl(S, R^{-1})$

2)

$\mathbf{var}\ S : *_s \mid R : br(S)$

$\quad u : sym(S, R)$

$\quad\quad \mathbf{var}\ x, y : S \mid v : R^{-1}xy$

$\quad\quad\quad v : Ryx$

$\quad\quad\quad uyx : (Ryx \Rightarrow Rxy)$

$\quad\quad\quad a_1 := uyxv \ : \ Rxy$

$\quad\quad\quad a_1 : R^{-1}yx$

$\quad\quad a_2 := \lambda x, y : S.\lambda v : R^{-1}xy.a_1 \ : \ sym(S, R^{-1})$

3)

$\mathbf{var}\ S : *_s \mid R : br(S)$

$\quad u : antisym(S, R)$

$\quad\quad \mathbf{var}\ x, y : S \mid v : R^{-1}xy \mid w : R^{-1}yx$

$\quad\quad\quad v : Ryx$

$\quad\quad\quad w : Rxy$

$\quad\quad\quad uxy : (Rxy \Rightarrow Ryx \Rightarrow x = y)$

$\quad\quad\quad a_1 := uxywv \ : \ x = y$

$\quad\quad a_2 := \lambda x, y : S.\lambda v : R^{-1}xy.\lambda w : R^{-1}yx.a_1 \ : \ antisym(S, R^{-1})$

4)

$\mathbf{var}\ S : *_s \mid R : br(S)$

$\quad u : trans(S, R)$

$\quad\quad \mathbf{var}\ x, y, z : S \mid v : R^{-1}xy \mid w : R^{-1}yz$

$\quad\quad\quad w : Rzy$

$\quad\quad\quad v : Ryx$

$\quad\quad\quad uzyx : (Rzy \Rightarrow Ryx \Rightarrow Rzx)$

$\quad\quad\quad a_1 := uzyxwv \ : \ Rzx$

$\quad\quad\quad a_1 : R^{-1}xz$

$\quad\quad a_2 := \lambda x, y, z : S.\lambda v : R^{-1}xy.\lambda w : R^{-1}yz.a_1 \ : \ trans(S, R^{-1})$

□

*Theorem 5.4.* **Invariance under intersection.**

Suppose $R$ and $Q$ are relations on type $S$. Then the following hold.

1) $R$ and $Q$ are reflexive ⇒ $R \cap Q$ is reflexive.
2) $R$ and $Q$ are symmetric ⇒ $R \cap Q$ is symmetric.
3) $R$ or $Q$ is antisymmetric ⇒ $R \cap Q$ is antisymmetric.
4) $R$ and $Q$ are transitive ⇒ $R \cap Q$ is transitive.

**DOI: 10.3844/jmssp.2022.148.162**

*Proof.* 1)

$\mathbf{var}\ S : *_s \mid R, Q : br(S)$
$u : refl(S, R) \mid v : refl(S, Q)$
$\mathbf{var}\ x : S$
$a_1 := ux \ : \ Rxx$
$a_2 := vx \ : \ Qxx$
$a_3 := \wedge\text{-in}\ (Rxx, Qxx, a_1, a_2) \ : \ (R \cap Q)xx$
$a_4 := \lambda x : S.a_3 \ : \ refl(S, R \cap Q)$

2)

$\mathbf{var}\ S : *_s \mid R, Q : br(S)$
$u : sym(S, R) \mid v : sym(S, Q)$
$\mathbf{var}\ x, y : S \mid w : (R \cap Q)xy$
$w : Rxy \wedge Qxy$
$a_1 := \wedge\text{-el}_1(Rxy, Qxy, w) \ : \ Rxy$
$a_2 := \wedge\text{-el}_2(Rxy, Qxy, w) \ : \ Qxy$
$a_3 := uxya_1 \ : \ Ryx$
$a_4 := vxya_2 \ : \ Qyx$
$a_5 := \wedge\text{-in}\ (Ryx, Qyx, a_3, a_4) \ : \ (R \cap Q)yx$
$a_6 := \lambda x, y : S.\lambda w : (R \cap Q)xy.a_5 \ : \ sym(S, R \cap Q)$

3)

$\mathbf{var}\ S : *_s \mid R, Q : br(S)$
Notation $A := antisym(S, R) : *_p$
Notation $B := antisym(S, Q) : *_p$
Notation $C := antisym(S, R \cap Q) : *_p$
$u : A \vee B$
$v : A$
$\mathbf{var}\ x, y : S \mid w_1 : (R \cap Q)xy \mid w_2 : (R \cap Q)yx$
$w_1 : Rxy \wedge Qxy$
$a_1 := \wedge\text{-el}_1(Rxy, Qxy, w_1) \ : \ Rxy$
$w_2 : Ryx \wedge Qyx$
$a_2 := \wedge\text{-el}_1(Ryx, Qyx, w_2) \ : \ Ryx$
$vxy : (Rxy \Rightarrow Ryx \Rightarrow x = y)$
$a_3 := vxya_1a_2 \ : \ x = y$
$a_4 := \lambda v : A.\lambda x, y : S.\lambda w_1 : (R \cap Q)xy.\lambda w_2 : (R \cap Q)yx.a_3$
$: \ (A \Rightarrow C)$
$v : B$
$\mathbf{var}\ x, y : S \mid w_1 : (R \cap Q)xy \mid w_2 : (R \cap Q)yx$
$w_1 : Rxy \wedge Qxy$
$a_5 := \wedge\text{-el}_2(Rxy, Qxy, w_1) \ : \ Qxy$
$w_2 : Ryx \wedge Qyx$
$a_6 := \wedge\text{-el}_2(Ryx, Qyx, w_2) \ : \ Qyx$
$vxy : (Qxy \Rightarrow Qyx \Rightarrow x = y)$
$a_7 := vxya_5a_6 \ : \ x = y$
$a_8 := \lambda v : B.\lambda x, y : S.\lambda w_1 : (R \cap Q)xy.\lambda w_2 : (R \cap Q)yx.a_7$
$: \ (B \Rightarrow C)$
$a_9 := \vee\text{-el}(A, B, C, u, a_4, a_8) : C$
$a_9 : antisym(S, R \cap Q)$

4)

$\mathbf{var}\ S : *_s \mid R, Q : br(S)$
$u_1 : trans(S, R) \mid u_2 : trans(S, Q)$
$\mathbf{var}\ x, y, z : S \mid v : (R \cap Q)xy \mid w : (R \cap Q)yz$
$v : Rxy \wedge Qxy$
$a_1 := \wedge\text{-el}_1(Rxy, Qxy, v) \ : \ Rxy$
$a_2 := \wedge\text{-el}_2(Rxy, Qxy, v) \ : \ Qxy$
$w : Ryz \wedge Qyz$
$a_3 := \wedge\text{-el}_1(Ryz, Qyz, w) \ : \ Ryz$
$a_4 := \wedge\text{-el}_2(Ryz, Qyz, w) \ : \ Qyz$
$a_5 := u_1xyza_1a_3 \ : \ Rxz$
$a_6 := u_2xyza_2a_4 \ : \ Qxz$
$a_7 := \wedge\text{-in}\ (Rxz, Qxz, a_5, a_6) \ : \ (R \cap Q)xz$
$a_8 := \lambda x, y, z : S.\lambda v : (R \cap Q)xy.\lambda w : (R \cap Q)yz.a_7$
$: \ trans(S, R \cap Q)$

□

*Theorem 5.5.* **Invariance under union.**

Suppose $R$ and $Q$ are relations on type S. Then the following hold.

1) $R$ or $Q$ is reflexive ⇒ $R \cup Q$ is reflexive.
2) $R$ and $Q$ are symmetric ⇒ $R \cup Q$ is symmetric.

*Proof.* 1)

$\mathbf{var}\ S : *_s \mid R, Q : br(S)$
$u : refl(S, R) \mid x : S$
$ux \ : \ Rxx$
$a_1 := \vee\text{-in}_1(Rxx, Qxx, ux) \ : \ (R \cup Q)xx$
$a_2 := \vee\text{-in}_2(Rxx, Qxx, ux) \ : \ (Q \cup R)xx$
$a_3 := \lambda u : refl(S, R).\lambda x : S.a_1$
$: \ (refl(S, R) \Rightarrow refl(S, R \cup Q))$
$a_4(R, Q) := \lambda u : refl(S, R).\lambda x : S.a_2$
$: \ (refl(S, R) \Rightarrow refl(S, Q \cup R))$
$a_5 := a_4(Q, R) \ : \ (refl(S, Q) \Rightarrow refl(S, R \cup Q))$
$u : refl(S, R) \vee refl(S, Q)$
$a_7 := \vee\text{-el}(refl(S, R), refl(S, Q), refl(S, R \cup Q), u, a_3, a_5)$
$: \ refl(S, R \cup Q)$

2)

$\mathbf{var}\ S : *_s \mid R, Q : br(S)$
$u_1 : sym(S, R) \mid u_2 : sym(S, Q)$
$\mathbf{var}\ x, y : S \mid v : (R \cup Q)xy$
$v : Rxy \vee Qxy$
$w : Rxy$
$a_1 := u_1xyw \ : \ Ryx$

**DOI: 10.3844/jmssp.2022.148.162**

$a_2 := \vee\text{-in}_1(Ryx, Qyx, a_1) \ : \ (R \cup Q)yx$

$a_3 := \lambda w : Rxy.a_2 \ : \ (Rxy \Rightarrow (R \cup Q)yx)$

$w : Qxy$

$a_4 := a_2xyw \ : \ Qyx$

$a_5 := \vee\text{-in}_2(Ryx, Qyx, a_4) \ : \ (R \cup Q)yx$

$a_6 := \lambda w : Qxy.a_5 \ : \ (Qxy \Rightarrow (R \cup Q)yx)$

$a_7 := \vee\text{-el}(Rxy, Qxy, (R \cup Q)yx, v, a_3, a_6) \ : \ (R \cup Q)yx$

$a_8 := \lambda x, y : S.\lambda v : (R \cup Q)xy.a_7 \ : \ sym(S, R \cup Q)$

□

*Theorem 5.6.* **Invariance under composition.**

*Suppose R and Q are relations on type S. Then the following hold.*

1) *$R \circ R^{-1}$ is always symmetric.*
2) *R and Q are reflexive ⇒ $R \circ Q$ is reflexive.*
3) *Suppose R and Q are symmetric. Then*
   *$R \circ Q$ is symmetric ⇔ $R \circ Q = Q \circ R$.*

*Proof.* 1)

$\textbf{var } S : *_s \mid R : br(S)$

$\textbf{var } x, y : S \mid u : (R \circ R^{-1})xy$

Notation $P := \lambda z : S.Rxz \wedge R^{-1}zy \ : \ S \to *_p$

$u : (\exists z : S.Pz)$

$\textbf{var } z : S \mid v : Pz$

$v : Rxz \wedge R^{-1}zy$

$a_1 := \wedge\text{-el}_1(Rxz, R^{-1}zy, v) : Rxz$

$a_2 := \wedge\text{-el}_2(Rxz, R^{-1}zy, v) : R^{-1}zy$

$a_2 : Ryz$

$a_1 : R^{-1}zx$

$a_3 := \text{prod-term}\ (S, R, R^{-1}, y, z, x, a_2, a_1) \ : \ (R \circ R^{-1})yx$

$a_4 := \lambda z : S.\lambda v : Pz.a_3 \ : \ (\forall z : S.(Pz \Rightarrow (R \circ R^{-1})yx))$

$a_5 := \exists\text{-el}\ (S, P, u, (R \circ R^{-1})yx, a_4) : (R \circ R^{-1})yx$

$a_6 := \lambda x, y : S.\lambda u : (R \circ R^{-1})xy.a_5 \ : \ sym(S, R \circ R^{-1})$

2)

$\textbf{var } S : *_s \mid R, Q : br(S)$

$u : refl(S, R) \mid v : refl(S, Q)$

$\textbf{var } x : S$

$ux \ : \ Rxx$

$vx \ : \ Qxx$

$a_1 := \text{prod-term}\ (S, R, Q, x, x, x, ux, vx) \ : \ (R \circ Q)xx$

$a_2 := \lambda x : S.a_1 \ : \ refl(S, R \circ Q)$

3) The derivation below uses the proof term *sym-criterion* (*S*, *R*) from Theorem 5.1.3) for the second criterion of symmetry and the proof term *conv-prod* from Theorem 4.3.2).

$\textbf{var } S : *_s$

$\textbf{var } R : br(S)$

$a_1 := sym\text{-}criterion(S, R) \ : \ sym(S, R) \Leftrightarrow (R^{-1} = R)$

$a_2(R) := \wedge\text{-el}_1(sym(S, R) \Rightarrow (R^{-1} = R), (R^{-1} = R)$
$\quad \Rightarrow sym(S, R), a_1) \ : \ sym(S, R) \Rightarrow (R^{-1} = R)$

$a_3(R) := \wedge\text{-el}_2(sym(S, R) \Rightarrow (R^{-1} = R), (R^{-1} = R)$
$\quad \Rightarrow sym(S, R), a_1) \ : \ (R^{-1} = R) \Rightarrow sym(S, R)$

$\textbf{var } R, Q : br(S) \mid u : sym(S, R) \mid v : sym(S, Q)$

$a_4 := a_2(R)u : (R^{-1} = R)$

$a_5 := a_2(Q)v : (Q^{-1} = Q)$

$a_6 := conv\text{-}prod(S, R, Q) \ : \ (R \circ Q)^{-1} = Q^{-1} \circ R^{-1}$

Notation $P_1 := \lambda K : br(S).((R \circ Q)^{-1} = K \circ R^{-1}) \ : \ br(S) \to *_p$

Notation $P_2 := \lambda K : br(S).((R \circ Q)^{-1} = Q \circ K) \ : \ br(S) \to *_p$

$a_6 : P_1(Q^{-1})$

$a_7 := eq\text{-}subs(br(S), P_1, Q^{-1}, Q, a_5, a_6) \ : \ (R \circ Q)^{-1} = Q \circ R^{-1}$

$a_7 : P_2(R^{-1})$

$a_8 := eq\text{-}subs(br(S), P_2, R^{-1}, R, a_4, a_7) \ : \ (R \circ Q)^{-1} = Q \circ R$

Notation $A := sym(S, R \circ Q) \ : \ *_p$

Notation $B := (R \circ Q = Q \circ R) \ : \ *_p$

$w : A$

$a_9 := a_2(R \circ Q)w \ : \ (R \circ Q)^{-1} = R \circ Q$

$a_{10} := eq\text{-}sym(br(S), (R \circ Q)^{-1}, R \circ Q, a_9) \ : \ R \circ Q = (R \circ Q)^{-1}$

$a_{11} := eq\text{-}trans(br(S), R \circ Q, (R \circ Q)^{-1}, Q \circ R, a_{10}, a_8)$
$\quad : \ R \circ Q = Q \circ R$

$a_{12} := \lambda w : A.a_{11} \ : \ A \Rightarrow B$

$w : B$

$w : (R \circ Q = Q \circ R)$

$a_{13} := eq\text{-}sym(br(S), R \circ Q, Q \circ R, w) \ : \ Q \circ R = R \circ Q$

$a_{14} := eq\text{-}trans(br(S), (R \circ Q)^{-1}, Q \circ R, R \circ Q, a_8, a_{13})$
$\quad : \ (R \circ Q)^{-1} = R \circ Q$

$a_{15} := a_3(R \circ Q)a_{14} \ : \ sym(S, R \circ Q)$

$a_{16} := \lambda w : B.a_{15} \ : \ B \Rightarrow A$

$a_{17} := bi\text{-}impl(A, B, a_{12}, a_{16}) : (sym(S, R \circ Q) \Leftrightarrow R \circ Q = Q \circ R)$

□

## 6. Special Binary Relations

### 6.1. Equivalence Relation and Partition

*Theorem 6.1.* **Invariance of equivalence relation under converse operation and intersection.**

Suppose *R* and *Q* are equivalence relations on type *S*. Then the following hold.

1) $R^{-1}$ is an equivalence relation on *S*.
2) $R \cap Q$ is an equivalence relation on *S*.

*Proof*

1) Can easily be derived from Theorem 5.3.1), 2), 4) using intuitionistic logic.
2) Can easily be derived from Theorem 5.4. 1), 2), 4) using intuitionistic logic.

The formal proofs are skipped. □

Next is a formalization of the fact that there is a correspondence between equivalence relations on *S* and partitions of *S*. Equivalence classes are introduced in Nederpelt and Geuvers (2014) as follows.

$\mathbf{var}\ S : *_s \mid R : br(S) \mid u : equiv\text{-}rel(S, R)$

$\quad \mathbf{var}\ x : S$

$\quad\quad \boldsymbol{class}(S, R, u, x) := \{y : S \mid Rxy\} : ps(S)$

$\quad\quad$ Notation $[\boldsymbol{x}]_{\boldsymbol{R}}$ for $class(S, R, u, x)$

Next, a partition of type *S* is defined:

$\mathbf{var}\ S : *_s \mid R : S \to ps(S)$

$\quad \boldsymbol{partition}(S, R) := (\forall x : S.x\varepsilon Rx)$

$\quad\quad \wedge \forall x, y, z : S.(z\varepsilon Rx \Rightarrow z\varepsilon Ry \Rightarrow Rx = Ry))$

As usual, one can regard a partition *R* as a collection *Rx* ($x \in S$) of subsets of *S*. From this point of view, the above diagram expresses the standard two facts for a partition:

(1) any element of *S* belongs to one of the subsets from the collection (namely *Rx*);
(2) if intersection of two subsets *Rx* and *Ry* is non-empty, then they coincide.

(1) implies that each subset from the collection is non-empty and that the union of all subsets from the collection is *S*.

*Theorem 6.2.*

Any equivalence relation R on type S is a partition of S and vice versa.

*Proof*. The type of partitions of S is S → ps(S), which is $S \to S \to *_p$, and it is the same as the type br(S) of relations on S. The proof consists of two steps.

*Step 1. Any equivalence relation is a partition.*

$\mathbf{var}\ S : *_s \mid R : S \to S \to *_p$

$\quad u : equiv\text{-}rel(S, R)$

$\quad\quad a_1 := \wedge\text{-}el_1(refl(S, R), sym(S, R), \wedge\text{-}el_1(refl(S, R) \wedge sym(S, R),$

$\quad\quad\quad trans(S, R), u)) : refl(S, R)$

$\quad\quad \mathbf{var}\ x : S$

$\quad\quad\quad a_2 := a_1 x : Rxx$

$\quad\quad\quad a_2 : (x\varepsilon Rx)$

$\quad\quad a_3 := \lambda x : S.a_2 \ :\ (\forall x : S.x\varepsilon X)$

This proves the first part of the definition of *partition* (*S*, R), and the second part was proven in Nederpelt and Geuvers (2014), pg. 291.

*Step 2. Any partition is an equivalence relation.*

$\mathbf{var}\ S : *_s \mid R : S \to S \to *_p$

$\quad u : partition(S, R)$

$\quad\quad$ Notation $A := \forall x : S.(x\varepsilon Rx)$

$\quad\quad$ Notation $B := \forall x, y, z : S.(z\varepsilon Rx \Rightarrow z\varepsilon Ry \Rightarrow Rx = Ry)$

$\quad\quad u : A \wedge B$

$\quad\quad a_1 := \wedge\text{-}el_1(A, B, u) : A$

$\quad\quad a_2 := \wedge\text{-}el_2(A, B, u) : B$

$\quad\quad \mathbf{var}\ x : S$

$\quad\quad\quad a_3 := a_1 x : x\varepsilon Rx$

$\quad\quad\quad a_3 : Rxx$

$\quad\quad a_4 := \lambda x : S.a_3 \ :\ refl(S, R)$

$\quad\quad \mathbf{var}\ x, y : S \mid v : Rxy$

$\quad\quad\quad a_5 := a_1 y \ :\ (y\varepsilon Ry)$

$\quad\quad\quad v : (y\varepsilon Rx)$

$\quad\quad\quad a_6 := a_2 xyyva_5 \ :\ Rx = Ry$

$\quad\quad\quad a_7 := a_1 x \ :\ (x\varepsilon Rx)$

$\quad\quad\quad a_8 := eq\text{-}subs(ps(S), \lambda Z : ps(S).x\varepsilon Z, Rx, Ry, a_6, a_7) \ :\ (x\varepsilon Ry)$

$\quad\quad\quad a_8 : Ryx$

$\quad\quad a_9 := \lambda x, y : S.\lambda v : Rxy.a_8 \ :\ sym(S, R)$

$\quad\quad \mathbf{var}\ x, y, z : S \mid v : Rxy \mid w : Ryz$

$\quad\quad\quad v : y\varepsilon Rx$

$\quad\quad\quad a_{10} := a_9 yzw \ :\ Rzy$

$\quad\quad\quad a_{10} : (y\varepsilon Rz)$

$\quad\quad\quad a_{11} := a_2 zxya_{10} v \ :\ Rz = Rx$

$\quad\quad\quad a_{12} := a_1 z \ :\ (z\varepsilon Rz)$

$\quad\quad\quad a_{13} := eq\text{-}subs(ps(S), \lambda Z : ps(S).z\varepsilon Z, Rz, Rx, a_{11}, a_{12}) \ :\ z\varepsilon Rx$

$\quad\quad\quad a_{13} : Rxz$

$\quad\quad a_{14} := \lambda x, y, z : S.\lambda v : Rxy.\lambda w : Ryz.a_{13} \ :\ trans(S, R)$

$\quad\quad a_{15} := \wedge\text{-}in(refl(S, R) \wedge sym(S, R), trans(S, R), \wedge\text{-}in(refl(S, R),$

$\quad\quad\quad sym(S, R), a_4, a_9), a_{14}) \ :\ equiv\text{-}rel(S, R)$

□

## 6.2. Partial Order

*Theorem 6.3.* **Invariance of partial order under converse operation and intersection.**

Suppose *R* and *Q* are partial orders on type *S*. Then the following hold.

1) $R^{-1}$ is a partial order on *S*.
2) $R \cap Q$ is a partial order on *S*.

*Proof.*
1) can easily be derived from Theorem 5.3.1), 3), 4) using intuitionistic logic.
2) can easily be derived from Theorem 5.4. 1), 3), 4) using intuitionistic logic.

The formal proofs are skipped. □

*Example 6.4*

⊆ is a partial order on the power set *ps*(*S*) of type *S*.

**DOI: 10.3844/jmssp.2022.148.162**

*Proof.*

This is the formal proof.

$\mathbf{var}\ S : *_s$

$\text{Notation } R := \lambda X, Y : ps(S).X \subseteq Y \ : \ br(ps(S))$

$\text{Notation } A := refl(ps(S), R)$

$\text{Notation } B := antisym(ps(S), R)$

$\text{Notation } C := trans(ps(S), R)$

$\mathbf{var}\ X : ps(S)$

$a_1 := \lambda x : S.\lambda u : (x\varepsilon X).u \ : \ X \subseteq X$

$a_2 := \lambda X : ps(S).a_1 \ : \ A$

$\mathbf{var}\ X, Y : ps(S) \mid u : X \subseteq Y \mid v : Y \subseteq X$

$a_3 := \wedge\text{-}in(X \subseteq Y, Y \subseteq X, u, v) \ : \ X = Y$

$a_4 := \lambda X, Y : ps(S).\lambda u : X \subseteq Y.\lambda v : Y \subseteq X.a_3 \ : \ B$

$\mathbf{var}\ X, Y, Z : ps(S) \mid u : X \subseteq Y \mid v : Y \subseteq Z$

$\mathbf{var}\ x : S \mid w : x\varepsilon X$

$a_5 := uxw \ : \ (x\varepsilon Y)$

$a_6 := vxa_5 \ : \ (x\varepsilon Z)$

$a_7 := \lambda x : S.\lambda w : (x\varepsilon X).a_6 \ : \ X \subseteq Z$

$a_8 := \lambda X, Y, Z : ps(S).\lambda u : X \subseteq Y.\lambda v : Y \subseteq Z.a_7 \ : \ C$

$a_9 := \wedge\text{-}in(A \wedge B, C, \wedge\text{-}in(A, B, a_2, a_4), a_8) \ : \ A \wedge B \wedge C$

$a_9 : part\text{-}ord(ps(S), R)$

□

### *6.3. Well-Ordering and Transfinite Induction*

Notation ≤ will be used for a partial order. The following diagram defines the strict order <, the least element of a partially ordered set, and the well-ordering of type *S*.

$\mathbf{var}\ S : *_s \mid \ \leqslant : br(S) \mid u : part\text{-}ord(S, \leqslant)$

$\text{Definition } < := \lambda x, y : S.(x \leqslant y \wedge \neg(x = y))$

$\mathbf{var}\ X : ps(S) \mid x : S$

$\text{Definition } \boldsymbol{least}(S, \leqslant, X, x) := x\varepsilon X \wedge \forall y : S.(y\varepsilon X \Rightarrow x \leqslant y)$

$\text{Definition } \boldsymbol{well\text{-}ord}(S, \leqslant) := part\text{-}ord(S, \leqslant)$
$\quad \wedge \forall X : ps(S).[\exists x : S.x\varepsilon X \Rightarrow \exists x : S.least(S, \leqslant, X, x)]$

*Theorem 6.5.* **Transfinite Induction.**

Suppose ≤ is a well-ordering of type *S*. Then for any predicate *P* on *S*:

$\forall x : S.[(\forall y : S.(y < x \Rightarrow Py) \Rightarrow Px] \Rightarrow \forall x : S.Px.$

*Proof*

Here is the formal proof.

$\mathbf{var}\ S : *_s \mid \ \leqslant : br(S) \mid u_1 : well\text{-}ord(S, \leqslant) \mid P : S \to *_p$

$u_2 \ : \ \forall x : S.[\forall y : S.(y < x \Rightarrow Py) \Rightarrow Px]$

$\text{Notation } A := part\text{-}ord(S, \leqslant)$

$\text{Notation } B := [\forall X : ps(S).(\exists x : S.x\varepsilon X$
$\quad \Rightarrow \exists x : S.least(S, \leqslant, X, x))]$

$u_1 : A \wedge B$

$a_1 := \wedge\text{-}el_1(A, B, u_1) \ : A$

$a_2 := \wedge\text{-}el_2(A, B, u_1) \ : B$

$a_3 := \wedge\text{-}el_1(refl(S, \leqslant) \wedge antisym(S, \leqslant), trans(S, \leqslant), a_1)$
$\quad : \ refl(S, \leqslant) \wedge antisym(S, \leqslant)$

$a_4 := \wedge\text{-}el_2(refl(S, \leqslant), antisym(S, \leqslant), a_3) \ : \ antisym(S, \leqslant)$

$\text{Notation } X := \lambda x : S.\neg Px \ : \ ps(S)$

$v_1 : (\exists x : S.x\varepsilon X)$

$a_5 := a_2 X v_1 \ : \ [\exists x : S.least(S, \leqslant, X, x)]$

$\mathbf{var}\ x : S \mid v_2 : least(S, \leqslant, X, x)$

$a_6 := \wedge\text{-}el_1(x\varepsilon X, \forall y : S.(y\varepsilon X \Rightarrow x \leqslant y), v_2) \ : \ x\varepsilon X$

$a_6 : \neg Px$

$a_7 := \wedge\text{-}el_2(x\varepsilon X, \forall y : S.(y\varepsilon X \Rightarrow x \leqslant y), v_2)$
$\quad : \ [\forall y : S.(y\varepsilon X \Rightarrow x \leqslant y)]$

$\mathbf{var}\ y : S \mid w_1 : y < x$

$a_8 := \wedge\text{-}el_1(y \leqslant x, \neg(x = y), w_1) \ : \ y \leqslant x$

$a_9 := \wedge\text{-}el_2(y \leqslant x, \neg(x = y), w_1) \ : \ \neg(x = y)$

$w_2 : \neg Py$

$w_2 : y\varepsilon X$

$a_{10} := a_7 y w_2 : x \leqslant y$

$a_{11} := a_4 x y a_{10} a_8 : x = y$

$a_{12} := a_9 a_{11} \ : \ \bot$

$a_{13} := \lambda w_2 : \neg Py.a_{12} \ : \ \neg\neg Py$

$a_{14} := doub\text{-}neg(Py)a_{13} \ : \ Py$

$a_{15} := \lambda y : S.\lambda w_1 : y < x.a_{14} \ : \ [\forall y : S.(y < x \Rightarrow Py)]$

$a_{16} := u_2 x a_{15} \ : \ Px$

$a_{17} := a_6 a_{16} \ : \ \bot$

$a_{18} := \lambda x : S.\lambda v_2 : least(S, \leqslant, X, x).a_{17}$
$\quad : \ [\forall x : S.(least(S, \leqslant, X, x) \Rightarrow \bot)]$

$a_{19} := \exists\text{-}el(S, \lambda x : S.least(S, \leqslant, X, x), a_5, \bot, a_{18}) \ : \ \bot$

$a_{20} := \lambda v_1 : (\exists x : S.x\varepsilon X).a_{19} \ : \ \neg(\exists x : S.x\varepsilon X)$

$\mathbf{var}\ x : S$

$w : \neg Px$

$w : x\varepsilon X$

$a_{21} := \exists\text{-}in(S, \lambda z : S.z\varepsilon X, x, w) \ : \ (\exists z : S.z\varepsilon X)$

$a_{22} := a_{20} a_{21} \ : \ \bot$

$a_{23} := \lambda w : \neg Px.a_{22} \ : \ \neg\neg Px$

$a_{24} := doub\text{-}neg(Px)a_{23} \ : \ Px$

$a_{25} := \lambda x : S.a_{24} \ : \ (\forall x : S.Px)$

Here the Double Negation theorem is used (twice) with the proof term *doub-neg*. This is the only place in this study where classical (not intuitionistic) logic is used.

**DOI: 10.3844/jmssp.2022.148.162**

## 7. Conclusion

Starting with the definitions from Nederpelt and Geuvers (2014) of binary relations and properties of reflexivity, symmetry, antisymmetry, and transitivity, this study formalizes in the theory λ*D* (the Calculus of Constructions with Definitions) criteria for these properties and proves their invariance under operations of union, intersection, composition, and taking converse. The author provides a formal definition of partition and formally proves correspondence between equivalence relations and partitions. The author derives formal proof that ⊆ is a partial order on the power set. Finally, the author formally proves the principle of transfinite induction for a type with well-ordering.

The results can be transferred to the proof assistants that are based on the Calculus of Constructions. Since binary relation is an abstract concept used in many areas of mathematics, the results can be useful for further formalizations of mathematics in λ*D*. Next direction of research is formalization of parts of probability theory in λ*D* that was outlined in Kachapova (2018).

## Acknowledgment

The author thanks the Editor in Chief and Reviewer.

## Ethics

This is a mathematical article; no ethical issues can arise after its publication.

## References

Barendregt, H. (2012). The Lambda Calculus, its Syntax and Semantics. vol 40 (Studies in Logic, Mathematical Logic and Foundations). https://www.collegepublications.co.uk/logic/mlf/?00021

Bertot, Y., & Castéran, P. (2013). *Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions*. Springer Science & Business Media. https://link.springer.com/book/10.1007/978-3-662-07964-5

Coq Development Team. (2021). *The coq proof assistant, reference manual*, 2021. Coq Development Team. https://coq.inria.fr/distrib/current/refman/

Granstrom, J. G. (2011). *Treatise on intuitionistic type theory* (Vol. 22). Springer Science & Business Media. https://link.springer.com/book/10.1007/978-94-007-1736-7

Kachapova, F. (2018). Formalizing Probability Concepts in a Type Theory. *Journal of Mathematics and Statistics*. https://thescipub.com/abstract/10.3844/jmssp.2018.209.218

Nederpelt, R. P., & Kamareddine, F. D. (2004). Logical reasoning: A first course. ISBN: 9780954300678.

Nederpelt, R., & Geuvers, H. (2014). *Type theory and formal proof*. Cambridge University Press. ISBN: 978-1-107-03650-5.

# APPENDIX

## Appendix A. Proof of Theorem 4.3

*Proof.* 1)

**var** $S : *_s \mid R : br(S)$

  **var** $x, y : S$

    $u : (R^{-1})^{-1}xy$

      $u : R^{-1}yx$

      $u : Rxy$

    $a_1 := \lambda u : (R^{-1})^{-1}xy.u \ : \ (R^{-1})^{-1}xy \Rightarrow Rxy$

    $u : Rxy$

      $u : R^{-1}yx$

      $u : (R^{-1})^{-1}xy$

    $a_2 := \lambda u : Rxy.u \ : \ Rxy \Rightarrow (R^{-1})^{-1}xy$

  $a_3 := \lambda x, y : S.a_1 \ : \ (R^{-1})^{-1} \subseteq R$

  $a_4 := \lambda x, y : S.a_2 \ : \ R \subseteq (R^{-1})^{-1}$

  $a_5 := \textit{rel-equal}((R^{-1})^{-1}, R, a_3, a_4) \ : \ (R^{-1})^{-1} = R$

2)

**var** $S : *_s \mid R, Q : br(S)$

  Notation $A := (R \circ Q)^{-1} \ : br(S)$

  Notation $B := Q^{-1} \circ R^{-1} \ : br(S)$

  **var** $x, y : S$

    Notation $P_1 := \lambda z : S.Ryz \wedge Qzx \ : \ S \to *_p$

    Notation $P_2 := \lambda z : S.Q^{-1}xz \wedge R^{-1}zy \ : \ S \to *_p$

    $u : Axy$

      $u : (R \circ Q)yx$

      $u : (\exists z : S.P_1 z)$

      $z : S \mid v : P_1 z$

        $v : Ryz \wedge Qzx$

        $a_1 := \wedge\text{-el}_1(Ryz, Qzx, v) : Ryz$

        $a_2 := \wedge\text{-el}_2(Ryz, Qzx, v) : Qzx$

        $a_1 : R^{-1}zy$

        $a_2 : Q^{-1}xz$

        $a_3 := \text{prod-term}\,(S, Q^{-1}, R^{-1}, x, z, y, a_2, a_1)$
          $: \ (Q^{-1} \circ R^{-1})xy$

        $a_3 : Bxy$

      $a_4 := \lambda z : S.\lambda v : P_1 z.a_3 \ : \ (\forall z : S.(P_1 z \Rightarrow Bxy))$

      $a_5 := \exists\text{-el}\,(S, P_1, u, Bxy, a_4) : Bxy$

  $a_6 := \lambda x, y : S.\lambda u : Axy.a_5 \ : \ A \subseteq B$

**DOI: 10.3844/jmssp.2022.148.162**

$a_5 := \exists\text{-el}\,(S, P_1, u, Bxy, a_4) : Bxy$

$a_6 := \lambda x, y : S.\lambda u : Axy.a_5 \ : \ A \subseteq B$

$\textbf{var}\ x, y : S \mid u : Bxy$

$u : (\exists z : S.P_2 z)$

$z : S \mid v : P_2 z$

$v : Q^{-1}xz \wedge R^{-1}zy$

$a_7 := \wedge\text{-el}_1(Q^{-1}xz, R^{-1}zy, v) : Q^{-1}xz$

$a_8 := \wedge\text{-el}_2(Q^{-1}xz, R^{-1}zy, v) : R^{-1}zy$

$a_7 : Qzx$

$a_8 : Ryz$

$a_9 := \text{prod-term}\,(S, R, Q, y, z, x, a_8, a_7) \ : \ (R \circ Q)yx$

$a_9 : (R \circ Q)^{-1}xy$

$a_9 : Axy$

$a_{10} := \lambda z : S.\lambda v : P_2 z.a_9 \ : \ (\forall z : S.(P_2 z \Rightarrow Axy))$

$a_{11} := \exists\text{-el}\,(S, P_2, u, Axy, a_{10}) : Axy$

$a_{12} := \lambda x, y : S.\lambda u : Bxy.a_{11} \ : \ B \subseteq A$

$\textbf{\textit{conv-prod}}(S, R, Q) := \textit{rel-equal}(A, B, a_6, a_{12})$

$: \ (R \circ Q)^{-1} = Q^{-1} \circ R^{-1}$

3)

$\textbf{var}\ S : *_s \mid R, Q : br(S)$

Notation $A := (R \cap Q)^{-1} \ : br(S)$

Notation $B := R^{-1} \cap Q^{-1} \ : br(S)$

$\textbf{var}\ x, y : S \mid u : Axy$

$u : (R \cap Q)^{-1}xy$

$u : (R \cap Q)yx$

$u : Ryx \wedge Qyx$

$a_1 := \wedge\text{-el}_1(Ryx, Qyx, v) : Ryx$

$a_2 := \wedge\text{-el}_2(Ryx, Qyx, v) : Qyx$

$a_1 : R^{-1}xy$

$a_2 : Q^{-1}xy$

$a_3 := \wedge\text{-in}(R^{-1}xy, Q^{-1}xy, a_1, a_2) : Bxy$

$a_4 := \lambda x, y : S.\lambda u : Axy.a_3 \ : \ A \subseteq B$

$\textbf{var}\ x, y : S \mid u : Bxy$

$u : R^{-1}xy \wedge Q^{-1}xy$

$a_5 := \wedge\text{-el}_1(R^{-1}xy, Q^{-1}xy, v) : R^{-1}xy$

$a_6 := \wedge\text{-el}_2(R^{-1}xy, Q^{-1}xy, v) : Q^{-1}xy$

$a_5 : Ryx$

$a_6 : Qyx$

$a_7 := \wedge\text{-in}(Ryx, Qyx, a_5, a_6) : (R \cap Q)yx$

$a_7 : (R \cap Q)^{-1}xy$

$a_7 : Axy$

$a_8 := \lambda x, y : S.\lambda u : Bxy.a_7 \ : \ B \subseteq A$

$a_9 := \textit{rel-equal}(A, B, a_4, a_8) : (R \cap Q)^{-1} = R^{-1} \cap Q^{-1}$

4)

$\textbf{var}\ S : *_s \mid R, Q : br(S)$

Notation $A := (R \cup Q)^{-1} \ : br(S)$

Notation $B := R^{-1} \cup Q^{-1} \ : br(S)$

$\textbf{var}\ x, y : S \mid u : Axy$

$u : (R \cup Q)yx$

$u : Ryx \vee Qyx$

$v : Ryx$

$v : R^{-1}xy$

$a_1 := \vee\text{-in}_1(R^{-1}xy, Q^{-1}xy, v) : Bxy$

$a_2 := \lambda v : Ryx.a_1 \ : \ Ryx \Rightarrow Bxy$

$v : Qyx$

$v : Q^{-1}xy$

$a_3 := \vee\text{-in}_2(R^{-1}xy, Q^{-1}xy, v) : Bxy$

$a_4 := \lambda v : Qyx.a_3 \ : \ Qyx \Rightarrow Bxy$

$a_5 := \vee\text{-el}\,(Ryx, Qyx, Bxy, u, a_2, a_4) \ : \ Bxy$

$a_6 := \lambda x, y : S.\lambda u : Axy.a_5 \ : \ A \subseteq B$

$\textbf{var}\ x, y : S \mid u : Bxy$

$u : R^{-1}xy \vee Q^{-1}xy$

$v : R^{-1}xy$

$v : Ryx$

$a_7 := \vee\text{-in}_1(Ryx, Qyx, v) : Ryx \vee Qyx$

$a_7 : (R \cup Q)^{-1}xy$

$a_7 : Axy$

$a_8 := \lambda v : R^{-1}xy.a_7 \ : \ R^{-1}xy \Rightarrow Axy$

$v : Q^{-1}xy$

$v : Qyx$

$a_9 := \vee\text{-in}_2(Ryx, Qyx, v) : Ryx \vee Qyx$

$a_9 : (R \cup Q)^{-1}xy$

$a_9 : Axy$

$a_{10} := \lambda v : Q^{-1}xy.a_9 \ : \ Q^{-1}xy \Rightarrow Axy$

$a_{11} := \vee\text{-el}\,(R^{-1}xy, Q^{-1}xy, Axy, u, a_8, a_{10}) \ : \ Axy$

$a_{12} := \lambda x, y : S.\lambda u : Bxy.a_{11} \ : \ B \subseteq A$

$a_{13} := \textit{rel-equal}(A, B, a_6, a_{12}) \ : \ (R \cup Q)^{-1} = R^{-1} \cup Q^{-1}$

5)

$\textbf{var}\ S : *_s \mid R, P, Q : br(S)$

Notation $A := R \circ (P \cup Q) \ : br(S)$

Notation $B := R \circ P \cup R \circ Q \ : br(S)$

$\textbf{var}\ x, y : S$

Notation $P_0 := \lambda z : S.Rxz \wedge (P \cup Q)zy \ : \ S \to *_p$

$u : Axy$

$u : (\exists z : S.P_0 z)$

$z : S \mid v : P_0 z$

$v : Rxz \wedge (P \cup Q)zy$

$a_1 := \wedge\text{-}el_1(Rxz, (P \cup Q)zy, v) : Rxz$

$a_2 := \wedge\text{-}el_2(Rxz, (P \cup Q)zy, v) : (P \cup Q)zy$

$a_2 : Pzy \vee Qzy$

$w : Pzy$

$a_3 := \text{prod-term}\ (S, R, P, x, z, y, a_1, w)\ :\ (R \circ P)xy$

$a_4 := \vee\text{-}in_1((R \circ P)xy, (R \circ Q)xy, a_3) : Bxy$

$a_5 := \lambda w : Pzy.a_4\ :\ Pzy \Rightarrow Bxy$

$w : Qzy$

$a_6 := \text{prod-term}\ (S, R, Q, x, z, y, a_1, w)\ :\ (R \circ Q)xy$

$a_7 := \vee\text{-in}\ _2((R \circ P)xy, (R \circ Q)xy, a_6) : Bxy$

$a_8 := \lambda w : Qzy.a_7\ :\ Qzy \Rightarrow Bxy$

$a_9 := \vee\text{-el}\ (Pzy, Qzy, Bxy, a_2, a_5, a_8) : Bxy$

$a_{10} := \lambda z : S.\lambda v : P_0 z.a_9\ :\ (\forall z : S.(P_0 z \Rightarrow Bxy))$

$a_{11} := \exists\text{-el}\ (S, P_0, u, Bxy, a_{10}) : Bxy$

$a_{12} := \lambda x, y : S.\lambda u : Axy.a_{11}\ :\ A \subseteq B$

$\mathbf{var}\ x, y : S$

Notation $P_1 := \lambda z : S.Rxz \wedge Pzy\ :\ S \to *_p$

Notation $P_2 := \lambda z : S.Rxz \wedge Qzy\ :\ S \to *_p$

$u : Bxy$

$u : (R \circ P)xy \vee (R \circ Q)xy$

$v : (R \circ P)xy$

$v : (\exists z : S.P_1 z)$

$z : S \mid w : P_1 z$

$w : Rxz \wedge Pzy$

$a_{13} := \wedge\text{-el}_1(Rxz, Pzy, w)\ :\ Rxz$

$a_{14} := \wedge\text{-el}_2(Rxz, Pzy, w)\ :\ Pzy$

$a_{15} := \vee\text{-}in_1(Pzy, Qzy, a_{14}) : (P \cup Q)zy$

$a_{16} := \text{prod-term}\ (S, R, (P \cup Q), x, z, y, a_{13}, a_{15})$

$:\ Axy$

$a_{17} := \lambda z : S.\lambda w : P_1 z.a_{16}\ :\ (\forall z : S.(P_1 z \Rightarrow Axy))$

$a_{18} := \exists\text{-el}\ (S, P_1, v, Axy, a_{17}) : Axy$

$a_{19} := \lambda v : (R \circ P)xy.a_{18}\ :\ ((R \circ P)xy \Rightarrow Axy)$

$v : (R \circ Q)xy$

$v : (\exists z : S.P_2 z)$

$z : S \mid w : P_2 z$

$a_{20} := \wedge\text{-el}_1(Rxz, Qzy, w)\ :\ Rxz$

$a_{21} := \wedge\text{-el}_2(Rxz, Qzy, w)\ :\ Qzy$

$a_{22} := \vee\text{-}in_2(Pzy, Qzy, a_{21}) : (P \cup Q)zy$

$a_{23} := \text{prod-term}\ (S, R, (P \cup Q), x, z, y, a_{20}, a_{22})$

$:\ Axy$

$a_{24} := \lambda z : S.\lambda w : P_2 z.a_{23}\ :\ (\forall z : S.(P_2 z \Rightarrow Axy))$

$a_{25} := \exists\text{-el}\ (S, P_2, v, Axy, a_{24}) : Axy$

$a_{26} := \lambda v : (R \circ Q)xy.a_{25}\ :\ ((R \circ Q)xy \Rightarrow Axy)$

$a_{27} := \vee\text{-el}((R \circ P)xy, (R \circ Q)xy, Axy, u, a_{19}, a_{26})\ :\ Axy$

$a_{28} := \lambda x, y : S.\lambda u : Bxy.a_{27}\ :\ B \subseteq A$

$a_{29} := rel\text{-}equal(A, B, a_{12}, a_{28})\ :\ R \circ (P \cup Q) = R \circ P \cup R \circ Q$

6) is proven similarly to 5).

7)

$\mathbf{var}\ S : *_s \mid R, P, Q : br(S)$

Notation $A := R \circ (P \cap Q)\ :\ br(S)$

Notation $B := R \circ P \cap R \circ Q\ :\ br(S)$

$\mathbf{var}\ x, y : S$

Notation $P := \lambda z : S.Rxz \wedge (P \cap Q)zy\ :\ *_p$

$u : Axy$

$u : (\exists z : S.Pz)$

$\mathbf{var}\ z : S \mid v : Pz$

$v : Rxz \wedge (P \cap Q)zy$

$a_1 := \wedge\text{-el}_1(Rxz, (P \cap Q)zy, v) : Rxz$

$a_2 := \wedge\text{-el}_2(Rxz, (P \cap Q)zy, v) : (P \cap Q)zy$

$a_2 : Pzy \wedge Qzy$

$a_3 := \wedge\text{-el}_1(Pzy, Qzy, a_2) : Pzy$

$a_4 := \wedge\text{-el}_2(Pzy, Qzy, a_2) : Qzy$

$a_5 := \text{prod-term}\ (S, R, P, x, z, y, a_1, a_3)\ :\ (R \circ P)xy$

$a_6 := \text{prod-term}\ (S, R, Q, x, z, y, a_1, a_4)\ :\ (R \circ Q)xy$

$a_7 := \wedge\text{-in}\ ((R \circ P)xy, (R \circ Q)xy, a_5, a_6)\ :\ Bxy$

$a_8 := \lambda z : S.\lambda v : Pz.a_7\ :\ (\forall z : S.(Pz \Rightarrow Bxy))$

$a_9 := \exists\text{-el}\ (S, P, u, Bxy, a_8) : Bxy$

$a_{10} := \lambda x, y : S.\lambda u : Axy.a_9\ :\ R \circ (P \cap Q) \subseteq R \circ P \cap R \circ Q$

8) is proven similarly to 7).

9)

$\mathbf{var}\ S : *_s \mid R, P, Q : br(S)$

Notation $A := (R \circ P) \circ Q\ :\ br(S)$

Notation $B := R \circ (P \circ Q)\ :\ br(S)$

$\mathbf{var}\ x, y : S$

Notation $P_1(x, y) := \lambda z : S.(R \circ P)xz \wedge Qzy\ :\ S \to *_p$

Notation $P_2(x, y) := \lambda z : S.Rxz \wedge (P \circ Q)zy\ :\ S \to *_p$

Notation $P_3(x, y) := \lambda z : S.Rxz \wedge Pzy\ :\ S \to *_p$

Notation $P_4(x, y) := \lambda z : S.Pxz \wedge Qzy\ :\ S \to *_p$

$\mathbf{var}\ x, y : S \mid u : Axy$

$u : (\exists z : S.P_1(x, y)z)$

$\mathbf{var}\ z : S \mid v : P_1(x, y)z$

$a_1 := \wedge\text{-el}_1((R \circ P)xz, Qzy, v) : (R \circ P)xz$

$a_2 := \wedge\text{-el}_2((R \circ P)xz, Qzy, v) : Qzy$

$a_1 : (\exists z_1 : S.P_3(x, z)z_1)$

$\mathbf{var}\ z_1 : S \mid w : P_3(x, z)z_1$

$w : Rxz_1 \wedge Pz_1 z$

$a_3 := \wedge\text{-el}_1(Rxz_1, Pz_1 z, w) : Rxz_1$

$a_4 := \wedge\text{-el}_2(Rxz_1, Pz_1 z, w) : Pz_1 z$

**DOI: 10.3844/jmssp.2022.148.162**

$a_5 := \text{prod-term}\,(S, P, Q, z_1, z, y, a_4, a_2) \ : \ (P \circ Q)z_1y$

$a_6 := \text{prod-term}\,(S, R, (P \circ Q), x, z_1, y, a_3, a_5) \ : \ Bxy$

$a_7 := \lambda z_1 : S.\lambda w : P_3(x, z)z_1.a_6$

$(\forall z_1 : S.(P_3(x, z)z_1 \Rightarrow Bxy))$

$a_8 := \exists\text{-el}\,(S, P_3(x, z), a_1, Bxy, a_7) : Bxy$

$a_9 := \lambda z : S.\lambda v : P_1(x, y)z.a_8 \ : \ (\forall z : S.(P_1(x, y)z \Rightarrow Bxy))$

$a_{10} := \exists\text{-el}\,(S, P_1(x, y), u, Bxy, a_9) : Bxy$

$a_{11} := \lambda x, y : S.\lambda u : Axy.a_{10} \ : \ A \subseteq B$

$\boxed{\textbf{var}\ x, y : S \mid u : Bxy}$

$u : (\exists z : S.P_2(x, y)z)$

$\boxed{\textbf{var}\ z : S \mid v : P_2(x, y)z}$

$a_{12} := \wedge\text{-el}_1(Rxz, (P \circ Q)zy, v) : Rxz$

$a_{13} := \wedge\text{-el}_2(Rxz, (P \circ Q)zy, v) : (P \circ Q)zy$

$a_{13} : (\exists z_1 : S.P_4(z, y)z_1)$

$\boxed{\textbf{var}\ z_1 : S \mid w : P_4(z, y)z_1}$

$w : Pzz_1 \wedge Qz_1y$

$a_{14} := \wedge\text{-el}_1(Pzz_1, Qz_1y, w) : Pzz_1$

$a_{15} := \wedge\text{-el}_2(Pzz_1, Qz_1y, w) : Qz_1y$

$a_{16} := \text{prod-term}\,(S, R, P, x, z, z_1, a_{12}, a_{14})$

$: \ (R \circ P)xz_1$

$a_{17} := \text{prod-term}\,(S, R \circ P, Q, x, z_1, y, a_{16}, a_{15}) \ : \ Axy$

$a_{18} := \lambda z_1 : S.\lambda w : P_4(z, y)z_1.a_{17}$

$: \ (\forall z_1 : S.(P_4(z, y)z_1 \Rightarrow Axy))$

$a_{19} := \exists\text{-el}\,(S, P_4(z, y), a_{13}, Axy, a_{18}) : Axy$

$a_{20} := \lambda z : S.\lambda v : P_2(x, y)z.a_{19} \ : \ (\forall z : S.(P_2(x, y)z \Rightarrow Axy))$

$a_{21} := \exists\text{-el}\,(S, P_2(x, y), u, Axy, a_{20}) : Axy$

$a_{22} := \lambda x, y : S.\lambda u : Bxy.a_{21} \ : \ B \subseteq A$

$a_{23} := \textit{rel-equal}(A, B, a_{11}, a_{22}) \ : \ (R \circ P) \circ Q = R \circ (P \circ Q)$

□

## Appendix B. Proof of Theorem 5.1

*Proof*. Each statement here is a bi-implication, so the proof term *bi-impl* from Lemma 2.3 is used.

1)

$\boxed{\textbf{var}\ S : *_s \mid R : br(S)}$

Notation $A := refl(S, R) : *_p$

Notation $B := id_s \subseteq R : *_p$

$\boxed{u : A}$

$\boxed{\textbf{var}\ x, y : S \mid v : (id_S)xy}$

$v : x =_S y$

Notation $P := \lambda z : S.Rxz \ : \ S \to *_p$

$ux : Px$

$a_1 := \textit{eq-subs}(S, P, x, y, v, ux) \ : \ Py$

$a_1 : Rxy$

$a_2 := \lambda x, y : S.\lambda v : (id_S)xy.a_1 \ : \ (id_S \subseteq R)$

$a_2 : B$

$a_3 := \lambda u : A.a_2 \ : \ (A \Rightarrow B)$

$\boxed{u : B}$

$\boxed{\textbf{var}\ x : S}$

$a_4 := \textit{eq-refl}(S, x) \ : \ x =_S x$

$a_4 : (id_S)xx$

$uxx : (id_S)xx \Rightarrow Rxx$

$a_5 := uxxa_4 \ : \ Rxx$

$a_6 := \lambda x : S.a_5 \ : \ (\forall x : S.Rxx)$

$a_6 : A$

$a_7 := \lambda u : B.a_6 \ : \ (B \Rightarrow A)$

$a_8 := \textit{bi-impl}(A, B, a_3, a_7) \ : \ refl(S, R) \Leftrightarrow id_s \subseteq R$

2) and 3) are proven together as follows.

$\boxed{\textbf{var}\ S : *_s \mid R : br(S)}$

Notation $A := sym(S, R) : *_p$

Notation $B := R^{-1} \subseteq R : *_p$

Notation $C := R^{-1} = R : *_p$

$\boxed{u : A}$

$\boxed{\textbf{var}\ x, y : S \mid v : R^{-1}xy}$

$v : Ryx$

$uyx : (Ryx \Rightarrow Rxy)$

$a_1 := uyxv \ : \ Rxy$

$a_2 := \lambda x, y : S.\lambda u : R^{-1}xy.a_1 \ : \ (R^{-1} \subseteq R)$

$\boxed{\textbf{var}\ x, y : S \mid v : Rxy}$

$uxy : (Rxy \Rightarrow Ryx)$

$a_3 := uxyv \ : \ Ryx$

$a_3 : R^{-1}xy$

$a_4 := \lambda x, y : S.\lambda u : Rxy.a_3 \ : \ (R \subseteq R^{-1})$

$a_5 := \textit{rel-equal}(S, R^{-1}, R, a_2, a_4) \ : \ R^{-1} = R$

$a_6 := \lambda u : A.a_2 \ : \ A \Rightarrow B$

$a_7 := \lambda u : A.a_5 \ : \ A \Rightarrow C$

$\boxed{u : B}$

$\boxed{\textbf{var}\ x, y : S \mid v : Rxy}$

$v : R^{-1}yx$

$uyx : (R^{-1}yx \Rightarrow Ryx)$

$a_8 := uyxv \ : \ Ryx$

$a_9 := \lambda x, y : S.\lambda v : Rxy.a_8 \ : \ sym(S, R)$

$a_{10} := \lambda u : B.a_8 \ : \ (B \Rightarrow A)$

$\boxed{u : C}$

$u : R^{-1} \subseteq R \wedge R \subseteq R^{-1}$

$a_{11} := \wedge\text{-el}_1(R^{-1} \subseteq R, R \subseteq R^{-1}, u) : R^{-1} \subseteq R$

**DOI: 10.3844/jmssp.2022.148.162**

$a_{11} : B$

$a_{12} := a_{10}a_{11} \ : \ A$

$a_{13} := \lambda u : C.a_{12} \ : \ (C \Rightarrow A)$

$a_{14} := \textit{bi-impl}(A, B, a_6, a_{10}) \ : \ \textit{sym}(S, R) \Leftrightarrow R^{-1} \subseteq R$

$\textbf{\textit{sym-criterion}}(S, R) := \textit{bi-impl}(A, C, a_7, a_{13})$

$\textit{sym}(S, R) \Leftrightarrow R^{-1} = R$

4)

$\textbf{var } S : *_s \,|\, R : br(S)$

Notation $A := \textit{antisym}(S, R) : *_p$

Notation $B := R \cap R^{-1} \subseteq id_S : *_p$

$u : A$

$\textbf{var } x, y : S \mid v : (R \cap R^{-1})xy$

$v : R^{-1}xy \wedge Rxy$

$a_1 := \wedge\text{-el}_1(R^{-1}xy, Rxy, v) : R^{-1}xy$

$a_2 := \wedge\text{-el}_2(R^{-1}xy, Rxy, v) : Rxy$

$a_1 : Ryx$

$uxy : Rxy \Rightarrow Ryx \Rightarrow x = y$

$a_3 := uxya_2a_1 \ : \ (x = y)$

$a_3 : (id_S)xy$

$a_4 := \lambda x, y : S.\lambda v : (R \cap R^{-1})xy.a_3 \ : \ (R \cap R^{-1} \subseteq id_S)$

$a_4 : B$

$a_5 := \lambda u : A.a_4 \ : \ (A \Rightarrow B)$

$u : B$

$\textbf{var } x, y : S \mid v : Rxy \mid w : Ryx$

$w : R^{-1}xy$

$a_6 := \wedge\text{-in}_1(R^{-1}xy, Rxy, w, v) : (R^{-1} \cap R)xy$

$a_7 := uxya_6 \ : \ (id_S)xy$

$a_7 : x = y$

$a_8 := \lambda x, y : S.\lambda v : Rxy.\lambda w : Ryx.a_7 \ : \ \textit{antisym}(S, R)$

$a_8 : A$

$a_9 := \lambda u : B.a_8 \ : \ (B \Rightarrow A)$

$a_{10} := \textit{bi-impl}(A, B, a_5, a_9)$

$(\textit{antisym}(S, R) \Leftrightarrow (R \cap R^{-1} \subseteq id_S))$

5)

$\textbf{var } S : *_s \,|\, R : br(S)$

Notation $A := \textit{trans}(S, R) : *_p$

Notation $B := R \circ R \subseteq R : *_p$

$u : A$

$\textbf{var } x, y : S$

Notation $P := \lambda z : S.Rxz \wedge Rzy \ : \ S \to *_p$

$v : (R \circ R)xy$

$v : (\exists z : S.Pz)$

$\textbf{var } z : S \mid w : Pz$

$w : Rxz \wedge Rzy$

$a_1 := \wedge\text{-el}_1(Rxz, Rzy, w) : Rxz$

$a_2 := \wedge\text{-el}_2(Rxz, Rzy, w) : Rzy$

$a_3 := uxzya_1a_2 \ : \ Rxy$

$a_4 := \lambda z : S.\lambda w : Pz.a_3 \ : \ (\forall z : S.(Pz \Rightarrow Rxy))$

$a_5 := \exists\text{-}el(S, P, v, Rxy, a_4) \ : \ Rxy$

$a_6 := \lambda x, y : S.\lambda v : (R \circ R)xy.a_5 \ : \ (R \circ R \subseteq R)$

$a_6 : B$

$a_7 := \lambda u : A.a_6 \ : \ (A \Rightarrow B)$

$u : B$

$\textbf{var } x, y, z : S \mid v : Rxy \mid w : Ryz$

$a_8 := \textit{prod-term}(S, R, R, x, y, z, v, w) \ : \ (R \circ R)xz$

$a_9 := uxz \ : \ ((R \circ R)xz \Rightarrow Rxz)$

$a_{10} := a_9a_8 \ : \ Rxz$

$a_{11} := \lambda x, y, z : S.\lambda v : Rxy.\lambda w : Ryz.a_{10} \ : \ \textit{trans}(S, R)$

$a_{11} : A$

$a_{12} := \lambda u : B.a_{11} \ : \ (B \Rightarrow A)$

$a_{13} := \textit{bi-impl}(A, B, a_7, a_{12}) : (\textit{trans}(S, R) \Leftrightarrow (R \circ R \subseteq R))$