South American Journal of Logic
Vol. 7, n. 1, pp. 63–115, 2026
ISSN: 2446-6719

A Topological Application of Labelled Natural Deduction

Tiago M. L. de Veras, Arthur F. Ramos, Ruy J. G. B. de Queiroz
and Anjolina G. de Oliveira

Abstract

We use a labelled deduction system based on the concept of computational paths (sequences of rewrites) as equalities between two terms of the same type. We also define a term rewriting system that is used to make computations between these computational paths, establishing equalities between equalities. We then proceed to show the main result here: using this system to obtain the calculation of the fundamental group of the circle, of the torus and the real projective plane.

Keywords: Fundamental Group, Labelled Natural Deduction, Term Rewriting System, Computational Paths, Algebraic Topology.

Introduction

The identity type is arguably one of the most interesting entities of Martin-Löf type theory. From any type A, it is possible to construct the identity type $Id_A(x, y)$. This type establishes the relation of identity between two terms of A, i.e., if there is $x =_p y : A$, then p is a witness or proof that x is indeed equal to y. The proposal of the Univalence Axiom made the identity type one of the most studied aspects of type theory. It proposes that $x = y$ is equivalent to saying that $x \simeq y$, that is, the identity is an equivalence of equivalences. Another important aspect is the fact that it is possible to interpret the as paths between two points of the same space. This interpretation gives rise to the interesting interpretation of equality as a collection of homotopical paths. This connection of type theory and homotopy theory makes type theory a suitable foundation for both computation and mathematics. Nevertheless, this interpretation is only a semantical one [29] and it was not proposed with a syntactical counterpart for the concept of path in type theory. For that reason, the addition of paths to the syntax of homotopy type theory has been recently proposed by De Queiroz,

Ramos and De Oliveira [17] [24], in these works, the authors use an entity known as 'computational path', proposed by De Queiroz and Gabbay in 1994 [16], and show that it can be used to formalize the identity type in a more explicit manner.

On the other hand, one of the main interesting points of the interpretation of logical connectives via deductive systems which use a labelling system is the clear separation between a functional calculus on the labels (the names that record the steps of the proof) and a logical calculus on the formulas [11, 16]. Moreover, this interpretation has important applications. Previous publications [17, 16, 21, 23] claim that the harmony that comes with this separation makes labelled natural deduction a suitable framework to study and develop a theory of equality for natural deduction. Take, for example, the following cases taken from the λ-calculus [21]:

$$(\lambda x.(\lambda y.yx)(\lambda w.zw))v \rhd_\eta (\lambda x.(\lambda y.yx)z)v \rhd_\beta (\lambda y.yv)z \rhd_\beta zv$$
$$(\lambda x.(\lambda y.yx)(\lambda w.zw))v \rhd_\beta (\lambda x(\lambda w.zw)x)v \rhd_\eta (\lambda x.zx)v \rhd_\beta zv$$

In the theory of the $\beta\eta$-equality of λ-calculus, we can indeed say that $(\lambda x.(\lambda y.yx)(\lambda w.zw))v$ is equal to zv. Moreover, as we can see above, we have at least two ways of obtaining these equalities. We can go further, and call s the first sequence of *rewrites* that establish that $(\lambda x.(\lambda y.yx)(\lambda w.zw))v$ is indeed equal to zv. The second one, for example, we can call r. Thus, we can say that this equality is established by s and r. As we will see in this paper, we s and r are examples of an entity known as *computational path*.

Since we now have labels (computational paths) that establishes the equality between two terms, interesting questions might arise: is s different of r or are they normal forms of this equality proof? If s is equal to r, how can we prove this? We can answer questions like this when we work in a labelled natural deduction framework. The idea is that we are not limited by the calculus on the formulas, but we can also define and work with rules that apply to the labels. That way, we can use these rules to formally establish the equality between these labels, i.e., establish equalities between equalities. In this work, we will use a system proposed by [13] and known as LND_{EQ}-TRS.

In that context, the contribution of this paper will be to propose a surprising connection: it is possible to use a labelled natural deduction system together with LND_{EQ}-TRS to obtain topological results about fundamental groups.

Indeed, in this paper we will develop a theory and show that it is powerful enough to calculate the fundamental group of a circle, torus and real projective plane. For this, e use a labelled deduction system based on the concept of computational paths (sequence of rewrites). Taking into account that in mathematics [12] the calculation of this fundamental group is quite laborious, we believe our work accomplishes this calculation in a less complex form.

Nevertheless, to obtain this result we need to first formally define the concept of computational paths and define LND_{EQ}-TRS.

1 Computational Paths

In this paper, we introduce the main work tool, an entity known as computational paths. In [24], we have seen that it is possible to interpret the identity type semantically, considering the terms as homotopical paths between two points of a space. Thus, inspired by the path-based approach of the homotopy interpretation, we can use a similar approach to define the identity type in type theory, this entity is know as computational paths.

The interpretation will be similar to the homotopy case: a term $p : Id_A(a, b)$ will be a computational path between terms $a, b : A$, and such path will be the result of a sequence of rewrites. In the sequel, we shall define formally the concept of a computational path. The main idea, i.e. proofs of equality statements as (reversible) sequences of rewrites, is not new, as it goes back to a paper entitled "Equality in labeled deductive systems and the functional interpretation of propositional equality, presented in December 1993 at the *9th Amsterdam Colloquium*, and published in the proceedings in 1994[16].

Indeed, one of the most interesting aspects of the identity type is the fact that it can be used to construct higher structures. This is a rather natural consequence of the fact that it is possible to construct higher identities. For any $a, b : A$, we have type $Id_A(a, b)$. If this type is inhabited by any $p, q : Id_A(a, b)$, then we have type $Id_{Id_A(a,b)}(p, q)$. If the latter type is inhabited, we have a higher equality between p and q[4]. This concept is also present in computational paths. One can show the equality between two computational paths s and t by constructing a third one between s and t. We show in this paper a system of rules used to establish equalities between computational paths[13].

Another important question we want to answer is one that arises naturally when talking about equality: Is there a canonical proof for an expression $t_1 = t_2$? In the language of computational paths, is there a normal path between t_1 and t_2 such that every other path can be reduced to this one? In [25], it was proved that the answer is negative, this model also refutes the Uniquiness of Identity Proofs (UIP))

1.1 Introducing Computational Paths

Before we enter in details of computational paths, let's recall what motivated the introduction of computational paths to type theory. In type theory, our types are interpreted using the so-called Brower-Heyting-Kolmogorov Interpretation.

That way, a semantic interpretation of formulas are not given by truth-values, but by the concept of proof as a primitive notion. Thus, we have [17]:

a proof of the proposition:	**is given by:**
$A \wedge B$	a proof of A **and** a proof of B
$A \vee B$	a proof of A **or** a proof of B
$A \to B$	a **function** that turns a proof of A into a proof of B
$\forall x^D.P(x)$	a **function** that turns an element a into a proof of $P(a)$
$\exists x^D.P(x)$	an element a (witness) **and a proof of** $P(a)$

Also, based on the Curry-Howard functional interpretation of logical connectives, one have [17]:

a proof of the proposition:	**has the canonical form of:**
$A \wedge B$	$\langle p, q\rangle$ where p is a proof of A and q is a proof of B
$A \vee B$	$i(p)$ where p is a proof of A or $j(q)$ where q is a proof of B ('i' and 'j' abbreviate 'into the left/right disjunct')
$A \to B$	$\lambda x.b(x)$ where $b(p)$ is a proof of B provided p is a proof of A
$\forall x^A.B(x)$	$\Lambda x.f(x)$ where $f(a)$ is a proof of $B(a)$ provided a is an arbitrary individual chosen from the domain A
$\exists x^A.B(x)$	$\varepsilon x.(f(x), a)$ where a is a witness from the domain A, $f(a)$ is a proof of $B(a)$

If one looks closely, there is one interpretation missing in the BHK-Interpretation. What constitutes a proof of $t_1 = t_2$? In other words, what is a proof of an equality statement? In [17] it was proposed that an equality between these two terms should be a sequence of rewritings starting at t_1 and ending at t_2

We answer this by proposing that an equality between those two terms should be a sequence of rewrites starting from t_1 and ending at t_2. Thus, we would have [17]:

a proof of the proposition:	**is given by:**
$t_1 = t_2$	? (Perhaps a sequence of rewrites starting from t_1 and ending in t_2?)

We call computational path the sequence of rewrites between these terms.

1.2 Formal Definition

Since computational path is a generic term, it is important to emphasize the fact that we are using the term computational path in the sense defined by[21]. A computational path is based on the idea that it is possible to formally define when two computational objects $a, b : A$ are equal. These two objects are equal if one can reach b from a by applying a sequence of axioms or rules. This sequence of operations forms a path. Since it is between two computational objects, it is said that this path is a computational one. Also, an application of an axiom or a rule transforms (or rewrite) an term in another. For that reason, a computational path is also known as a sequence of rewrites. Nevertheless, before we define formally a computational path, we can take a look at one famous equality theory, the $\lambda\beta\eta - equality$[5]:

Definition 1.1 *The* $\lambda\beta\eta$-equality *is composed by the following axioms:*

(α) $\lambda x.M = \lambda y.M[y/x]$ *if* $y \notin FV(M)$;

(β) $(\lambda x.M)N = M[N/x]$;

(ρ) $M = M$;

(η) $(\lambda x.Mx) = M$ $(x \notin FV(M))$.

And the following rules of inference:

(μ) $\dfrac{M = M'}{NM = NM'}$ (τ) $\dfrac{M = N \qquad N = P}{M = P}$

(ν) $\dfrac{M = M'}{MN = M'N}$ (σ) $\dfrac{M = N}{N = M}$

(ξ) $\dfrac{M = M'}{\lambda x.M = \lambda x.M'}$

Definition 1.2 *[5]* P *is* β*-equal or* β*-convertible to* Q *(notation* $P =_\beta Q$*) iff* Q *is obtained from* P *by a finite (perhaps empty) series of* β*-contractions and reversed* β*-contractions and changes of bound variables. That is,* $P =_\beta Q$ *iff* ***there exist*** $P_0, \ldots, P_n$ $(n \geq 0)$ *such that* $P_0 \equiv P$, $P_n \equiv Q$, $(\forall i \leq n-1)(P_i \triangleright_{1\beta} P_{i+1}$ *or* $P_{i+1} \triangleright_{1\beta} P_i$ *or* $P_i \equiv_\alpha P_{i+1})$.

(Note that equality has an **existential** force, which will show in the proof rules for the identity type.)

The same happens with $\lambda\beta\eta$-equality:

Definition 1.3 *($\lambda\beta\eta$-equality[5]) The equality-relation determined by the theory $\lambda\beta\eta$ is called $=_{\beta\eta}$; that is, we define*

$$M =_{\beta\eta} N \quad \Leftrightarrow \quad \lambda\beta\eta \vdash M = N.$$

Example 1.4 *Take the term $M \equiv (\lambda x.(\lambda y.yx)(\lambda w.zw))v$. Then, it is $\beta\eta$-equal to $N \equiv zv$ because of the sequence:*
$(\lambda x.(\lambda y.yx)(\lambda w.zw))v, \quad (\lambda x.(\lambda y.yx)z)v, \quad (\lambda y.yv)z, \quad zv$
which starts from M and ends with N, and each member of the sequence is obtained via 1-step β- or η-contraction of a previous term in the sequence. To take this sequence into a path*, one has to apply transitivity twice, as we do in the example below.*

Example 1.5 *The term $M \equiv (\lambda x.(\lambda y.yx)(\lambda w.zw))v$ is $\beta\eta$-equal to $N \equiv zv$ because of the sequence:*
$(\lambda x.(\lambda y.yx)(\lambda w.zw))v, \quad (\lambda x.(\lambda y.yx)z)v, \quad (\lambda y.yv)z, \quad zv$
Now, taking this sequence into a path leads us to the following:
The first is equal to the second based on the grounds:
$\eta((\lambda x.(\lambda y.yx)(\lambda w.zw))v, (\lambda x.(\lambda y.yx)z)v)$
The second is equal to the third based on the grounds:
$\beta((\lambda x.(\lambda y.yx)z)v, (\lambda y.yv)z)$
Now, the first is equal to the third based on the grounds:
$\tau(\eta((\lambda x.(\lambda y.yx)(\lambda w.zw))v, (\lambda x.(\lambda y.yx)z)v), \beta((\lambda x.(\lambda y.yx)z)v, (\lambda y.yv)z))$
Now, the third is equal to the fourth one based on the grounds:
$\beta((\lambda y.yv)z, zv)$
Thus, the first one is equal to the fourth one based on the grounds:
$\tau(\tau(\eta((\lambda x.(\lambda y.yx)(\lambda w.zw))v, (\lambda x.(\lambda y.yx)z)v), \beta((\lambda x.(\lambda y.yx)z)v, (\lambda y.yv)z)), \beta((\lambda y.yv)z, zv))).$

The aforementioned theory establishes the equality between two λ-terms. Since we are working with computational objects as terms of a type, we can consider the following definition:

Definition 1.6 *The equality theory of Martin Löf's type theory has the following basic proof rules for the Π-type:*

$$(\beta) \quad \frac{\begin{array}{c}[x:A]\\ N:A \qquad M:B\end{array}}{(\lambda x.M)N = M[N/x] : B[N/x]} \qquad (\xi) \quad \frac{\begin{array}{c}[x:A]\\ M = M' : B\end{array}}{\lambda x.M = \lambda x.M' : (\Pi x:A)B}$$

$$(\rho) \quad \frac{M:A}{M = M : A} \qquad (\mu) \quad \frac{M = M' : A \qquad N : (\Pi x:A)B}{NM = NM' : B[M/x]}$$

$$(\sigma) \quad \frac{M = N : A}{N = M : A} \qquad (\nu) \quad \frac{N : A \qquad M = M' : (\Pi x:A)B}{MN = M'N : B[N/x]}$$

$$(\tau) \qquad \frac{M = N : A \qquad N = P : A}{M = P : A}$$

$$(\eta) \qquad \frac{M : (\Pi x : A)B}{(\lambda x.Mx) = M : (\Pi x : A)B}\ (x \notin FV(M))$$

We are finally able to formally define computational paths:

Definition 1.7 *Let a and b be elements of a type A. Then, a* computational path *s from a to b is a composition of rewrites (each rewrite is an application of the inference rules of the equality theory of type theory or is a change of bound variables). We denote that by $a =_s b$.*

As we have seen in *example 1.5*, composition of rewrites are applications of the rule τ. Since change of bound variables is possible, each term is considered up to α-equivalence.

1.3 Equality Equations

One can use the aforementioned axioms to show that computational paths establishes the three fundamental equations of equality: the reflexivity, symmetry and transitivity:

$$\frac{a =_t b : A \qquad b =_u c : A}{a =_{\tau(t,u)} c : A}\ \textit{transitivity} \qquad \frac{a : A}{a =_\rho a : A}\ \textit{reflexivity}$$

$$\frac{a =_t b : A}{b =_{\sigma(t)} a : A}\ \textit{symmetry}$$

1.4 Identity Type

We have said that it is possible to formulate the identity type using computational paths. As we have seen, the best way to define any formal entity of type theory is by a set of natural deductions rules. Thus, we define our path-based approach as the following set of rules:

- Formation and Introduction rules [17, 24]:

$$\frac{A \text{ type} \qquad a : A \qquad b : A}{Id_A(a,b) \text{ type}}\ Id - F$$

$$\frac{a =_s b : A}{s(a,b) : Id_A(a,b)}\ Id - I$$

One can notice that our formation rule is exactly equal to the traditional identity type. From terms $a, b : A$, one can form that is inhabited only if there is a proof of equality between those terms, i.e., $Id_A(a, b)$.

The difference starts with the introduction rule. In our approach, one can notice that we do not use a reflexive constructor r. In other words, the reflexive path is not the main building block of our identity type. Instead, if we have a computational path $a =_s b : A$, we introduce $s(a, b)$ as a term of the identity type. That way, one should see $s(a, b)$ as a sequence of rewrites and substitutions (i.e., a computational path) which would have started from a and arrived at b

- Elimination rule [17, 24]:

$$\frac{\begin{array}{cc} & [a =_g b : A] \\ m : Id_A(a, b) & h(g) : C \end{array}}{REWR(m, \acute{g}.h(g)) : C}\, Id - E$$

Let's recall the notation being used. First, one should see $h(g)$ as a functional expression h which depends on g. Also, one should notice the use of '$\acute{}$' in $\acute{g}$. One should see '$\acute{}$' as an abstractor that binds the occurrences of the variable g introduced in the local assumption $[a =_g b : A]$ as a kind of *Skolem-type* constant denoting the *reason* why a was assumed to be equal to b.

We also introduce the constructor $REWR$. In a sense, it is similar to the constructor J of the traditional approach, since both arise from the elimination rule of the identity type. The behavior of $REWR$ is simple. If from a computational path g that establishes the equality between a and b one can construct $h(g) : C$, then if we also have this equality established by a term C, we can put together all this information in $REWR$ to construct C, eliminating the type $Id_A(a, b)$ in the process. The idea is that we can substitute g for m in $\acute{g}.h(g)$, resulting in $h(m/g) : C$. This behavior is established next by the reduction rule.

- Reduction rule [17, 24]:

$$\frac{\dfrac{\dfrac{a =_m b : A}{m(a, b) : Id_A(a, b)}\, Id - I \qquad \begin{array}{c}[a =_g b : A] \\ h(g) : C\end{array}}{REWR(m, \acute{g}.h(g)) : C}\, Id - E}{\begin{array}{c}[a =_m b : A] \\ h(m/g) : C\end{array}} \quad \triangleright_\beta$$

- Induction rule:

$$\frac{e : Id_A(a,b) \qquad \dfrac{[a =_t b : A]}{t(a,b) : Id_A(a,b)}\, Id-I}{REWR(e, \acute{t}.t(a,b)) : Id_A(a,b)}\, Id-E \quad \rhd_\eta \quad e : Id_A(a,b)$$

Our introduction and elimination rules reassure the concept of equality as an **existential force**. In the introduction rule, we encapsulate the idea that a witness of a identity type $Id_A(a,b)$ only exists if there exist a computational path establishing the equality of a and b. Also, one can notice that elimination rule is similar to the elimination rule of the existential quantifier.

1.5 Path-based Examples

The objective of this subsection is to show how to use in practice the rules that we have just defined. The idea is to show construction of terms of some important types. The constructions that we have chosen to build are the reflexive, transitive and symmetric type of the identity type. Those were not random choices. The main reason is the fact that reflexive, transitive and symmetric types are essential to the process of building a groupoid model for the identity type[6]. As we shall see, these constructions come naturally from simple computational paths constructed by the application of axioms of the equality of type theory, as we have done in our construction of the type of computational paths as a weak groupoid [27].

Before we start the constructions, we think that it is essential to understand how to use the eliminations rules. The process of building a term of some type is a matter of finding the right reason. In the case of J, the reason is the correct $x, y : A$ and $z : Id_A(a,b)$ that generates the adequate $C(x,y,z)$. In our approach, the reason is the correct path $a =_g b$ that generates the adequate $g(a,b) : Id(a,b)$.

1.5.1 Reflexivity

One could find strange the fact that we need to prove the reflexivity. Nevertheless, just remember that our approach is not based on the idea that reflexivity is the base of the identity type. As usual in type theory, a proof of something comes down to a construction of a term of a type. In this case, we need to construct a term of type $\Pi_{(a:A)} Id_A(a,a)$. The reason is extremely simple: from a term $a : A$, we obtain the computational path $a =_\rho a : A$ [24]:

$$\dfrac{\dfrac{\dfrac{[a:A]}{a =_\rho a : A}}{\rho(a,a) : Id_A(a,a)}\, Id - I}{\lambda a.\rho(a,a) : \Pi_{(a:A)} Id_A(a,a)}\, \Pi - I$$

1.5.2 Symmetry

The second proposed construction is the symmetry. Our objective is to obtain a term of type $\Pi_{(a:A)}\Pi_{(b:A)}(Id_A(a,b) \to Id_A(b,a))$.

We construct a proof using computational paths. As expected, we need to find a suitable reason. Starting from $a =_t b$, we could look at the axioms of *definition 4.1* to plan our next step. One of those axioms makes the symmetry clear: the σ axiom. If we apply σ, we will obtain $b =_{\sigma(t)} a$. From this, we can then infer that Id_A is inhabited by $(\sigma(t))(b,a)$. Now, it is just a matter of applying the elimination [24]:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[a:A] \quad [b:A] \qquad \dfrac{\dfrac{[a =_t b : A]}{b =_{\sigma(t)} a : A}}{(\sigma(t))(b,a) : Id_A(b,a)}\, Id - I}{[p(a,b) : Id_A(a,b)]}}{REWR(p(a,b), \acute{t}.(\sigma(t))(b,a)) : Id_A(b,a)}\, Id - E}{\lambda p.REWR(p(a,b), \acute{t}.(\sigma(t))(b,a)) : Id_A(a,b) \to Id_A(b,a)}\, \to -I}{\lambda b.\lambda p.REWR(p(a,b), \acute{t}.(\sigma(t))(b,a)) : \Pi_{(b:A)}(Id_A(a,b) \to Id_A(b,a))}\, \Pi - I}{\lambda a.\lambda b.\lambda p.REWR(p(a,b), \acute{t}.(\sigma(t))(b,a)) : \Pi_{(a:A)}\Pi_{(b:A)}(Id_A(a,b) \to Id_A(b,a))}\, \Pi - I$$

1.5.3 Transitivity

The third and last construction will be the transitivity. Our objective is to obtain a term of type

$$\Pi_{(a:A)}\Pi_{(b:A)}\Pi_{(c:A)}(Id_A(a,b) \to Id_A(b,c) \to Id_A(a,c)).$$

To build our path-based construction, the first step, as expected, is to find the reason. Since we are trying to construct the transitivity, it is natural to think that we should start with paths $a =_t b$ and $b =_u c$ and then, from these paths, we should conclude that there is a path z that establishes that $a =_z c$. To obtain z, we could try to apply the axioms of *definition 4.1*. Looking at the axioms, one is exactly what we want: the axiom τ. If we apply τ to $a =_t b$ and $b =_u c$, we will obtain a new path $\tau(t,u)$ such that $a =_{\tau(t,u)} c$. Using that construction as the reason, we obtain the following term [24]:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[a =_t b : A] \quad [b =_u c : A]}{a =_{\tau(t,u)} c : A}}{[c : A]} \quad \dfrac{}{(\tau(t,u))(a,c) : Id_A(a,c)}\, Id-I}{[a : A] \quad [b : A] \quad [s(b,c) : Id_A(b,c)]} \quad \dfrac{}{REWR(s(b,c), \acute{u}(\tau(t,u))(a,c)) : Id_A(a,c)}\, Id-E}{[w(a,b) : Id_A(a,b)]} \quad \dfrac{}{REWR(w(a,b), \acute{t}REWR(s(b,c), \acute{u}(\tau(t,u))(a,c))) : Id_A(a,c)}\, Id-E}{\lambda s.REWR(w(a,b), \acute{t}REWR(s(b,c), \acute{u}(\tau(t,u))(a,c))) : Id_A(b,c) \to Id_A(a,c)}\, \to -I}{\lambda w.\lambda s.REWR(w(a,b), \acute{t}REWR(s(b,c), \acute{u}(\tau(t,u))(a,c))) : Id_A(a,b) \to Id_A(b,c) \to Id_A(a,c)}\, \Pi - I}{\lambda c.\lambda w.\lambda s.REWR(w(a,b), \acute{t}REWR(s(b,c), \acute{u}(\tau(t,u))(a,c))) : \Pi_{(c:A)}(Id_A(a,b) \to Id_A(b,c) \to Id_A(a,c))}\, \Pi - I}{\lambda b.\lambda c.\lambda w.\lambda s.REWR(w(a,b), \acute{t}REWR(s(b,c), \acute{u}(\tau(t,u))(a,c))) : \Pi_{(b:A)}\Pi_{(c:A)}(Id_A(a,b) \to Id_A(b,c) \to Id_A(a,c))}\, \Pi - I}{\lambda a.\lambda b.\lambda c.\lambda w.\lambda s.REWR(w(a,b), \acute{t}REWR(s(b,c), \acute{u}(\tau(t,u))(a,c))) : \Pi_{(a:A)}\Pi_{(b:A)}\Pi_{(c:A)}(Id_A(a,b) \to Id_A(b,c) \to Id_A(a,c))}\, \Pi - I$$

As one can see, each step is just straightforward applications of introduction, elimination rules and abstractions. The only idea behind this construction is just the simple fact that the axiom τ guarantees the transitivity of paths.

1.6 Term Rewrite System

As we have just shown, a computational path establishes when two terms of the same type are equal. From the theory of computational paths, an interesting case arises. Suppose we have a path s that establishes that $a =_s b : A$ and a path t that establishes that $a =_t b : A$. Consider that s and t are formed by distinct compositions of rewrites. Is it possible to conclude that there are cases that s and t should be considered equivalent? The answer is *yes*. Consider the following examples [25]:

Example 1.8 Consider the path $a =_t b : A$. By the symmetric property, we obtain $b =_{\sigma(t)} a : A$. What if we apply the property again on the path $\sigma(t)$? We would obtain a path $a =_{\sigma(\sigma(t))} b : A$. Since we applied symmetry twice in succession, we obtained a path that is equivalent to the initial path t. For that reason, we would like to conclude the act of applying symmetry twice in succession is a redundancy. We say that the path $\sigma(\sigma(t))$ reduce to the path t.

Example 1.9 Consider the reflexive path $a =_\rho a : A$. It one applies the symmetric axiom, one ends up with $a =_{\sigma(\rho)} a : A$. Thus, the obtained path is equivalent to the initial one, since the symmetry was applied to the reflexive path. Therefore, $\sigma(\rho)$ is a redundant way of expressing the path ρ. Thus, $\sigma(\rho)$ should be reduced to ρ.

Example 1.10 Consider a path $a =_t b : A$. Applying the symmetry, one ends up with $b =_{\sigma(t)} a : A$. One can take those two paths and apply the transitivity, ending up with $a =_{\tau(t,\sigma(t))} a$. Since the path τ is the inverse of the $\sigma(\tau)$, the composition of those two paths should be equivalent to the reflexive path. Thus, $\tau(t, \sigma(t))$ should be reduced to ρ.

As one could see in the aforementioned examples, different paths should be considered equal if one is just a redundant form of the other. The examples that we have just seen are just straightforward and simple cases. Since the equality theory has a total of 7 axioms, the possibility of combinations that could generate redundancies are high. Fortunately, all possible redundancies were thoroughly mapped by[13]. In that work, a system that establishes all redundancies and creates rules that solve them was proposed. This system, known as $LND_{EQ} - TRS$, maps a total of 39 rules that solve redundancies.

1.7 LND-EQ-TRS

In this subsection, we show the rules that compose the $LND_{EQ} - TRS$. All those rules comes from the mapping of redundancies between computational paths, as we have seen in the 3 previous examples.

1.7.1 Subterm Substitution

Before we introduce the rewriting rules, it is important to introduce the concept of subterm substitution. In Equational Logic, the subterm substitution is given by the following inference rule[18]:

$$\frac{s = t}{s\theta = t\theta}$$

where θ is a substitution. One problem is that such rule does not respect the sub-formula property. To deal with that,[1] proposes two inference rules:

$$\frac{M = N \qquad C[N] = O}{C[M] = O}\, IL \qquad \frac{M = C[N] \qquad N = O}{M = C[O]}\, IR$$

where M, N and O are terms.

As proposed in [17], we can define similar rules using computational paths, as follows:

$$\frac{x =_r \mathcal{C}[y] : A \qquad y =_s u : A'}{x =_{\mathtt{sub_L}(r,s)} \mathcal{C}[u] : A} \qquad \frac{x =_r w : A' \qquad \mathcal{C}[w] =_s u : A}{\mathcal{C}[x] =_{\mathtt{sub_R}(r,s)} u : A}$$

where C is the context in which the sub-term detached by '[ ]' appears and A' could be a sub-domain of A, equal to A or disjoint to A.

In the rule above, $\mathcal{C}[u]$ should be understood as the result of replacing every occurrence of y by u in C.

1.7.2 Rewriting Rules

In this subsection, our objective is to show all rewrite reductions and their associated rewriting rules. The idea is to analyze all possible occurrences of redundancies in proofs which involves the rules of rewritings.

We start with the transitivity:

Definition 1.11 (reductions involving τ [17])

$$\frac{x =_r y : A \quad y =_{\sigma(r)} x : A}{x =_{\tau(r,\sigma(r))} x : A} \quad \triangleright_{tr} \quad x =_\rho x : A$$

$$\frac{y =_{\sigma(r)} x : A \quad x =_r y : A}{y =_{\tau(\sigma(r),r)} y : A} \quad \triangleright_{tsr} \quad y =_\rho y : A$$

$$\frac{u =_r v : A \quad v =_\rho v : A}{u =_{\tau(r,\rho)} v : A} \quad \triangleright_{trr} \quad u =_r v : A$$

$$\frac{u =_\rho u : A \quad u =_r v : A}{u =_{\tau(\rho,r)} v : A} \quad \triangleright_{tlr} \quad u =_r v : A$$

Associated rewriting rules:

$$\tau(r, \sigma(r)) \triangleright_{tr} \rho$$

$$\tau(\sigma(r), r) \triangleright_{tsr} \rho$$

$$\tau(r, \rho) \triangleright_{trr} r$$

$$\tau(\rho, r) \triangleright_{tlr} r.$$

These reductions can be generalized to transformations where the reasons r and $\sigma(r)$ (transf. 1 and 2) and r and ρ (transf. 3 and 4) appear in some context, as illustrated by the following example: [17]:

Example 1.12

$$\frac{\dfrac{x =_r y : A}{i(x) =_{\xi_1(r)} i(y) : A + B} \quad \dfrac{\dfrac{x =_r y : A}{y =_{\sigma(r)} x : A}}{i(y) =_{\xi_1(\sigma(r))} i(x) : A + B}}{i(x) =_{\tau(\xi_1(r),\xi_1(\sigma(r)))} i(x) : A + B} \quad \triangleright_{tr} \quad \frac{x =_r y : A}{i(x) =_{\xi_1(r)} i(y) : A + B}$$

Associated rewriting: $\tau(\xi_1(r), \xi_1(\sigma(r))) \triangleright_{tr} \xi_1(r)$.

For the general context $\mathcal{C}[\ ]$:
Associated rewritings:
$\tau(\mathcal{C}[r], \mathcal{C}[\sigma(r)]) \triangleright_{tr} \mathcal{C}[\rho]$
$\tau(\mathcal{C}[\sigma(r)], \mathcal{C}[r]) \triangleright_{tsr} \mathcal{C}[\rho]$
$\tau(\mathcal{C}[r], \mathcal{C}[\rho]) \triangleright_{trr} \mathcal{C}[r]$
$\tau(\mathcal{C}[\rho], \mathcal{C}[r]) \triangleright_{tlr} \mathcal{C}[r]$

The transitivity rules are pretty straightforward. We have more complicated cases [17]:

Definition 1.13

$$\dfrac{a : A \quad \dfrac{\begin{array}{c}[x : A]\\ \vdots \\ b(x) =_r g(x) : B\end{array}}{\lambda x.b(x) =_{\xi(r)} \lambda x.g(x) : A \to B}\ \to\text{-}intr}{APP(\lambda x.b(x), a) =_{\nu(\xi(r))} APP(\lambda x.g(x), a) : B}\ \to\text{-}elim$$

$$\triangleright_{mxl} \quad \begin{array}{c} a : A \\ b(a/x) =_r g(a/x) : B \end{array}$$

Associated rewriting rule:

$$\nu(\xi(r)) \triangleright_{mxl} r.$$

Definition 1.14 (reductions involving ρ and σ [17])

$$\dfrac{x =_\rho x : A}{x =_{\sigma(\rho)} x : A} \quad \triangleright_{sr} \quad x =_\rho x : A$$

$$\dfrac{\dfrac{x =_r y : A}{y =_{\sigma(r)} x : A}}{x =_{\sigma(\sigma(r))} y : A} \quad \triangleright_{sr} \quad x =_r y : A$$

Associated rewritings:
$\sigma(\rho) \triangleright_{sr} \rho$
$\sigma(\sigma(r)) \triangleright_{sr} r$

Definition 1.15 (Substitution rules [17])

$$\dfrac{u =_r \mathcal{C}[x] : A \quad x =_\rho x : A'}{u =_{\mathtt{sub}_{\mathtt{L}}(r,\rho)} \mathcal{C}[x] : A} \quad \triangleright_{slr} \quad u =_r \mathcal{C}[x] : A$$

$$\dfrac{x =_\rho x : A' \quad \mathcal{C}[x] =_r z : A}{\mathcal{C}[x] =_{\mathtt{sub}_{\mathtt{R}}(\rho,r)} z : A} \quad \triangleright_{srr} \quad \mathcal{C}[x] =_r z : A$$

$$\dfrac{\dfrac{z =_s \mathcal{C}[y] : A \quad y =_r w : A'}{z =_{\mathtt{sub}_{\mathtt{L}}(s,r)} \mathcal{C}[w] : D} \quad \dfrac{y =_r w : A'}{w =_{\sigma(r)} y : D'}}{z =_{\mathtt{sub}_{\mathtt{L}}(\mathtt{sub}_{\mathtt{L}}(s,r),\sigma(r))} \mathcal{C}[y] : A}\ \triangleright_{sls} \quad z =_s \mathcal{C}[y] : A$$

$$\frac{\dfrac{z =_s \mathcal{C}[y] : A \quad y =_r w : A'}{z =_{\mathtt{sub_L}(s,r)} \mathcal{C}[w] : A} \qquad \dfrac{y =_r w : A'}{w =_{\sigma(r)} y : A'}}{z =_{\mathtt{sub_L}(\mathtt{sub_L}(s,r),\sigma(r))} \mathcal{C}[y] : A} \rhd_{slss} \; z =_s \mathcal{C}[y] : A$$

$$\frac{x =_s w : A' \qquad \dfrac{\dfrac{x =_s w : A'}{w =_{\sigma(s)} x : A'} \quad \mathcal{C}[x] =_r z : A}{\mathcal{C}[w] =_{\mathtt{sub_R}(\sigma(s),r)} z : A}}{\mathcal{C}[x] =_{\mathtt{sub_R}(s,\mathtt{sub_R}(\sigma(s),r))} z : A} \rhd_{srs} \; \mathcal{C}[x] =_r z : A$$

$$\frac{\dfrac{x =_s w : A'}{w =_{\sigma(s)} x : A'} \qquad \dfrac{x =_s w : A' \quad \mathcal{C}[w] =_r z : A}{\mathcal{C}[x] =_{\mathtt{sub_R}(s,r)} z : A}}{\mathcal{C}[w] =_{\mathtt{sub_R}(\sigma(s),\mathtt{sub_R}(s,r))} z : A} \rhd_{srrr} \; \mathcal{C}[w] =_r z : A$$

Associated rewritings:
$\mathtt{sub_L}(\mathcal{C}[r], \mathcal{C}[\rho]) \rhd_{slr} \mathcal{C}[r]$
$\mathtt{sub_R}(\mathcal{C}[\rho], \mathcal{C}[r]) \rhd_{srr} \mathcal{C}[r]$
$\mathtt{sub_L}(\mathtt{sub_L}(s, \mathcal{C}[r]), \mathcal{C}[\sigma(r)]) \rhd_{sls} s$
$\mathtt{sub_L}(\mathtt{sub_L}(s, \mathcal{C}[\sigma(r)]), \mathcal{C}[r]) \rhd_{slss} s$
$\mathtt{sub_R}(s, \mathtt{sub_R}(\mathcal{C}[\sigma(s)], r)) \rhd_{srs} r$
$\mathtt{sub_R}(\mathcal{C}[\sigma(s)], \mathtt{sub_R}(\mathcal{C}[s], r)) \rhd_{srrr} r$

Definition 1.16 ([17])

$$\frac{\dfrac{\dfrac{x =_r y : A \qquad z : B}{\langle x, z\rangle =_{\xi_1(r)} \langle y, z\rangle : A \times B} \times\text{-}intr}{FST(\langle x, z\rangle) =_{\mu_1(\xi_1(r))} FST(\langle y, z\rangle) : A} \times\text{-}elim}{\rhd_{mx2l} \quad x =_r y : A}$$

$$\frac{\dfrac{\dfrac{x =_r x' : A \qquad y =_s z : B}{\langle x, y\rangle =_{\xi_\wedge(r,s)} \langle x', z\rangle : A \times B} \times\text{-}intr}{FST(\langle x, y\rangle) =_{\mu_1(\xi_\wedge(r,s))} FST(\langle x', z\rangle) : A} \times\text{-}elim}{\rhd_{mx2l} \quad x =_r x' : A}$$

$$\frac{\dfrac{\dfrac{x =_r y : A \qquad z =_s w : B}{\langle x, z\rangle =_{\xi_\wedge(r,s)} \langle y, w\rangle : A \times B} \times\text{-}intr}{SND(\langle x, z\rangle) =_{\mu_2(\xi_\wedge(r,s))} SND(\langle y, w\rangle) : B} \times\text{-}elim}{\rhd_{mx2r} \quad z =_s w : B}$$

$$\dfrac{\dfrac{x : A \qquad z =_s w : B}{\langle x, z\rangle =_{\xi_2(s)} \langle x, w\rangle : A \times B}\ \times\text{-}intr}{SND(\langle x, z\rangle) =_{\mu_2(\xi_2(s))} SND(\langle x, w\rangle) : B}\ \times\text{-}elim$$

$$\triangleright_{mx2r} \quad z =_s w : B$$

Associated rewritings:
$\mu_1(\xi_1(r)) \triangleright_{mx2l1} r$
$\mu_1(\xi_\wedge(r, s)) \triangleright_{mx2l2} r$
$\mu_2(\xi_\wedge(r, s)) \triangleright_{mx2r1} s$
$\mu_2(\xi_2(s)) \triangleright_{mx2r2} s$

β_{rewr}-+-*reduction*

$$\dfrac{\dfrac{a =_r a' : A}{i(a) =_{\xi_1(r)} i(a') : A + B}\ +\ \text{-}intr \qquad \begin{matrix}[x : A] & [y : B] \\ f(x) =_s k(x) : C & g(y) =_u h(y) : C\end{matrix}}{D(i(a), \acute{x}f(x), \acute{y}g(y)) =_{\mu(\xi_1(r),s,u)} D(i(a'), \acute{x}k(x), \acute{y}h(y)) : C}\ +\ \text{-}elim$$

$$\triangleright_{mx3l} \quad \begin{matrix}a =_r a' : A \\ f(a/x) =_s k(a'/x) : C\end{matrix}$$

$$\dfrac{\dfrac{b =_r b' : B}{j(b) =_{\xi_2(r)} j(b') : A + B}\ +\ \text{-}intr \qquad \begin{matrix}[x : A] & [y : B] \\ f(x) =_s k(x) : C & g(y) =_u h(y) : C\end{matrix}}{D(j(b), \acute{x}f(x), \acute{y}g(y)) =_{\mu(\xi_2(r),s,u)} D(j(b'), \acute{x}k(x), \acute{y}h(y)) : C}\ +\ \text{-}elim$$

$$\triangleright_{mx3r} \quad \begin{matrix}b =_s b' : B \\ g(b/y) =_u h(b'/y) : C\end{matrix}$$

Associated rewritings:
$\mu(\xi_1(r), s, u) \triangleright_{mx3l} s$
$\mu(\xi_2(r), s, u) \triangleright_{mx3r} u$

β_{rewr}-Π-*reduction*

$$\dfrac{a : A \qquad \dfrac{\begin{matrix}[x : A] \\ f(x) =_r g(x) : B(x)\end{matrix}}{\lambda x.f(x) =_{\xi(r)} \lambda x.g(x) : \Pi x : A.B(x)}}{APP(\lambda x.f(x), a) =_{\nu(\xi(r))} APP(\lambda x.g(x), a) : B(a)}$$

$$\triangleright_{mxl} \quad \begin{matrix}a : A \\ f(a/x) =_r g(a/x) : B(a)\end{matrix}$$

Associated rewriting:
$\nu(\xi(r)) \triangleright_{mxl} r$

β_{rewr}-Σ-*reduction*

$$\dfrac{\dfrac{a =_r a' : A \quad f(a) : B(a)}{\varepsilon x.(f(x), a) =_{\xi_1(r)} \varepsilon x.(f(x), a') : \Sigma x : A.B(x)} \quad \dfrac{[t : A, g(t) : B(t)]}{d(g,t) =_s h(g,t) : C}}{E(\varepsilon x.(f(x), a), \acute{g}\acute{t}d(g,t)) =_{\mu(\xi_1(r),s)} E(\varepsilon x.(f(x), a'), \acute{g}\acute{t}h(g,t)) : C} \triangleright_{mxr} \dfrac{a =_r a' : A \quad f(a) : B(a)}{d(f/g, a/t) =_s h(f/g, a'/t) : C}$$

$$\dfrac{\dfrac{a : A \quad f(a) =_r i(a) : B(a)}{\varepsilon x.(f(x), a) =_{\xi_2(r)} \varepsilon x.(i(x), a) : \Sigma x : A.B(x)} \quad \dfrac{[t : A, g(t) : B(t)]}{d(g,t) =_s h(g,t) : C}}{E(\varepsilon x.(f(x), a), \acute{g}\acute{t}d(g,t)) =_{\mu(\xi_2(r),s)} E(\varepsilon x.(i(x), a), \acute{g}\acute{t}h(g,t)) : C} \triangleright_{mxl} \dfrac{a : A \quad f(a) =_r i(a) : B(a)}{d(f/g, a/t) =_s h(i/g, a/t) : C}$$

Associated rewritings:
$\mu(\xi_1(r), s) \triangleright_{mxr} s$
$\mu(\xi_2(r), s) \triangleright_{mxl} s$

Definition 1.17 (η_{rewr} **[17]**)

η_{rewr}- $\times$-*reduction*

$$\dfrac{\dfrac{x =_r y : A \times B}{FST(x) =_{\mu_1(r)} FST(y) : A} \times\text{-}elim \quad \dfrac{x =_r y : A \times B}{SND(x) =_{\mu_2(r)} SND(y) : B} \times\text{-}elim}{\langle FST(x), SND(x)\rangle =_{\xi(\mu_1(r),\mu_2(r))} \langle FST(y), SND(y)\rangle : A \times B} \times\text{-}intr \triangleright_{mx} x =_r y : A \times B$$

η_{rewr}- +-*reduction*

$$\dfrac{c =_t d : A + B \quad \dfrac{[a_1 =_r a_2 : A]}{i(a_1) =_{\xi_1(r)} i(a_2) : A + B} +\text{-}intr \quad \dfrac{[b_1 =_s b_2 : B]}{j(b_1) =_{\xi_2(s)} j(b_2) : A + B} +\text{-}intr}{D(c, \acute{a_1}i(a_1), \acute{b_1}j(b_1)) =_{\mu(t,\xi_1(r),\xi_2(s))} D(d, \acute{a_2}i(a_2), \acute{b_2}j(b_2))} +\text{-}elim \triangleright_{mxx} c =_t d : A + B$$

Π-η_{rewr}-*reduction*

$$\dfrac{\dfrac{[t : A] \quad c =_r d : \Pi x : A.B(x)}{APP(c,t) =_{\nu(r)} APP(d,t) : B(t)} \Pi\text{-}elim}{\lambda t.APP(c,t) =_{\xi(\nu(r))} \lambda t.APP(d,t) : \Pi t : A.B(t)} \Pi\text{-}intr \triangleright_{xmr} c =_r d : \Pi x : A.B(x)$$

where c and d do not depend on x.

Σ-η_{rewr}-*reduction*

$$\dfrac{c =_s b : \Sigma x : A.B(x) \quad \dfrac{[t : A] \quad [g(t) =_r h(t) : B(t)]}{\varepsilon y.(g(y), t) =_{\xi_2(r)} \varepsilon y.(h(y), t) : \Sigma y : A.B(y)}\Sigma\text{-}intr}{\dfrac{E(c, \acute{g}\acute{t}\varepsilon y.(g(y), t)) =_{\mu(s,\xi_2(r))} E(b, \acute{h}\acute{t}\varepsilon y.(h(y), t)) : \Sigma y : A.B(y)}{\triangleright_{mxlr} \quad c =_s b : \Sigma x : A.B(x)}\Sigma\text{-}elim}$$

Associated rewritings:
$\xi(\mu_1(r), \mu_2(r)) \triangleright_{mx} r$
$\mu(t, \xi_1(r), \xi_2(s)) \triangleright_{mxx} t$
$\xi(\nu(r)) \triangleright_{xmr} r$
$\mu(s, \xi_2(r)) \triangleright_{mxlr} s$

Definition 1.18 (σ and τ [17])

$$\dfrac{\dfrac{x =_r y : A \quad y =_s w : A}{x =_{\tau(r,s)} w : A}}{w =_{\sigma(\tau(r,s))} x : A} \quad \triangleright_{stss} \quad \dfrac{\dfrac{y =_s w : A}{w =_{\sigma(s)} y : A} \quad \dfrac{x =_r y : A}{y =_{\sigma(r)} x : A}}{w =_{\tau(\sigma(s),\sigma(r))} x : A}$$

Associated rewriting:
$\sigma(\tau(r, s)) \triangleright_{stss} \tau(\sigma(s), \sigma(r))$

Definition 1.19 (σ and sub [17])

$$\dfrac{\dfrac{x =_r \mathcal{C}[y] : A \quad y =_s w : A'}{x =_{\mathtt{sub_L}(r,s)} \mathcal{C}[w] : A}}{\mathcal{C}[w] =_{\sigma(\mathtt{sub_L}(r,s))} x : A} \quad \triangleright_{ssbl} \quad \dfrac{\dfrac{y =_s w : A'}{w =_{\sigma(s)} y : A'} \quad \dfrac{x =_r \mathcal{C}[y] : A}{\mathcal{C}[y] =_{\sigma(r)} x : A}}{\mathcal{C}[w] =_{\mathtt{sub_R}(\sigma(s),\sigma(r))} x : A}$$

$$\dfrac{\dfrac{x =_r y : A' \quad \mathcal{C}[y] =_s w : A}{\mathcal{C}[x] =_{\mathtt{sub_R}(r,s)} w : A}}{w =_{\sigma(\mathtt{sub_R}(r,s))} \mathcal{C}[x] : D} \quad \triangleright_{ssbr} \quad \dfrac{\dfrac{\mathcal{C}[y] =_s w : A}{w =_{\sigma(s)} \mathcal{C}[y] : A} \quad \dfrac{x =_r y : A'}{y =_{\sigma(r)} x : A'}}{w =_{\mathtt{sub_L}(\sigma(s),\sigma(r))} \mathcal{C}[x] : A}$$

Associated rewritings:
$\sigma(\mathtt{sub_L}(r, s)) \triangleright_{ssbl} \mathtt{sub_R}(\sigma(s), \sigma(r))$
$\sigma(\mathtt{sub_R}(r, s)) \triangleright_{ssbr} \mathtt{sub_L}(\sigma(s), \sigma(r))$

Definition 1.20 (σ and ξ [17])

$$\frac{\dfrac{x =_r y : A}{i(x) =_{\xi_1(r)} i(y) : A + B}}{i(y) =_{\sigma(\xi_1(r))} i(x) : A + B} \triangleright_{sx} \frac{\dfrac{x =_r y : A}{y =_{\sigma(r)} x : A}}{i(y) =_{\xi_1(\sigma(r))} i(x) : A + B}$$

$$\frac{\dfrac{x =_r y : A \quad z =_s w : B}{\langle x, z\rangle =_{\xi(r,s)} \langle y, w\rangle : A \times B}}{\langle y, w\rangle =_{\sigma(\xi(r,s))} \langle x, z\rangle : A \times B} \triangleright_{sxss} \frac{\dfrac{x =_r y : A}{y =_{\sigma(r)} x : A} \quad \dfrac{z =_s w : B}{w =_{\sigma(s)} z : B}}{\langle y, w\rangle =_{\xi(\sigma(r),\sigma(s))} \langle x, z\rangle : A \times B}$$

$$\frac{\dfrac{\dfrac{[x : A]}{f(x) =_s g(x) : B(x)}}{\lambda x.f(x) =_{\xi(s)} \lambda x.g(x) : \Pi x : A.B(x)}}{\lambda x.g(x) =_{\sigma(\xi(s))} \lambda x.f(x) : \Pi x : A.B(x)} \triangleright_{smss} \frac{\dfrac{\dfrac{[x : A]}{f(x) =_s g(x) : B(x)}}{g(x) =_{\sigma(s)} f(x) : B(x)}}{\lambda x.g(x) =_{\xi(\sigma(s))} \lambda x.f(x) : \Pi x : A.B(x)}$$

Associated rewritings:
$\sigma(\xi(r)) \triangleright_{sx} \xi(\sigma(r))$
$\sigma(\xi(r,s)) \triangleright_{sxss} \xi(\sigma(r), \sigma(s))$
$\sigma(\xi(s) \triangleright_{smss} \xi(\sigma(s))$

Definition 1.21 (σ and μ [17])

$$\frac{\dfrac{x =_r y : A \times B}{FST(x) =_{\mu_1(r)} FST(y) : A}}{FST(y) =_{\sigma(\mu_1(r))} FST(x) : A} \triangleright_{sm} \frac{\dfrac{x =_r y : A \times B}{y =_{\sigma(r)} x : A \times B}}{FST(y) =_{\mu_1(\sigma(r))} FST(x) : A}$$

$$\frac{\dfrac{x =_r y : A \times B}{SND(x) =_{\mu_2(r)} SND(y) : A}}{SND(y) =_{\sigma(\mu_2(r))} SND(x) : A} \triangleright_{sm} \frac{\dfrac{x =_r y : A \times B}{y =_{\sigma(r)} x : A \times B}}{SND(y) =_{\mu_2(\sigma(r))} SND(x) : A}$$

$$\frac{\dfrac{x =_s y : A \quad f =_r g : A \to B}{APP(f, x) =_{\mu(s,r)} APP(g, y) : B}}{APP(g, y) =_{\sigma(\mu(s,r))} APP(f, x) : B}$$

$$\rhd_{smss}\ \frac{\dfrac{x =_s y : A}{y =_{\sigma(s)} x : A} \quad \dfrac{f =_r g : A \to B}{g =_{\sigma(r)} f : A \to B}}{APP(g,y) =_{\mu(\sigma(s),\sigma(r))} APP(f,x) : B}$$

$$\frac{\dfrac{\dfrac{x =_r y : A + B \quad \begin{matrix}[s:A]\\ \vdots \\ d(s) =_u f(s) : C\end{matrix} \quad \begin{matrix}[t:B]\\ \vdots \\ e(t) =_v g(t) : C\end{matrix}}{D(x, \acute{s}d(s), \acute{t}e(t)) =_{\mu(r,u,v)} D(y, \acute{s}f(s), \acute{t}g(t)) : C}}{D(y, \acute{s}f(s), \acute{t}g(t)) : C =_{\sigma(\mu(r,u,v))} D(x, \acute{s}d(s), \acute{t}e(t)) : C}$$

$$\rhd_{smsss}\ \frac{\dfrac{x =_r y : A + B}{y =_{\sigma(r)} x : A + B} \quad \dfrac{[s:A]}{d(s) =_u f(s) : C} \quad \dfrac{[t:B]}{e(t) =_v g(t) : C}}{f(s) =_{\sigma(u)} d(s) : C \quad g(t) =_{\sigma(v)} e(t) : C}}{D(y, \acute{s}f(s), \acute{t}g(t)) =_{\mu(\sigma(r),\sigma(u),\sigma(v))} D(x, \acute{s}d(s), \acute{t}e(t)) : C}$$

$$\frac{\dfrac{e =_s b : \Sigma x : A.B(x) \quad \begin{matrix}[t:A,\ g(t):B(t)]\\ d(g,t) =_r f(g,t) : C\end{matrix}}{E(e, \acute{g}\acute{t}d(g,t)) =_{\mu(s,r)} E(b, \acute{g}\acute{t}f(g,t)) : C}}{E(b, \acute{g}\acute{t}f(g,t)) =_{\sigma(\mu(s,r))} E(e, \acute{g}\acute{t}d(g,t)) : C}$$

$$\rhd_{smss}\ \frac{\dfrac{e =_s b : \Sigma x : A.B(x)}{b =_{\sigma(s)} e : \Sigma x : A.B(x)} \quad \dfrac{[t:A,\ g(t):B(t)]}{d(g,t) =_r f(g,t) : C}}{f(g,t) =_{\sigma(r)} d(g,t) : C}}{E(b, \acute{g}\acute{t}f(g,t)) =_{\mu(\sigma(s),\sigma(r))} E(e, \acute{g}\acute{t}d(g,t)) : C}$$

Associated rewritings:
$\sigma(\mu_1(r)) \rhd_{sm} \mu_1(\sigma(r))$
$\sigma(\mu_2(r)) \rhd_{sm} \mu_2(\sigma(r))$
$\sigma(\mu(s,r)) \rhd_{smss} \mu(\sigma(s), \sigma(r))$
$\sigma(\mu(r,u,v)) \rhd_{smsss} \mu(\sigma(r), \sigma(u), \sigma(v))$

Definition 1.22 (τ **and** sub **[17])**

$$\frac{\dfrac{x =_r \mathcal{C}[y] : A \quad y =_s w : A'}{x =_{\mathtt{sub}_{\mathtt{L}}(r,s)} \mathcal{C}[w] : A} \quad \mathcal{C}[w] =_t z : A}{x =_{\tau(\mathtt{sub}_{\mathtt{L}}(r,s),t)} z : A}$$

$$\rhd_{tsbll}\ \frac{x =_r \mathcal{C}[y] : A \quad \dfrac{y =_s w : A' \quad \mathcal{C}[w] =_t z : A}{\mathcal{C}[y] =_{\mathtt{sub}_{\mathtt{R}}(s,t)} z : A}}{x =_{\tau(r,\mathtt{sub}_{\mathtt{R}}(s,t))} z : A}$$

$$\frac{\dfrac{y =_s w : A \quad \mathcal{C}[w] =_t z : A}{\mathcal{C}[y] =_{\mathtt{sub}_{\mathtt{R}}(s,t)} z : A} \quad z =_u v : A}{\mathcal{C}[y] =_{\tau(\mathtt{sub}_{\mathtt{R}}(s,t),u)} v : A}$$

$$\rhd_{tsbrl}\ \frac{y =_s w : D' \qquad \dfrac{\mathcal{C}[w] =_t z : A \quad z =_u v : A}{\mathcal{C}[w] =_{\tau(t,u)} v : A}}{\mathcal{C}[y] =_{\mathtt{sub_R}(s,\tau(t,u))} v : A}$$

$$\frac{x =_r \mathcal{C}[z] : A \qquad \dfrac{\mathcal{C}[z] =_\rho \mathcal{C}[z] : A \quad z =_s w : A'}{\mathcal{C}[z] =_{\mathtt{sub_L}(\rho,s)} \mathcal{C}[w] : A}}{x =_{\tau(r,\mathtt{sub_L}(\rho,s))} \mathcal{C}[w] : A}$$

$$\rhd_{tsblr}\ \frac{x =_r \mathcal{C}[z] : A \quad z =_s w : A'}{x =_{\mathtt{sub_L}(r,s)} \mathcal{C}[w] : A}$$

$$\frac{x =_r \mathcal{C}[w] : A \qquad \dfrac{w =_s z : A' \quad \mathcal{C}[z] =_\rho \mathcal{C}[z] : A}{\mathcal{C}[w] =_{\mathtt{sub_R}(s,\rho)} \mathcal{C}[z] : A}}{x =_{\tau(r,\mathtt{sub_R}(s,\rho))} \mathcal{C}[z] : A}$$

$$\rhd_{tsbrr}\ \frac{x =_r \mathcal{C}[w] : D \quad w =_s z : A'}{x =_{\mathtt{sub_L}(r,s)} \mathcal{C}[z] : A}$$

Definition 1.23 (τ and τ [17])

$$\frac{\dfrac{x =_t y : A \quad y =_r w : A}{x =_{\tau(t,r)} w : A} \qquad w =_s z : A}{x =_{\tau(\tau(t,r),s)} z : A}$$

$$\rhd_{tt}\ \frac{x =_t y : A \qquad \dfrac{y =_r w : A \quad w =_s z : A}{y =_{\tau(r,s)} z : A}}{x =_{\tau(t,\tau(r,s))} z : A}$$

Associated rewritings:
$\tau(\mathtt{sub_L}(r,s),t) \rhd_{tsbll} \tau(r,\mathtt{sub_R}(s,t))$
$\tau(\mathtt{sub_R}(s,t),u)) \rhd_{tsbrl} \mathtt{sub_R}(s,\tau(t,u))$
$\tau(r,\mathtt{sub_L}(\tau,s)) \rhd_{tsblr} \mathtt{sub_L}(r,s)$
$\tau(r,\mathtt{sub_R}(s,\tau)) \rhd_{tsbrr} \mathtt{sub_L}(r,s)$
$\tau(\tau(t,r),s) \rhd_{tt} \tau(t,\tau(r,s))$

Thus, we put together all those rules to compose our rewrite system:

Definition 1.24 ($LND_{EQ} - TRS$ [17])
1. $\sigma(\rho) \rhd_{sr} \rho$
2. $\sigma(\sigma(r)) \rhd_{ss} r$
3. $\tau(\mathcal{C}[r],\mathcal{C}[\sigma(r)]) \rhd_{tr} \mathcal{C}[\rho]$
4. $\tau(\mathcal{C}[\sigma(r)],\mathcal{C}[r]) \rhd_{tsr} \mathcal{C}[\rho]$
5. $\tau(\mathcal{C}[r],\mathcal{C}[\rho]) \rhd_{trr} \mathcal{C}[r]$
6. $\tau(\mathcal{C}[\rho],\mathcal{C}[r]) \rhd_{tlr} \mathcal{C}[r]$
7. $\mathtt{sub_L}(\mathcal{C}[r],\mathcal{C}[\rho]) \rhd_{slr} \mathcal{C}[r]$

8. $\mathtt{sub}_{\mathtt{R}}(\mathcal{C}[\rho], \mathcal{C}[r]) \rhd_{srr} \mathcal{C}[r]$
9. $\mathtt{sub}_{\mathtt{L}}(\mathtt{sub}_{\mathtt{L}}(s, \mathcal{C}[r]), \mathcal{C}[\sigma(r)]) \rhd_{sls} s$
10. $\mathtt{sub}_{\mathtt{L}}(\mathtt{sub}_{\mathtt{L}}(s, \mathcal{C}[\sigma(r)]), \mathcal{C}[r]) \rhd_{slss} s$
11. $\mathtt{sub}_{\mathtt{R}}(\mathcal{C}[s], \mathtt{sub}_{\mathtt{R}}(\mathcal{C}[\sigma(s)], r)) \rhd_{srs} r$
12. $\mathtt{sub}_{\mathtt{R}}(\mathcal{C}[\sigma(s)], \mathtt{sub}_{\mathtt{R}}(\mathcal{C}[s], r)) \rhd_{srrr} r$
13. $\mu_1(\xi_1(r)) \rhd_{mx2l1} r$
14. $\mu_1(\xi_\wedge(r, s)) \rhd_{mx2l2} r$
15. $\mu_2(\xi_\wedge(r, s)) \rhd_{mx2r1} s$
16. $\mu_2(\xi_2(s)) \rhd_{mx2r2} s$
17. $\mu(\xi_1(r), s, u) \rhd_{mx3l} s$
18. $\mu(\xi_2(r), s, u) \rhd_{mx3r} u$
19. $\nu(\xi(r)) \rhd_{mxl} r$
20. $\mu(\xi_2(r), s) \rhd_{mxr} s$
21. $\xi(\mu_1(r), \mu_2(r)) \rhd_{mx} r$
22. $\mu(t, \xi_1(r), \xi_2(s)) \rhd_{mxx} t$
23. $\xi(\nu(r)) \rhd_{xmr} r$
24. $\mu(s, \xi_2(r)) \rhd_{mx1r} s$
25. $\sigma(\tau(r, s)) \rhd_{stss} \tau(\sigma(s), \sigma(r))$
26. $\sigma(\mathtt{sub}_{\mathtt{L}}(r, s)) \rhd_{ssbl} \mathtt{sub}_{\mathtt{R}}(\sigma(s), \sigma(r))$
27. $\sigma(\mathtt{sub}_{\mathtt{R}}(r, s)) \rhd_{ssbr} \mathtt{sub}_{\mathtt{L}}(\sigma(s), \sigma(r))$
28. $\sigma(\xi(r)) \rhd_{sx} \xi(\sigma(r))$
29. $\sigma(\xi(s, r)) \rhd_{sxss} \xi(\sigma(s), \sigma(r))$
30. $\sigma(\mu(r)) \rhd_{sm} \mu(\sigma(r))$
31. $\sigma(\mu(s, r)) \rhd_{smss} \mu(\sigma(s), \sigma(r))$
32. $\sigma(\mu(r, u, v)) \rhd_{smsss} \mu(\sigma(r), \sigma(u), \sigma(v))$
33. $\tau(r, \mathtt{sub}_{\mathtt{L}}(\rho, s)) \rhd_{tsbll} \mathtt{sub}_{\mathtt{L}}(r, s)$
34. $\tau(r, \mathtt{sub}_{\mathtt{R}}(s, \rho)) \rhd_{tsbrl} \mathtt{sub}_{\mathtt{L}}(r, s)$
35. $\tau(\mathtt{sub}_{\mathtt{L}}(r, s), t) \rhd_{tsblr} \tau(r, \mathtt{sub}_{\mathtt{R}}(s, t))$
36. $\tau(\mathtt{sub}_{\mathtt{R}}(s, t), u) \rhd_{tsbrr} \mathtt{sub}_{\mathtt{R}}(s, \tau(t, u))$
37. $\tau(\tau(t, r), s) \rhd_{tt} \tau(t, \tau(r, s))$
38. $\tau(\mathcal{C}[u], \tau(\mathcal{C}[\sigma(u)], v)) \rhd_{tts} v$
39. $\tau(\mathcal{C}[\sigma(u)], \tau(\mathcal{C}[u], v)) \rhd_{tst} u.$

1.8 Normalization

In the previous subsection, we have seen a system of rewrite rules that resolves reductions in a computational path. When we talk about these kind of systems, two questions arise: Every computational path has a normal form? And if a computational path has a normal form, is it unique? To show that it has a normal form, one has to prove that every computational path terminates, i.e., after a finite number of rewrites, one will end up with a path that does not have

any additional reduction. To show that it is unique, one needs to show that the system is confluent. In other words, if one has a path with 2 or more reductions, one needs to show that the choice of the rewrite rule does not matter. In the end, one will always obtain the same end-path without any redundancies.

1.8.1 Termination

We are interested in the following theorem [23, 17]:

Theorem 1.25 (Termination property for $LND_{EQ} - TRS$) *LND_{EQ}−TRS is terminating.*

The proofs uses a special kind of ordering, known as *recursive parth ordering*, proposed by [3]:

Definition 1.26 (Recursive path ordering [3, 17]) *Let $>$ be a partial ordering on a set of operators F. The recursive path ordering $>^*$ on the set T(F) of terms over F is defined recursively as follows:*

$$s = f(s_1, \ldots, s_m) >^* g(t_1, \ldots, t_n) = t,$$

if and only if

1. *$f = g$ and $\{s_1, \ldots, s_m\} \gg^* \{t_1, \ldots, t_n\}$, or*
2. *$f > g$ and $\{s\} \gg^* \{t_1, \ldots, t_n\}$, or*
3. *$f \not\geq g$ and $\{s_1, \ldots, s_m\} \gg^*$ or $= \{t\}$*

where $\gg^*$ is the extension of $>^*$ to multisets.

This definition uses the notion of partial ordering in multisets. A given partial ordering $>$ on a set S may be extended to a partial ordering $\gg$ on finite multisets of elements of S, wherein a multiset is reduced by removing one or more elements and replacing them with any finite number of elements, each one which is smaller than one of the elements removed [3].

Thus, one can proof the termination property by showing that all rules $e \to d$ of the system, one has that $e >^* d$.We also need to define the precedence

ordering on the rewrite operators. We define as follows [17, 23]:

$$\begin{aligned}
&\sigma > \tau > \rho,\\
&\sigma > \xi,\\
&\sigma > \xi_{\wedge},\\
&\sigma > \xi_1,\\
&\sigma > \xi_2,\\
&\sigma > \mu,\\
&\sigma > \mu_1,\\
&\sigma > \mu_2,\\
&\sigma > \mathtt{sub}_{\mathtt{L}},\\
&\sigma > \mathtt{sub}_{\mathtt{R}},\\
&\tau > \mathtt{sub}_{\mathtt{L}}
\end{aligned}$$

Thus, one can prove the termination by showing that for every rule of $e \to d$ of $LND_{EQ} - TRS$, $e >^* d$. For almost every rule it is a straightforward and tedious process. We are not going to show all those steps in this work, but we can give the proof of two examples.

26. $\sigma(sub_L(r,s)) >^* sub_R(\sigma(s),\sigma(r))$:

    - $\sigma > sub_R$ from the precedence ordering on the rewrite operators.
    - $\{\sigma(sub_L(r,s))\} \gg^* \{\sigma(r),\sigma(r)\}$:
        - $\sigma(sub_L(r,s)) >^* \sigma(s)$ e $\sigma(sub_L(r,s)) >^* \sigma(r)$:
            - $\sigma = \sigma$
            - $\{sub(r,s)\} \gg \{s\}$ from the subterm condiction.
            - $\{sub(r,s)\} \gg \{r\}$ from the subterm condiction.

27. $\sigma(sub_R(r,s)) \rhd sub_L(\sigma(s),\sigma(r))$:

    - $\sigma > sub_L$ from the precedence ordering on the rewrite operators.
    - $\{\sigma(sub_R(r,s))\} \gg^* \{\sigma(r),\sigma(r)\}$:
        - $\sigma = \sigma$
        - $\{sub_R(r,s)\} \gg \{s\}$ from the subterm condiction.
        - $\{sub_R(r,s)\} \gg \{r\}$ from the subterm condiction.

All others proof can check the full proof in [23].

1.8.2 Confluence

Before we go to the proof of confluence, one needs to notice that $LND_{EQ}-TRS$ is a conditional term rewriting system. This means that some rules can only be applied if the terms of the associated equation follow some rules. For example, for the rule $\mu_1(\xi_\wedge(r,s)) \triangleright_{mx2l2} r$, it is necessary to have an β-Reduction like $FST\langle x,y\rangle$. With that in mind, one has the following definition [23]:

Definition 1.27 (Conditional term rewriting system) *In conditional term rewriting systems, the rules have conditions attached, which must be true for the rewrite occur. For example, a rewrite rule $e \to d$ with condition C is expressed as:*

$$C|e \to d$$

To prove the confluence, one should analyze all possible critical pairs using the superposition algorithm proposed by [7]. Thus, there should not be any divergent critical pair. For example, one can take the superposition of rules 1 and 2, obtaining: $\sigma(\sigma(\rho))$. We have two possible rewrites [23]:

- $\sigma(\sigma(\rho)) \triangleright_{sr} \sigma(\rho) \triangleright_{sr} \rho$
- $\sigma(\sigma(\rho)) \triangleright_{ss} \rho$.

As one can see, we ended up with the same term ρ. Thus, no divergence has been generated.

One should compare every pair of rules to find all critical pairs and see if there are divergences. If some divergence happens, the superposition algorithm proposed by [7] shows how to add new rules to the system in such a way that it becomes confluent. As a matter of fact, that was the reason why the rules 38 and 39 of $LND_{EQ}-TRS$ have been introduced to the system [17]:

38. $\tau(\mathcal{C}[u], \tau(\mathcal{C}[\sigma(u)], v)) \triangleright_{tts} v$
39. $\tau(\mathcal{C}[\sigma(u)], \tau(\mathcal{C}[u], v)) \triangleright_{tst} u$.

Those two rules introduced the following reductions to the system [23]:

$$\dfrac{x =_s u : D \qquad \dfrac{\dfrac{x =_s u : D}{u =_{\sigma(s)} x : D} \qquad x =_v w : D}{u =_{\tau(\sigma(s),v)} w : D}}{x =_{\tau(s,\tau(\sigma(s),v))} w : D} \quad \triangleright_{tts} \quad x =_v w$$

$$\dfrac{\dfrac{x =_s w : D}{w =_{\sigma(s)} x : D} \qquad \dfrac{x =_s w : D \qquad w =_v z : D}{x =_{\tau(s,v)} z : D}}{w =_{\tau(\sigma(s),\tau(s,v))} z : D} \rhd_{ss} \quad w =_v z$$

One can check a full proof of confluence in [13, 18, 15, 23].

1.8.3 Normalization Procedure

We can now state two normalization theorems:

Theorem 1.28 (normalization [23]) *Every derivation in the $LND_{EQ}-TRS$ converts to a normal form.*

Proof 1.29 *Direct consequence of the termination property.*

Theorem 1.30 (strong normalization [23]) *Every derivation in the $LND_{EQ}-TRS$ converts to a unique normal form.*

Proof 1.31 *Direct consequence of the termination and confluence properties.*

In this sense, every proof can be reduced to a normal one. To do that, one should identify the redundancies and, based on the rewrite rules, one can construct a proof without any redundancies. We show that in an example. It is the following [23]:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{f(x,z) =_s f(w,y) : D}{f(w,y) =_{\sigma(s)} f(x,z) : D} \qquad x =_r c : D}{f(w,y) =_{sub_L(\sigma(s),r)} f(c,z) : D}}{f(c,z) =_{\sigma(sub_L(\sigma(s),r))} f(w,y) : D} \qquad y =_t b : D}{f(c,z) =_{sub_L(\sigma(sub_L(\sigma(s),r)))} f(w,b) : D}$$

This deduction generates the following path: $sub_L(\sigma(sub_L(\sigma(s), r)))$. This path is not in normal form, having two redundancies [23]:

$$sub_L(\sigma(sub_L(\sigma(s), r))) \rhd_{ssbl} sub_L(sub_R(\sigma(r), \sigma(\sigma(s)), t)$$
$$sub_L(sub_R(\sigma(r), \sigma(\sigma(s)), t) \rhd_{ss} sub_L(sub_R(\sigma(r), s), t)$$

Thus, one can identify those reductions and conceive a deduction without any redundancies [23]:

$$\dfrac{\dfrac{\dfrac{x =_r c : D}{c =_{\sigma(r)} x : D} \qquad f(x,z) =_s f(w,y) : D}{f(c,z) =_{sub_R(\sigma(r),s)} f(w,y) : D} \qquad y =_t b : D}{f(c,z) =_{sub_L(sub_R(\sigma(r),s),t)} f(w,b) : D}$$

It is important to emphasize that although each computational path has a unique normal form, there may be two computational paths "r" and "s" between elements "a" and "b" which have different normal forms.

1.9 Rewrite Equality

As we have just seen, the $LND_{EQ} - TRS$ has 39 rewrite rules. We call each rule as a *rewrite rule* (abbreviation: *rw-rule*). We have the following definition:

Definition 1.32 (Rewrite Rule [24]) *An rw-rule is any of the rules defined in $LND_{EQ} - TRS$.*

Similarly to the β-reduction of λ-calculus, we have a definition for rewrite reduction:

Definition 1.33 (Rewrite reduction [24]) *Let s and t be computational paths. We say that $s \rhd_{1rw} t$ (read as: s rw-contracts to t) iff we can obtain t from s by an application of only one rw-rule. If s can be reduced to t by finite number of rw-contractions, then we say that $s \rhd_{rw} t$ (read as s rw-reduces to t).*

We also have rewrite contractions and equality:

Definition 1.34 (Rewrite contraction and equality [24]) *Let s and t be computational paths. We say that $s =_{rw} t$ (read as: s is rw-equal to t) iff t can be obtained from s by a finite (perhaps empty) series of rw-contractions and reversed rw-contractions. In other words, $s =_{rw} t$ iff there exists a sequence $R_0, ..., R_n$, with $n \geq 0$, such that*

$$(\forall i \leq n-1)(R_i \rhd_{1rw} R_{i+1} \text{ or } R_{i+1} \rhd_{1rw} R_i)$$
$$R_0 \equiv s, \quad R_n \equiv t$$

A fundamental result is the fact that rewrite equality is an equivalence relation [24]:

Proposition 1.35 *Rewrite equality is transitive, symmetric and reflexive.*

Proof 1.36 *Comes directly from the fact that rw-equality is the transitive, reflexive and symmetric closure of rw.*

Rewrite reduction and equality play fundamental roles in the groupoid model of a type based on computational paths, as we are going to see in the sequel.

1.10 LNDEQ-TRS(2)

Until now, this subsection has concluded that there exist redundancies which are resolved by a system called $LND_{EQ} - TRS$. This system establishes rules that reduces these redundancies. Moreover, we concluded that these redundancies are just redundant uses of the equality axioms showed in *section 2*. In fact, since these axioms just define an equality theory for type theory, one can specify and say that these are redundancies of the equality of type theory. As we mentioned, the $LND_{EQ} - TRS$ has a total of 39 rules[13, 17]. Since the rw-equality is based on the rules of $LND_{EQ} - TRS$, one can just imagine the high number of redundancies that rw-equality could cause. In fact, a thoroughly study of all the redundancies caused by these rules led to the work done in [25], that only interested in the redundancies caused by the fact that rw-equality is transitive, reflexive and symmetric with the addition of only one specific rw_2-rule. This way up, was created a system, called $LND_{EQ} - TRS_2$, that resolves all the redundancies caused by rw-equality (the same way that $LND_{EQ} - TRS$ resolves all the redundancies caused by equality). Since we know that rw-equality is transitive, symmetric and reflexive, it should have the same redundancies that the equality had involving only these properties. Since rw-equality is just a sequence of rw-rules (also similar to equality, since equality is just a computational path, i.e., a sequence of identifiers), then we could put a name on these sequences. For example, if s and t are rw-equal because there exists a sequence $\theta : R_0, ..., R_n$ that justifies the rw-equality, then we can write that $s =_{rw_\theta} t$. Thus, we can rewrite, using rw-equality, all the rules that originated the rules involving τ, σ and ρ. For example, we have [24]:

$$\dfrac{\dfrac{x =_{rw_t} y : A \qquad y =_{rw_r} w : A}{x =_{rw_{\tau(t,r)}} w : A} \qquad w =_{rw_s} z : A}{x =_{rw_{\tau(\tau(t,r),s)}} z : A}$$

$$\rhd_{tt_2} \dfrac{x =_{rw_t} y : A \qquad \dfrac{y =_{rw_r} w : A \qquad w =_{rw_s} z : A}{y =_{rw_{\tau(r,s)}} z : A}}{x =_{rw_{\tau(t,\tau(r,s))}} z : A}$$

Therefore, we obtain the rule tt_2, that resolves one of the redundancies caused by the transitivity of rw-equality (the 2 in tt_2 indicates that it is a rule that resolves a redundancy of rw-equality). In fact, using the same reasoning, we can obtain, for rw-equality, all the redundancies that we have shown in **definition 1.24**. In other words, we have tr_2, tsr_2, trr_2, tlr_2, sr_2, ss_2 and tt_2. Since we have now rules of $LND_{EQ} - TRS_2$, we can use all the concepts that we have just defined for $LND_{EQ} - TRS$. The only difference is that instead of having rw-rules and rw-equality, we have rw_2-rules and rw_2-equality.

There is an important rule specific to this system. It stems from the fact that transitivity of reducible paths can be reduced in different ways, but generating the same result. For example, consider the simple case of $\tau(s,t)$ and consider that it is possible to reduce s to s' and t to t'. There is two possible rw-sequences that reduces this case: The first one is $\theta : \tau(s,t) \rhd_{1rw} \tau(s',t) \rhd_{1rw} \tau(s',t')$ and the second $\theta' : \tau(s,t) \rhd_{1rw} \tau(s,t') \rhd_{1rw} \tau(s',t')$. Both rw-sequences obtained the same result in similar ways, the only difference being the choices that have been made at each step. Since the variables, when considered individually, followed the same reductions, these rw-sequences should be considered redundant relative to each other and, for that reason, there should be rw_2-rule that establishes this reduction. This rule is called *independence of choice* and is denoted by cd_2. Since we already understand the necessity of such a rule, we can define it formally:

Definition 1.37 (Independence of choice [24]) *Let θ and ϕ be rw-equalities expressed by two rw-sequences: $\theta : \theta_1, ..., \theta_n$, with $n \geq 1$, and $\phi : \phi_1, ..., \phi_m$, with $m \geq 1$. Let T be the set of all possible rw-equalities from $\tau(\theta_1, \phi_1)$ to $\tau(\theta_n, \theta_m)$ described by the following process: $t \in T$ is of the form $\tau(\theta_{l_1}, \phi_{r_1}) \rhd_{1rw} \tau(\theta_{l_2}, \phi_{r_2}) \rhd_{1rw} ... \rhd_{1rw} \tau(\theta_{l_x}, \phi_{r_y})$, with $l_1 = 1, r_1 = 1$, $l_x = n, r_y = m$ and $l_{i+1} = 1 + l_i$ and $r_{i+1} = r_i$ or $l_{i+1} = l_i$ and $r_{i+1} = 1 + r_i$. The independence of choice, denoted by cd_2, is defined as the rule of $LND_{EQ} - TRS_2$ that establishes the equality between any two different terms of T. In other words, if $x, y \in T$ and $x \neq y$, then $x =_{cd_2} y$ and $y =_{cd_2} x$.*

Analogously to the rw-equality, rw_2-equality is also an equivalence relation [24]:

Proposition 1.38 *rw_2-equality is transitive, symmetric and reflexive.*

Proof 1.39 *Analogous to Proposition 1.35.*

2 A Topological Application of Labelled Natural Deduction.

Once we have built up all the necessary basis of computational paths to develop our work, it would be interesting to consult two proofs of the calculation of the fundamental group of the circle: The first is the mathematically proven proof that appears in the book of algebraic topology [12] in chapter 9, section 54. The second is a proof using homotopic type theory, which is in the book of [28] in chapter 8. Both cases have the proofs of the fundamental group of the circle, but to obtain such a success the amount of information needed is much higher and much more complex than we will propose in the sequel.

In homotopy theory, the fundamental group is the one formed by all equivalence classes up to homotopy of paths (loops) starting from a point x_0 and also ending at x_0. Since we use computational paths as the syntactic counterpart of homotopic paths in type theory, we use computational paths to propose the following definition:

Definition 2.1 *Let*

(i) A *be a type.*

(ii) $x_0 : A$ *a base point.*

(iii) $x_0 \underset{\alpha_i}{=} x_0$, *be a family of generator paths with* $i \in I$.

(iv) A family of relationships between the terms paths $\tau_j(x_0 \underset{\alpha_r}{=} x_0, x_0 \underset{\alpha_s}{=} x_0)$.

We can define the structure $\Pi_1(A, x_0)$ *as the set of terms* α_{x_0}, *given by finite applications of* τ, σ, *and* ρ *in* α_i, *modulo rw equality and modulo the family of identity type terms* Id_{τ_j}.

Since each element in $\Pi_1(A, x_0)$ is a loop in x_0, we will give a important definition indispensable to our work:

Definition 2.2 *We can define and denote by*

$$[loop^n]_{rw}$$

the path naturally obtained by the application the of path-axioms ρ, τ *and* σ *to the base path* $x_0 \underset{loop}{=} x_0$, *where* $n \in \mathbb{N}$. *Particularly we can say:*

(i) $[loop^0]_{rw} = [\rho_{x_0}]_{rw}$, $n = 0$.

(ii) $[loop^n]_{rw} = \tau\big([loop^{n-1}]_{rw}, [loop^1]_{rw}\big)$, $n > 0$.

(iii) $[loop^n]_{rw} = \sigma([loop^{-n}]_{rw})$, $-n > 0$.

For example, we have:

a) $\tau([loop^1]_{rw}, [loop^1]_{rw}) = [loop^2]_{rw}$

a) $\tau\Big(\sigma([loop^1]_{rw}), \sigma([loop^1]_{rw})\Big) = \sigma([loop^2]_{rw}) = [loop^{-2}]_{rw}$

c) $\tau\Big(\sigma([loop^1]_{rw}), [loop^1]_{rw}\Big) \underset{tsr}{=} [\rho]_{rw}$.

Here we need to show relevant information regarding the equalities we can get using these paths. Consider the following examples:

(p_1)

$$\begin{aligned}\tau\Big(\tau\big([loop^1]_{rw}, [loop^1]_{rw}\big), \sigma\big([loop^1]_{rw}\big)\Big) &\underset{tt}{=} \tau\Big([loop^1]_{rw}, \tau\Big([loop^1]_{rw}, \sigma\big([loop^1]_{rw}\big)\Big)\Big)\\ &\underset{tr}{=} \tau\big([loop^1]_{rw}, [\rho]_{rw}\big)\\ &\underset{trr}{=} [loop^1]_{rw}\end{aligned}$$

(p_2)

$$\begin{aligned}\tau\Big(\tau\Big([loop^1]_{rw}, \sigma\big([loop^1]_{rw}\big)\Big), [loop^1]_{rw}\Big) &\underset{tr}{=} \tau\big([\rho]_{rw}, [loop^1]_{rw}\big)\\ &\underset{tlr}{=} [loop^1]_{rw}\end{aligned}$$

Notice that the paths (p_1) and (p_2) initially look like distinct paths. Nevertheless, applying only the properties of computational paths, together with the rewrite rules (*rw-rules*), we end up with the path $[loop^1]_{rw}$ in both derivations. So we can say that:
By (p_1),

$$\tau\Big(\tau\big([loop^1]_{rw}, [loop^1]_{rw}\big), \sigma\big([loop^1]_{rw}\big)\Big) \underset{trr}{=} [loop^1]_{rw}$$

and by (p_2),

$$\tau\Big(\tau\Big([loop^1]_{rw}, \sigma\big([loop^1]_{rw}\big)\Big), [loop^1]_{rw}\Big) \underset{tlr}{=} [loop^1]_{rw}.$$

They are said to be *rw-equal* to the base path $[loop^1]_{rw}$ because they can be rewritten to $[loop^1]_{rw}$ after applying *rw-rules*. Therefore, we can say that these paths are in the same equivalence class as $[loop^1]_{rw}$ and thus, they are equal up to *rw*-equality.

2.1 Fundamental Group of Circle

Definition 2.3 (The circle S^1) *The circle is the type generated by:*

(i) *A base point* - $x_0 : S^1$

(ii) *A base computational path* - $x_0 \underset{loop}{=} x_0 : S^1$.

The first thing one should notice is that this definition does not use only the points of the type S^1, but also a base computational path called *loop* between those points. That is why it is called a higher inductive type [28]. Our approach differs from the one developed in the HTT book [28] on the fact that we do not need to simulate the path-space between those points, since we add computational paths to the syntax of the theory.

In Martin-Löf's type theory, the existence of those additional paths comes from establishing that the paths should be freely generated by the constructors [28]. In our approach, we do not have to appeal to this kind of argument, since all paths come naturally from direct applications of the axioms and the inference rules which define the theory of equality. We proceed with the following definition:

Definition 2.4 *In S^1, we define the following canonical loops (canonical paths):*

(i) $[loop^0]_{rw} = [\rho_{x_0}]_{rw}$, $n = 0$

(ii) $[loop^n]_{rw} = \sigma([loop^{-n}]_{rw})$, $n < 0$.

(iii) $[loop^n]_{rw} = \tau\big([loop^{n-1}]_{rw}, [loop^1]_{rw}\big)$, $n > 0$.

Lemma 2.5 *All paths in S^1 are rw-equals to a path $[loop^n]_{rw}$, for $n \in \mathbb{N}$.*

Proof 2.6 *Let φ be a computational path in S^1.*

I. *If $\varphi = \rho$:*

(i) $\varphi = [loop^0]_{rw}$, $n = 0$.

(ii) $\varphi = \sigma([loop^n]_{rw}) = \sigma(\sigma([loop^{-n}]_{rw})) \underset{ss}{=} [loop^{-n}]_{rw} = \rho$, $n = 0$.

(iii) $\varphi = \tau\big([loop^m]_{rw}, [loop^n]_{rw}\big) = \rho$, *if* $m + n = 0$. *Therefore,*

$$\begin{aligned} \varphi = \tau\big([loop^m]_{rw}, [loop^n]_{rw}\big) \quad &= \tau\big([loop^{-n}]_{rw}, [loop^n]_{rw}\big) \\ &= \tau\big([loop^{-n}]_{rw}, \sigma([loop^{-n}]_{rw})\big) \\ &\underset{tr}{=} \rho. \end{aligned}$$

II. *If $\varphi = \sigma([loop^n])$:*

(i) *For $n = 0$ we have $\varphi = \sigma([loop^0]_{rw}) = \rho$.*

(ii) Suppose true for $n = k$ that every paths in S^1 are rw-equals to a path $[loop^n]_{rw}$. For $n = k + 1$ we have:

$$\begin{aligned}\varphi \quad &= \sigma([loop^{k+1}]_{rw}) \\ &= \sigma\Big(\tau\big([loop^k]_{rw}, [loop^1]_{rw}\big)\Big) \\ &\underset{stss}{=} \tau\big(\sigma([loop^k]_{rw}), \sigma([loop^1]_{rw})\big) \\ &= \tau\big([loop^{-k}]_{rw}, [loop^{-1}]_{rw}\big) \\ &= [loop^{-k-1}]_{rw} \\ &= [loop^{-(k+1)}]_{rw}.\end{aligned}$$

III. *If $\varphi = \tau\big([loop^{n-1}]_{rw}, [loop^1]_{rw}\big)$:*

(i) For $n = 0$, we have:

$$\varphi = \tau([loop^{-1}]_{rw}, [loop^1]_{rw}) = \tau(\sigma([loop^1]_{rw}), [loop^1]_{rw}) \underset{tsr}{=} \rho = [loop^0]_{rw}.$$

(ii) Suppose true for $n = k$, to $n = k + 1$ we have:

$$\begin{aligned}\varphi \quad &= \tau\big([loop^{k+1-1}]_{rw}, [loop^1]_{rw}\big) \\ &= \tau\Big([loop^k]_{rw}, \tau([loop^1]_{rw})\Big) \\ &\overset{hip}{=} \tau\Big(\tau([loop^{k-1}]_{rw}, [loop^1]), [loop^1]_{rw}\Big) \\ &\underset{rw}{=} [loop^1]_{rw} \circ [loop^k]_{rw} \\ &= [loop^{k+1}]_{rw}.\end{aligned}$$

All paths in S^1 are rw-equals to a path $[loop^n]_{rw}$, for $n \in \mathbb{N}$.

Lemma 2.7 *All paths in S^1 are generated by application ρ,τ and σ in base path $[loop^1]_{rw}$.*

Proof 2.8 *For the base case $[\rho]_{rw}$, it is trivially true, since we define it to be equal to $[loop^0]_{rw}$. From $[\rho]_{rw}$, one can construct more complex paths by composing with $[loop^1]_{rw}$ or $\sigma([loop^1]_{rw})$ on each step. Concatenating paths we have:*

(i) A path of the form $[\rho]_{rw}$ concatenated with $[loop^1]_{rw}$:

$$[\rho]_{rw} \circ [loop^1]_{rw} = \tau([loop^1]_{rw}, [\rho]_{rw}) \underset{trr}{=} [loop^1]_{rw}.$$

(ii) A path of the form $[\rho]_{rw}$ concatenated with $\sigma([loop^1]_{rw})$:

$$[\rho]_{rw} \circ \sigma([loop^1]_{rw}) = \tau(\sigma([loop^1]_{rw}), [\rho]_{rw}) \underset{trr}{=} \sigma([loop^1]_{rw}) = [loop^{-1}]_{rw}.$$

(iii) A path of the form $[loop^n]_{rw}$ *concatenated with* $[loop^1]_{rw}$*:*

$$[loop^n]_{rw} \circ [loop^1]_{rw} = \tau([loop^1]_{rw}, [loop^n]_{rw}) = [loop^{n+1}]_{rw}.$$

(iv) A path of the form $[loop^n]_{rw}$ *concatenated with* $\sigma([loop^1]_{rw})$*:*

$$\begin{aligned}
[loop^n]_{rw} \circ \sigma([loop^1]_{rw}) &= \tau\big(\sigma([loop^1]_{rw}), [loop^n]_{rw}\big) \\
&= \tau\Big(\sigma([loop^1]_{rw}), \tau\Big([loop^1]_{rw}, [loop^{n-1}]_{rw}\Big)\Big) \\
&\underset{\sigma(tt)}{=} \tau\Big(\tau\Big(\sigma([loop^1]_{rw}), [loop^1]_{rw}\Big), [loop^{n-1}]_{rw}\Big) \\
&\underset{tsr}{=} \tau\Big([\rho]_{rw}, [loop^{n-1}]_{rw}\Big) \\
&\underset{tlr}{=} [loop^{n-1}]_{rw}.
\end{aligned}$$

(v) A path of the form $[loop^{-n}]_{rw}$ *concatenated with* $[loop^1]_{rw}$*:*

$$\begin{aligned}
[loop^{-n}]_{rw} \circ [loop^1]_{rw} &= \tau\big([loop^1]_{rw}, [loop^{-n}]_{rw}\big) \\
&= \tau\Big([loop^1]_{rw}, \tau\Big(\sigma([loop^1]_{rw}), [loop^{-(n-1)}]_{rw}\Big)\Big) \\
&\underset{\sigma(tt)}{=} \tau\Big(\tau\Big([loop^1]_{rw}, \sigma([loop^1]_{rw})\Big), [loop^{-(n-1)}]_{rw}\Big) \\
&\underset{tr}{=} \tau\Big([\rho]_{rw}, [loop^{-(n-1)}]_{rw}\Big) \\
&\underset{tlr}{=} [loop^{-(n-1)}]_{rw}.
\end{aligned}$$

(vi) a path of the form $[loop^{-n}]_{rw}$ *concatenated with* $\sigma([loop^1]_{rw})$*:*

$$\begin{aligned}
[loop^{-n}]_{rw} \circ \sigma([loop^1]_{rw}) &= \tau\big(\sigma([loop^1]_{rw}), [loop^{-n}]_{rw}\big) \\
&= \tau\Big(\sigma([loop^1]_{rw}), \tau\Big([loop^1]_{rw}, [loop^{-(n+1)}]_{rw}\Big)\Big) \\
&\underset{\sigma(tt)}{=} \tau\Big(\tau\Big(\sigma([loop^1]_{rw}), [loop^1]_{rw}\Big), [loop^{-(n+1)}]_{rw}\Big) \\
&\underset{tsr}{=} \tau\Big([\rho]_{rw}, [loop^{-(n+1)}]_{rw}\Big) \\
&\underset{tlr}{=} [loop^{-(n+1)}]_{rw}.
\end{aligned}$$

For simplicity, we will denote by $x_0 \underset{r}{=} x_0$ whenever we refer to a computational path r is generated by ρ, σ and τ.

Proposition 2.9 $\left(\Pi_1(S^1, x_0), \circ\right)$ *is a group.*

Proof 2.10 *The first thing to define is the group operation* $\circ$. *Given any* $x_0 \underset{r}{=} x_0 : S^1$ *and* $x_0 \underset{t}{=} x_0 : S^1$, *we define* $r \circ s$ *as* $\tau(s, r)$. *Thus, we now need to check the group conditions:*

(i) ***Closure:*** *Given* $x_0 \underset{r}{=} x_0 : S^1$ *and* $x_0 \underset{t}{=} x_0 : S^1$, $r \circ s$ *must be a member of the group. Indeed,* $r \circ s = \tau(s, r)$ *is a computational path* $x_0 \underset{\tau(s,r)}{=} x_0 : S^1$.

(ii) ***Inverse:*** *Every member of the group must have an inverse. Indeed, if we have a path* r, *we can apply* $\sigma(r)$. *We claim that* $\sigma(r)$ *is the inverse of* r, *since we have:*

$$\sigma(r) \circ r = \tau(r, \sigma(r)) \underset{tr}{=} \rho$$
$$r \circ \sigma(r) = \tau(\sigma(r), r) \underset{tsr}{=} \rho$$

Since we are working up to rw-equality, the equalities hold strictly.

(iii) ***Identity:*** *We use the path* $x_0 \underset{\rho}{=} x_0 : S^1$ *as the identity. Indeed, we have:*

$$r \circ \rho = \tau(\rho, r) \underset{tlr}{=} r$$
$$\rho \circ r = \tau(r, \rho) \underset{trr}{=} r.$$

(iv) ***Associativity:*** *Given any members of the group* $x_0 \underset{r}{=} x_0 : S^1$, $x_0 \underset{t}{=} x_0$ *and* $x_0 \underset{s}{=} x_0$, *we want that* $r \circ (s \circ t) = (r \circ s) \circ t$:

$$r \circ (s \circ t) = \tau(\tau(t, s), r) \underset{tt}{=} \tau(t, \tau(s, r)) = (r \circ s) \circ t$$

All conditions have been satisfied. $\left(\Pi_1(S^1, x_0), \circ\right)$ *is a group.*

Thus, $\left(\Pi_1(S^1, x_0), \circ\right)$ is indeed a group. We call this group the fundamental group of S^1.

In [28] the next theorem was proof defining a pair of encode and decode functions, there it was necessary simulate a path-space and, by end, the work was very laborious. Nevertheless, since our computational paths are already part of the syntax, one does not need to rely on this kind of approach to simulate a path-space. In [12] the proof of this theorem is quite laborious. Work directly with the concept of computational paths, we hope that these same accounts can be performed more simple and affordably.

Theorem 2.11 $\Pi_1(S, x_0) \simeq \mathbb{Z}$

Proof 2.12 *Consider the application defined and denoted by:*

$$\begin{aligned} toPath: \quad & \mathbb{Z} \to \Pi_1(S) \\ & z \to toPath(z) = [loop^z]_{rw}. \end{aligned}$$

(i) toPath is a homomorphism.
Let $z = n + m \in \mathbb{Z}$, then:

$$\begin{aligned} toPath(z) &= toPath(n+m) \\ &= [loop^{n+m}]_{rw} \\ &= \tau([loop^n]_{rw}, [loop^m]_{rw}) \\ &= toPath(m) \circ toPath(n). \end{aligned}$$

By the other hand, how $z = m + n$ we have:

$$\begin{aligned} toPath(z) &= toPath(m+n) \\ &= [loop^{m+n}]_{rw} \\ &= \tau([loop^m]_{rw}, [loop^n]_{rw}) \\ &= toPath(n) \circ toPath(m). \end{aligned}$$

Thus, $toPath(n+m) = toPath(n) \circ toPath(m)$.

(ii) toPath is surjective.
*By **Lemma 2.5**, as every path in S^1 is rw-equal to one path $[loop^i]_{rw}$, we have that for all for all path $[loop^i]_{rw} \in \Pi_1(S^1), \exists i \in \mathbb{Z}$, such that, $toPath(i) = [loop^i]_{rw}$.*

(iii) $Ker(toPath) = \{0\}$.
Suppose there is $z \neq 0 \in \mathbb{Z}$, such that $z \in Ker(toPath)$. Thus,

$$toPath(z) = toPath(z+0) \overset{hom}{=} toPath(z) \circ toPath(0) = \tau(\rho, [loop^z]_{rw}) \overset{Ker}{=} \rho.$$

If $\tau(\rho, \alpha) = \rho$, by rw-rule $\underset{tr}{\triangleright}$ we have, $\alpha = \sigma(\rho) \Rightarrow z = 0 \to\leftarrow$. Therefore,

$$Ker(toPath) = \{0\}.$$

As ToPath is a homomorphism surjective with $Ker(toPath) = \{0\}$, then toPath is a isomorphism, that is, $\Pi_1(S, x_0) \simeq \mathbb{Z}$.

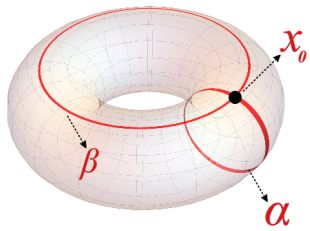


Figure 1: Paths α and β with base point x_0 in Torus

2.2 Fundamental Group of the Torus

Consider $\mathbb{T}^2$ as the surface known as Torus and the point $x_0 \in \mathbb{T}^2$. We will prove using computational paths that the fundamental group of the torus is isomorphic to $\mathbb{Z} \times \mathbb{Z}$. Here we will also use in **definition 2.1** with some simple adaptations. We will continue to work with paths up to *rw*-equality.

Since the fundamental groups are obtained by analysing the loops, we will be interested in working with *loops* that cannot be homotopic to base point x_0, like *loops* α and β. These loops will be the generators of $\mathbb{T}^2$, as shown in **figure1**, so we can give them a special definition for both.

Definition 2.13 (vertical loop) *We define and denote by*

$$\alpha^n = [loop_v^n]_{rw}$$

the path that passes through the inner part of $\mathbb{T}^2$ *in the vertical direction, naturally obtained by applications the of path-axioms* ρ, τ *and* σ *to the base path* $x_0 \underset{\alpha}{=} x_0$, *where* $n \in \mathbb{Z}$. *Particularly, we have*

(i) $[loop_v^0]_{rw} = [\rho]_{rw} = \alpha^0$, $n = 0$.

(ii) $[loop_v^{n+1}]_{rw} = \tau\big([loop_v^n]_{rw}, [loop_v^1]_{rw}\big) = \alpha^n$, $n > 0$.

(iii) $[loop_v^n]_{rw} = \sigma([loop_v^{-n}]_{rw}) = \alpha^{-n}$, $n < 0$.

In **figure 1**, this vertical path (loop) has the same orientation of the path denoted by α.

Definition 2.14 (horizontal loop) *We define and denote by*

$$\beta^m = [loop_h^m]_{rw}$$

the path that passes the inner part of $\mathbb{T}^2$ *in the horizontal direction, naturally obtained by applications the of path-axioms* ρ, τ *and* σ *to the base path* $x_0 \underset{\beta}{=} x_0$, *where* $n \in \mathbb{Z}$. *Particularly, we have:*

(i) $[loop_h^0]_{rw} = [\rho]_{rw} = \beta^0$, $m = 0$.

(ii) $[loop_h^{m+1}]_{rw} = \tau\big([loop_h^m]_{rw}, [loop_h^1]_{rw}\big) = \beta^m$, $m > 0$.

(iii) $[loop_h^m]_{rw} = \sigma([loop_h^{-m}]_{rw}) = \beta^{-m}$, $m < 0$.

In **figure 1**, this horizontal path (loop) has the same orientation of the path denoted by β. By ***definitions 2.13 and 2.14***, We can also represent the path homotopic to the constant one by: $[\rho]_{rw} = \alpha^0\beta^0$, or $[\rho]_{rw} = \alpha^0$, or $[\rho]_{rw} = \beta^0$. For simplicity, we denote it by ρ.

We now give the formal definition of the torus in homotopy type theory:

Definition 2.15 *The torus* $\mathbb{T}^2$ *is generated by:*

(i) A base point $x_0 : \mathbb{T}^2$.

(ii) Two base paths α *and* β *such that:* $x_0 \underset{\alpha}{=} x_0$ *and* $x_0 \underset{\beta}{=} x_0$.

(iii) One path co that establishes $\beta\alpha \underset{co}{=} \alpha\beta$, *i.e., a term* $co : Id(\beta\alpha, \alpha\beta)$.

Based on definition 2.15, we can establish the following definition in computational paths:

Definition 2.16 *In* $\mathbb{T}^2$, *we define the following canonical loops (canonical paths):*

(i) A base point $\alpha^0\beta^0 \underset{rw}{=} [\rho_{x_0}]_{rw}$.

(ii) The path $\beta^m\alpha^n = \tau(\alpha^n, \beta^m)$.

(iii) The path $= \sigma(\beta^m\alpha^n) = \sigma(\tau(\alpha^n, \beta^m))$.

(iv) One path co that establishes $\tau(\alpha^n, \beta^m) \underset{co}{=} \tau(\beta^m, \alpha^n)$.

By [12], given a point $x_0 \in \mathbb{T}^2$, the Torus can be expressed as a square whose sides are the base paths (loops) α and β, as shown in **figure 2**.

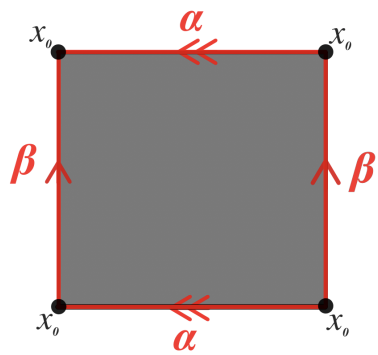


Figure 2: Square Torus representation with oriented paths α and β

Consider the following path in the figure:

$$\alpha^{-1}\circ\beta^{-1}\circ\alpha\circ\beta=\tau\Big(\tau\Big(\tau(\beta,\alpha),\sigma(\beta)\Big),\sigma(\alpha)\Big).$$

Proposition 2.17 *The aforementioned path is rw-equal to the reflexive path.*

Proof 2.18 *Indeed,*

$$\begin{array}{rcl}
\alpha^{-1}\circ\beta^{-1}\circ\alpha\circ\beta & = & \tau\Big(\tau\Big(\tau(\beta,\alpha),\sigma(\beta)\Big),\sigma(\alpha)\Big)\\
& \underset{tt}{=} & \tau\Big(\tau\Big(\beta,\tau\big(\alpha,\sigma(\beta)\big)\Big),\sigma(\alpha)\Big)\\
& \underset{co}{=} & \tau\Big(\tau\Big(\beta,\tau\big(\sigma(\beta),\alpha\big)\Big),\sigma(\alpha)\Big)\\
& \underset{\sigma(tt)}{=} & \tau\Big(\tau\Big(\tau(\beta,\sigma(\beta)),\alpha\Big),\sigma(\alpha)\Big)\\
& \underset{tr}{=} & \tau\Big(\tau\Big(\rho,\alpha\Big),\sigma(\alpha)\Big)\\
& \underset{tlr}{=} & \tau\Big(\alpha,\sigma(\alpha)\Big)\\
& \underset{tr}{=} & \rho.
\end{array}$$

and thus,

$$\alpha^{-1}\circ\beta^{-1}\circ\alpha\circ\beta=\tau\Big(\tau\Big(\tau(\beta,\alpha),\sigma(\beta)\Big),\sigma(\alpha)\Big)\underset{rw}{=}[\rho]_{rw}.$$

Lemma 2.19 *All path in $\mathbb{T}^2$ are rw-equal to the path $\beta^m\alpha^n$, with $m,n\in\mathbb{Z}$.*

Proof 2.20 *Let φ be a computational path in $\mathbb{T}^2$.*

- *If $\varphi=\rho$ then $\varphi=\tau(\alpha^0,\beta^0)=\beta^0\alpha^0$.*
- *If $\varphi=\sigma(\mu)=\sigma(\beta^m\alpha^n)=\sigma\Big(\tau(\alpha^n,\beta^m)\Big)\underset{stss}{=}\tau\Big(\sigma(\alpha^n),\sigma(\beta^m)\Big)=\tau(\alpha^n,\beta^{-m})=\beta^{-m}\alpha^{-n}$.*
- *If*

$$\begin{array}{rl}
\varphi & =\tau(\mu,\omega)\\
& =\tau(\beta^m\alpha^n,\beta^r\alpha^s)\\
& =\tau\Big(\tau(\alpha^n,\beta^m),\tau(\alpha^s,\beta^r)\Big)\\
& =(\beta^r\circ\alpha^s)\circ(\beta^m\circ\alpha^n)\\
& =\beta^r\circ\alpha^s\circ\beta^m\circ\alpha^n\\
& \underset{co}{=}\beta^r\circ\beta^m\circ\alpha^s\circ\alpha^n=\beta^{m+r}\alpha^{n+s}
\end{array}$$

Lemma 2.21 *All paths in* $\mathbb{T}^2$ *are generated by application* ρ,τ *and* σ *in base paths* $[loop_v^1]_{rw} = \alpha^1$ *and* $[loop_h^1]_{rw} = \beta^1$.

Proof 2.22 *Consider the following cases*

(i) Base case: $\beta^0\alpha^0 = \rho$.

(ii) $\rho \circ \alpha = \tau(\alpha, \rho) \underset{trr}{=} \alpha = \beta^0\alpha^1$.

(iii) $\rho \circ \beta = \tau(\beta, \rho) \underset{trr}{=} \beta = \beta^1\alpha^0$.

(iv) $\rho \circ \alpha^{-1} = \tau(\sigma(\alpha), \rho) \underset{trr}{=} \sigma(\alpha) = \beta^0\alpha^{-1}$.

(v) $\rho \circ \beta^{-1} = \tau(\sigma(\beta), \rho) \underset{trr}{=} \sigma(\beta) = \beta^{-1}\alpha^0$.

Assuming, by the induction hypothesis that every path in $\mathbb{T}^2$ *is rw-equal to* $\beta^m\alpha^n$, *we have:*

(1) $\rho \circ \beta^m\alpha^n = \tau(\beta^m\alpha^n, \rho) \underset{trr}{=} \beta^m\alpha^n$.

(2) $\alpha \circ \beta^m\alpha^n \underset{co}{=} \alpha \circ \alpha^n\beta^m = \alpha^{n+1}\beta^m \underset{co}{=} \beta^m\alpha^{n+1}$.

(3) $\beta \circ \beta^m\alpha^n = \beta^{m+1}\alpha^n$.

(4) $\beta^{-1} \circ \beta^m\alpha^n = (\beta^{-1} \circ (\beta \circ \beta^{m-1}))\alpha^n \underset{tt}{=} ((\beta^{-1} \circ \beta) \circ \beta^{m-1})\alpha^n \underset{tsr}{=} (\rho \circ \beta^{m-1})\alpha^n = \beta^{m-1}\alpha^n$.

(4) $\alpha^{-1} \circ \beta^m\alpha^n \underset{co}{=} \alpha^{-1} \circ \alpha^n\beta^m = (\alpha^{-1} \circ (\alpha \circ \alpha^{n-1}))\beta^m \underset{tt}{=} ((\alpha^{-1} \circ \alpha) \circ \alpha^{n-1})\beta^m \underset{tsr}{=} (\rho \circ \alpha^{n-1})\beta^m = \alpha^{n-1}\beta^m \underset{co}{=} \beta^m\alpha^{n-1}$.

So all paths in $\mathbb{T}^2$ *are rw-equal to* $\beta^m\alpha^n$.

Proposition 2.23 $\big(\Pi_1(\mathbb{T}^2, x_0), \circ\big)$ *is a group.*

Proof 2.24

(+): Sum

$$\cdot\ \frac{x_0 \underset{\beta^u\alpha^v}{=} x_0 \qquad x_0 \underset{\beta^r\alpha^s}{=} x_0}{x_0 \underset{\tau(\beta^u\alpha^v, \beta^r\alpha^s)}{=} x_0}$$

But,

$$\begin{aligned}
\tau(\beta^u\alpha^v, \beta^r\alpha^s) &= (\beta^r\alpha^s) \circ (\beta^u\alpha^v) \\
&= \beta^u\alpha^v\beta^r\alpha^s \\
&\underset{co}{=} \beta^u\beta^r\alpha^v\alpha^s \\
&= \beta^{u+r}\alpha^{v+s}
\end{aligned}$$

(σ): Inverse

$$\cdot\ \frac{x_0 \underset{\beta^m\alpha^n}{=} x_0 \qquad x_0 \underset{\sigma(\beta^m)\sigma(\alpha^n)}{=} x_0}{x_0 \underset{\tau(\beta^m\alpha^n,\sigma(\beta^m)\sigma(\alpha^n))}{=} x_0}$$

But,

$$\begin{aligned}\tau(\beta^m\alpha^n,\sigma(\beta^m)\sigma(\alpha^n)) &= (\sigma(\beta^m)\sigma(\alpha^n))\circ(\beta^m\alpha^n)\\ &= \sigma(\beta^m)\sigma(\alpha^n)\beta^m\alpha^n\\ &\underset{co}{=} \sigma(\beta^m)\beta^m\sigma(\alpha^n)\alpha^n\\ &\underset{tsr}{=} \rho_\beta\rho_\alpha\\ &\underset{trr}{=} \rho_{x_0}.\end{aligned}$$

On the other hand, we have:

$$\cdot\ \frac{x_0 \underset{\sigma(\beta^m)\sigma(\alpha^n)}{=} x_0 \qquad x_0 \underset{\beta^m\alpha^n}{=} x_0}{x_0 \underset{\tau(\sigma(\beta^m)\sigma(\alpha^n),\beta^m\alpha^n)}{=} x_0}$$

But,

$$\begin{aligned}\tau(\sigma(\beta^m)\sigma(\alpha^n),\beta^m\alpha^n) &= (\beta^m\alpha^n)\circ(\sigma(\beta^m)\sigma(\alpha^n))\\ &= \beta^m\alpha^n\sigma(\beta^m)\sigma(\alpha^n)\\ &\underset{co}{=} \beta^m\sigma(\beta^m)\alpha^n\sigma(\alpha^n)\\ &\underset{tr}{=} \rho_\beta\rho_\alpha \underset{trr}{=} \rho_{x_0}.\end{aligned}$$

(ϵ): Identity

$$\frac{x_0 \underset{\beta^m\alpha^n}{=} x_0 \qquad x_0 \underset{\rho_{x_0}}{=} x_0}{x_0 \underset{\tau\left(\beta^m\alpha^n,\rho_{x_0}\right)}{=} x_0}$$

But,

$$\begin{aligned}\tau(\beta^m\alpha^n,\rho_{x_0}) &= (\rho_{x_0})\circ(\beta^m\alpha^n)\\ &= \rho_{x_0}\beta^m\alpha^n\\ &\underset{tlr}{=} \beta^m\alpha^n\end{aligned}$$

and so

$$\tau(\beta^m\alpha^n,\rho_{x_0}) \underset{trr}{=} \beta^m\alpha^n.$$

On the other hand, we have:

$$\cdot\ \frac{x_0 \underset{\rho_{x_0}}{=} x_0 \qquad x_0 \underset{\beta^m\alpha^n}{=} x_0}{x_0 \underset{\tau\left(\rho_{x_0},\beta^m\alpha^n\right)}{=} x_0}$$

But,

$$\begin{aligned}\tau\left(\rho_{x_0},\beta^m\alpha^n\right) &= \left(\beta^m\alpha^n\right)\circ\left(\rho_{x_0}\right)\\ &= \beta^m\alpha^n\rho_{x_0}\\ &\underset{trr}{=} \beta^m\alpha^n\end{aligned}$$

and so

$$\tau(\rho_{x_0},\beta^m\alpha^n) \underset{trr}{=} \beta^m\alpha^n.$$

(∘): Associativity

$$\cdot\ \frac{\dfrac{x_0 \underset{\beta^m\alpha^n}{=} x_0 \qquad x_0 \underset{\beta^i\alpha^j}{=} x_0}{x_0 \underset{\tau(\beta^m\alpha^n,\beta^i\alpha^j)}{=} x_0} \qquad x_0 \underset{\beta^r\alpha^s}{=} x_0}{x_0 \underset{\tau(\tau(\beta^m\alpha^n,\beta^i\alpha^j),\beta^r\alpha^s)}{=} x_0}$$

But,

$$\begin{aligned}\tau\left(\tau\left(\beta^m\alpha^n,\beta^i\alpha^j\right),\beta^r\alpha^s\right) &= \left(\beta^r\alpha^s\right)\circ\tau(\beta^m\alpha^n,\beta^i\alpha^j)\\ &= \left(\beta^r\alpha^s\right)\circ\left(\beta^i\alpha^j\circ\beta^m\alpha^n\right)\\ &= \left(\beta^r\alpha^s\right)\circ\left(\beta^i\alpha^j\beta^m\alpha^n\right)\\ &= \beta^r\alpha^s\beta^i\alpha^j\beta^m\alpha^n.\end{aligned}$$

On the other hand, we have:

$$\cdot\ \frac{x_0 \underset{\beta^m\alpha^n}{=} x_0 \qquad \dfrac{x_0 \underset{\beta^i\alpha^j}{=} x_0 \qquad x_0 \underset{\beta^r\alpha^s}{=} x_0}{x_0 \underset{\tau(\beta^i\alpha^j,\beta^r\alpha^s)}{=} x_0}}{x_0 \underset{\tau(\beta^m\alpha^n,\tau(\beta^i\alpha^j,\beta^r\alpha^s))}{=} x_0}$$

But,

$$\begin{aligned}\tau\left(\beta^m\alpha^n, \tau\left(\beta^i\alpha^j, \beta^r\alpha^s\right)\right) &= \tau(\beta^i\alpha^j, \beta^r\alpha^s) \circ (\beta^m\alpha^n)\\ &= (\beta^r\alpha^s \circ \beta^i\alpha^j) \circ (\beta^m\alpha^n)\\ &= (\beta^r\alpha^s\beta^i\alpha^j) \circ (\beta^m\alpha^n)\\ &= \beta^r\alpha^s\beta^i\alpha^j\beta^m\alpha^n.\end{aligned}$$

Therefore, it follows that $\left(\Pi_1(\mathbb{T}^2, x_0), \circ\right)$ *is a group.*

Theorem 2.25 $\Pi_1\left(\mathbb{T}^2, x_0\right) \simeq \mathbb{Z} \times \mathbb{Z}$.

Proof 2.26 *Consider the map:*

$$\begin{aligned}toPath^2 : \mathbb{Z} \times \mathbb{Z} &\longrightarrow \Pi_1\left(\mathbb{T}^2, x_0\right)\\ (m, n) &\longrightarrow \beta^m\alpha^n.\end{aligned}$$

i $toPath^2$ *is a homomorphism.*
Let $(m_1 + m_2, n_1 + n_2) \in \mathbb{Z} \times \mathbb{Z}$, *then:*

$$\begin{aligned}toPath^2(m_1 + m_2, n_1 + n_2) &= \beta^{m_1+m_2}\alpha^{n_1+n_2}\\ &= \beta^{m_1}\beta^{m_2}\alpha^{n_1}\alpha^{n_2}\\ &\underset{co}{=} \beta^{m_1}\alpha^{n_1}\beta^{m_2}\alpha^{n_2}\\ &= toPath^2(m_1, n_1) \circ toPath^2(m_2, n_2).\end{aligned}$$

ii $toPath$ *is surjective.*
By ***Lemma 2.19****, as every path in* $\mathbb{T}^2$ *is rw-equal to one path* $\beta^m\alpha^n$, *we have that for all path* $\beta^i\alpha^j \in \Pi_1(\mathbb{T}^2), \exists (i, j) \in \mathbb{Z} \times \mathbb{Z}$, *such that,* $toPath(i, j) = \beta^i\alpha^j$.

iii $Ker(toPath^2) = \{(0, 0)\}$.
Suppose there is $(m, n) \neq (0, 0) \in \mathbb{Z} \times \mathbb{Z}$, *such that* $(m, n) \in Ker(toPath^2)$. *Thus,*

$$\begin{aligned}toPath^2(m, n) &= toPath^2(m + 0, n + 0)\\ &\overset{hom}{=} toPath^2(m, n) \circ toPath^2(0, 0)\\ &= \beta^m\alpha^n\beta^0\alpha^0\\ &= \tau(\beta^0\alpha^0, \beta^m\alpha^n)\\ &= \tau(\rho, \beta^m\alpha^n)\\ &= \rho.\end{aligned}$$

If $\tau(\rho, \alpha) = \rho$, *by rw-rule* $\underset{tr}{\triangleright}$ *we have,* $\alpha = \sigma(\rho) \Rightarrow (m, n) = (0, 0) \rightarrow\leftarrow$.
Therefore,

$$Ker(toPath^2) = \{(0, 0)\}.$$

As $ToPath^2$ *is a homomorphism surjective with* $Ker(toPath^2) = \{(0, 0)\}$, *then* $toPath^2$ *is a isomorphism, that is,* $\Pi_1(\mathbb{T}^2) \simeq \mathbb{Z} \times \mathbb{Z}$.

3 Fundamental Group of the Real Projective Plane

The real projective plane, denoted by $\mathbb{RP}^2$, is by definition the set of all straight lines that pass through the origin of space $\mathbb{R}^3$. We can define each of these lines by a position vector v_r, with $\|v_r\| \neq 0$, this way we have that $\mathbb{RP}^2$ is a quotient space of $\mathbb{R}^3 - (0,0)$ under the equivalence relation $v_r \sim \lambda v_r$ for scalars $\lambda \neq 0$. If we impose the condition that the vectors $\|v_r\| = 1$ then $\mathbb{RP}^2$ is a quotient space $\mathbb{S}^2$ under the equivalence relation $v_r \sim -v_r$, the sphere with antipodal points identified, where v_r is position vector.

Let $[v_r] = [x, y, z]$, where $[x, y, z] = \{v_r = (x, y, z), -v_r = (-x, -y, -z)\}$ with $z \neq 0$. This is equivalent to saying that $\mathbb{RP}^2$ is the quotient space of a upper hemisphere $\mathbb{D}^2$ with antipodal points of $\partial\mathbb{D}^2$ identified, as show in **figure 3**.

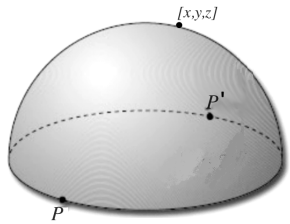


Figure 3: P and $P^{'}$ are antipodal points in $\partial\mathbb{D}^2$.

Let's then map it on the unit disk through the following map $[x, y, z] \longrightarrow (x, y, 0)$, as follows in the **figure 4**)

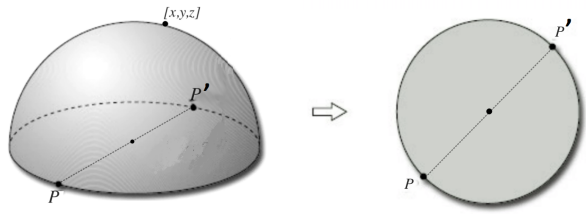


Figure 4: Mapping Projection in the unit disk on xy plane.

This way we have that $\mathbb{RP}^2$ is a quotient space of D with antipodal points of ∂D identified. Therefore we can study the fundamental group of $\mathbb{RP}^2$ by the disk shown on the right side of the **figure 4**.

We denote by α any *loop* that connects the identified antipodal points, so we can consider α as a *loop* (as follows in the **figure 5**) and any other *loop* that connects the identified antipodal points is homotopic to α. Note that $\forall Q \in D$, any loop based on Q is homotopic to the point, these are not in our interest to study.

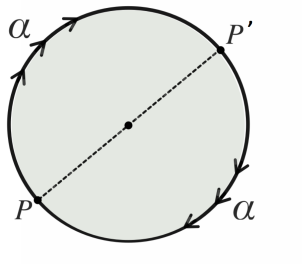


Figure 5: *loop* α.

Since we can represent the real projective plane $\mathbb{RP}^2$ for a disk D, we can define $\mathbb{RP}^2$, homotopically, as follows:

Definition 3.1 *The real projective plane* $\mathbb{RP}^2$ *is defined by:*

(i) The types $Q : D$*, such that* $Q \in D$*.*

(ii) The pair $P, P' : \partial D$*, such that:* P, P' *are the pairs of antipodal points identified in* ∂D*.*

(iii) A path α *such that:* $P \underset{\alpha}{=} P'$*.*

(iv) A path cicl that establishes $\alpha \circ \alpha \underset{cicl}{=} \rho$*, i.e,* $cicl : Id_{\mathbb{P}^2}(\alpha \circ \alpha, \rho)$*.*

Lemma 3.2 *All paths in* $\mathbb{P}^2$ *generated by* ρ*,* σ *and* τ *are rw-equal to* ρ *or* α*.*

Proof 3.3 *Consider the following base cases:*

Base case $\varphi = \rho$***:***

(i) Trivial case.

Base case $\varphi = \sigma(\phi)$***:***

(i) $\varphi = \sigma(\alpha) \underset{cicl}{=} \alpha$*.*

(ii) $\varphi = \sigma(\rho) \underset{rw}{=} \rho$*.*

Base case $\varphi = \tau(\phi, \kappa)$***:***

(i) $\varphi = \tau(\rho, \rho) \underset{trr}{=} \rho$

(ii) $\varphi = \tau(\alpha, \rho) \underset{trr}{=} \alpha$

(iii) $\varphi = \tau(\rho, \alpha) \underset{tlr}{=} \alpha$

(iv) $\varphi = \tau(\alpha, \alpha) \underset{cicl}{=} \rho$

Inductive case: *Assuming true for* n*, we have:*

If $[loop^n]_{rw} = [\rho]_{rw}$*, we have two possibilities for* $n + 1$*:*

(i) $[loop^{n+1}]_{rw} = [loop^n]_{rw} \circ [\rho]_{rw} = \tau(\rho, \rho) \underset{trr}{=} [\rho]_{rw}$.

(ii) $[loop^{n+1}]_{rw} = [loop^n]_{rw} \circ \alpha = [\rho]_{rw} \circ \alpha = \tau(\alpha, \rho) \underset{trr}{=} \alpha$.

If $[loop^n]_{rw} = \alpha$, *we have two possibilities for* $n+1$*:*

(i) $[loop^{n+1}]_{rw} = [loop^n]_{rw} \circ \alpha = \alpha \circ \rho = \tau(\rho, \alpha) \underset{tlr}{=} \alpha$.

(ii) $[loop^{n+1}]_{rw} = [loop^n] \circ \alpha = \alpha \circ \alpha = \tau(\alpha, \alpha) \underset{cicl}{=} [\rho]_{rw}$.

Thus, all paths in $\mathbb{P}^2$ generated by ρ or α are *rw-equal* to either α or ρ. Since we have $\alpha \circ \alpha = \tau(\alpha, \alpha) \underset{cicl}{=} [\rho]_{rw}$, the term *cicl* give us one important result: $\alpha = \sigma(\alpha)$.

Proposition 3.4 $\left(\Pi_1(\mathbb{P}^2), \circ\right)$ *is a group.*

Proof 3.5

(+): Sum

$$\frac{P \underset{\alpha}{=} P \qquad P \underset{\alpha}{=} P}{P \underset{\tau(\alpha,\alpha)}{=} P}$$

But,

$$\alpha \circ \alpha = \tau(\alpha, \alpha) \underset{cicl}{=} \rho \in \Pi_1\left(\mathbb{P}^2\right).$$

(σ): Inverse

$$\frac{P \underset{\alpha}{=} P \qquad P \underset{\sigma(\alpha)}{=} P}{P \underset{\tau(\alpha,\sigma(\alpha))}{=} P}$$

But,

$$\sigma(\alpha) \circ \alpha = \tau(\alpha, \sigma(\alpha)) \underset{tr}{=} \rho \in \Pi_1\left(\mathbb{P}^2\right).$$

On the other hand, we have:

$$\frac{P \underset{\sigma(\alpha)}{=} P \qquad P \underset{\alpha}{=} P}{P \underset{\tau(\sigma(\alpha),\alpha)}{=} P}$$

But,

$$\alpha \circ \sigma(\alpha) = \tau(\sigma(\alpha), \alpha) \underset{tsr}{=} \rho \in \Pi_1\left(\mathbb{P}^2\right).$$

(ϵ): Identity

$$\frac{P \underset{\alpha}{=} P \qquad P \underset{\rho}{=} P}{P \underset{\tau(\alpha,\rho)}{=} P}$$

But,

$$\rho \circ \alpha = \tau(\alpha, \rho) \underset{tlr}{=} \alpha \in \Pi_1\left(\mathbb{P}^2\right).$$

On the other hand, we have:

$$\frac{P \underset{\rho}{=} P \qquad P \underset{\alpha}{=} P}{P \underset{\tau(\rho,\alpha)}{=} P}$$

But,

$$\alpha \circ \rho = \tau(\rho, \alpha) \underset{trr}{=} \alpha \in \Pi_1\left(\mathbb{P}^2\right).$$

($\circ$): Associativity

$$\frac{\dfrac{P \underset{\alpha}{=} P \qquad P \underset{\alpha}{=} P}{P \underset{\tau(\alpha,\alpha)}{=} P} \qquad P \underset{\alpha}{=} P}{P \underset{\tau(\tau(\alpha,\alpha),\alpha)}{=} P}$$

But,

$$\begin{aligned} \tau(\tau(\alpha,\alpha),\alpha) &= \alpha \circ \tau(\alpha,\alpha) \\ &\underset{cicl}{=} \alpha \circ \rho \\ &= \tau(\rho,\alpha) \\ &\underset{trr}{=} \alpha \end{aligned}$$

On the other hand, we have:

$$\frac{P \underset{\alpha}{=} P \qquad \dfrac{P \underset{\alpha}{=} P \qquad P \underset{\alpha}{=} P}{P \underset{\tau(\alpha,\alpha)}{=} P}}{P \underset{\tau(\alpha,\tau(\alpha,\alpha))}{=} P}$$

But,

$$\begin{aligned}\tau(\alpha,\tau(\alpha,\alpha)) &= \tau(\alpha,\alpha)\circ\alpha\\ &\underset{cicl}{=} \rho\circ\alpha\\ &= \tau(\alpha,\rho)\\ &\underset{tlr}{=} \alpha\end{aligned}$$

Since $\tau(\tau(\alpha,\alpha),\alpha)=\tau(\alpha,\tau(\alpha,\alpha))$*, it follows that associativity is valid and therefore* $\left(\Pi_1(\mathbb{P}^2),\circ\right)$ *is a group generated by* ρ *and* α*.*

Theorem 3.6 $\Pi_1(\mathbb{P}^2)\simeq\mathbb{Z}_2$*.*

Proof 3.7 *Consider the application defined and denoted by:*

$$\begin{aligned}toPath_{\mathbb{Z}_2}: \quad & \mathbb{Z}_2\to\Pi_1\left(\mathbb{P}^2\right)\\ & z\to toPath_{\mathbb{Z}_2}=[loop^z]_{rw}.\end{aligned}$$

(i) $toPath_{\mathbb{Z}_2}$ *is a homomorphism.*
Let z_1 *and* $z_2\in\mathbb{Z}_2$*, then:*

$$\begin{aligned}toPathh_{\mathbb{Z}_2}(z_1+z_2) &= [loop^{z_1+z_2}]_{rw}\\ &= \tau([loop^{z_1}]_{rw},[loop^{z_2}]_{rw})\\ &= toPathh_{\mathbb{Z}_2}(z_2)\circ toPathh_{\mathbb{Z}_2}(z_1).\end{aligned}$$

By the other hand,we have:

$$\begin{aligned}toPathh_{\mathbb{Z}_2}(z_2+z_1) &= [loop^{z_2+z_1}]_{rw}\\ &= \tau([loop^{z_2}]_{rw},[loop^{z_1}]_{rw})\\ &= toPathh_{\mathbb{Z}_2}(z_1)\circ toPathh_{\mathbb{Z}_2}(z_2).\end{aligned}$$

Thus, $toPath_{\mathbb{Z}_2}(z_1+z_2)=toPath_{\mathbb{Z}_2}(z_1)\circ toPath(z_2)$*.*

(ii) $toPath_{\mathbb{Z}_2}$ *is surjective.*
By ***Lemma 3.2****, every path in* $\mathbb{P}^2$ *is rw-equal to* ρ *and* α*. So given any path in* $\Pi_1(\mathbb{P}^2)$*,for* $z=0$ *and* $z=1$ *we have* $\rho=toPath_{\mathbb{Z}_2}(0)$ *and* $\alpha=toPath_{\mathbb{Z}_2}(1)$*, respectively.*

(iii) $Ker(toPath_{\mathbb{Z}_2})=\{0\}$*.*
By ***Lemma 3.2****, there is only one element in* $z\in\mathbb{Z}_2$ *that* $toPathh_{\mathbb{Z}_2}(z)=0$*. Therefore,* $Ker(toPath_{\mathbb{Z}_2})=\{0\}$*.*

$$toInt=\begin{cases}toInt([loop^0]_{rw}=[\rho]_{rw})=0\\ toInt([loop^1]_{rw}=\alpha)=1\end{cases}$$

Thus, the isomorphism holds

4 Conclusion

Our main objective has been the calculation of the fundamental groups of many surfaces using a labelled deduction system based on the concept of computational paths (sequences of rewrites). The main advantage of this approach is that we avoid the use of more complex techniques, such as those made in algebraic topology in pure mathematics or by the method of encoding-decoding used in homotopy type theory. As a consequence, our calculations proved to be straightforward and simple. Using computational paths as our main tool, we have calculated the fundamental group of the circle, torus and projective plane. Therefore, we have shown that it is possible to use the theory of computational paths to obtain useful results in algebraic topology.

Finally, an almost natural question of our study would be: is it possible to calculate the fundamental group of the Klein bottle using the same technique? This question is a new north to develop our study.

References

[1] CHENADEC, P. L. On the logic of unification. Journal of Symbolic computation, Elsevier, v. 8,n. 1, p. 141–199, 1989.

[2] CURRY, H. B. 1900-1982, HINDLEY, J.R. and SELDIN, J. P. To H. B. Curry: Essays on Combinatory Logic Lambda Calculus and Formalism. edited by J.P. Seldin, J.R. Hindley. p.xxv+606pp,1980. London by Academic press.

[3] DERSHOWITZ, N. Orderings for term-rewriting systems.Theoretical computer science,Elsevier, v. 17, n. 3, p. 279–301, 1982.

[4] HARPER, R. Type Theory Foundations. 2012. Type Theory Foundations, Lecture at OregonProgramming Languages Summer School, Eugene, Oregon.

[5] HINDLEY, J. R. and SELDIN, J. P. Lambda-calculus and combinators: an introduction, 2008. Cambridge University Press.

[6] HOFMANN, M.; STREICHER, T. The groupoid model refutes uniqueness of identity proofs.In: IEEE.Logic in Computer Science, 1994. LICS'94. Proceedings., Symposium on. [S.l.], 1994.p. 208–212

[7] KNUTH, D. E.; BENDIX, P. B. Simple word problems in universal algebras. In:Computationalproblems in abstract algebra. [S.l.: s.n.], 1970. p. 263–297.

[8] LICATA, D. R. and Shulman, M. Calculating the Fundamental Group of the Circle in Homotopy Type Theory, 28th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2013. ACM/IEEE.

[9] MARTIN-Löf, P. An intuitionistic theory of types: predicative part, in H. E. Rose and J.C.Shepherdson (eds.) Logic Colloquium '73, Vol. 80 of Studies in Logic and The Foundations of Mathematics, pp 73 -– 118, North-Holland, Amsterdam,1975. p.viii+513pp, Proceedings of the Colloquium held in Bristol,UK,1973.

[10] MARTIN-Löf, P. Constructive Mathematics and Computer Programming, in L. J. Cohen, J. Los, H.Pfeiffer, and K.-P. Podewski (eds.). Logic, Methodology and Philosophy of Science VI, Series Studies in Logic and The Foundations of Mathematics,pp 153—175, North-Holland, Amsterdam. p.xiii+738pp,1992. Proceedings of the International Congress held in Hannover, August 22–29 1979.

[11] MARTIN-Löf, P. Intuitionistic Type Theory. Series Studies in Proof Theory. Bibliopolis Naples, iv+91pp, 1980. Notes by Giovanni Sambi of a series of lectures given in Padova.

[12] MUNKRES, J. R. Topology (2nd Edition), Chapter 9, Topic 54 - The fundamental group of circle, Pearson, 2000. Institute for Advanced Study.

[13] de OLIVEIRA, A. G. Proof transformations for labelled natural deduction via term rewriting. Master's thesis, Depto. de Informática, Universidade Federal de Pernambuco, Recife, Brazil, April 1995.

[14] de OLIVEIRA, A. G. and de QUEIROZ, R. J. G. B. A New Basic Set of Proof Transformations. In We Will Show Them! Essays in Honour of Dov Gabbay. South American Journal of Logic Volume 2 (2), p.499-528. S. Artemov, H. Barringer, A. Garcez, L. Lamb and J. Woods (eds.), 2005. College Publications, London, ISBN 1904987125.

[15] de OLIVEIRA, A. G. . and de QUEIROZ, R. J. G. B. A normalization procedure for the equational fragment of labelled natural deduction. Logic Journal of IGPL .Oxford Univ Press. Vol 7 (2):173-215. March 1999.

[16] de QUEIROZ, R. J. G. B. and GABBAY, D. M. Equality in Labelled Deductive Systems and the Functional Interpretation of Propositional Equality. p.547-565,1994. ILLC/Department of Philosophy, University of Amsterdam

[17] de QUEIROZ, R. J. G. B. and de OLIVEIRA, A. G. and Ramos, A. F. Propositional equality, identity types, and direct computational paths. South American Journal of Logic, 2(2). p.245-296, 2016. Special Issue A Festschrift for Francisco Miraglia, M. E. Coniglio, H. L. Mariano and V. C. Lopes (Guest Editors).

[18] de QUEIROZ, R. J. G. B. and de OLIVEIRA, A. G. Term rewriting systems with labelled deductive systems. Proceedings of Brazilian Symposium on Artificial Intelligence (SBIA'94),p.59-74,1994.

[19] de QUEIROZ, R. J. G. B. and GABBAY,D. M. The Functional Interpretation of the Existential Quantifier. Bulletin of the Interest Group in Pure and Applied Logics, 3(2 and 3). p.243–290, 1995. Abstract in JSL 58(2):753–754, 1993. Presented at Logic Colloquium '91, Uppsala, August 9–16.

[20] de QUEIROZ, R. J. G. B. and Gabbay,D. M. Labelled Natural Deduction. InLogic, Language and Reasoning. Essays in Honor of Dov Gabbay's 50th Anniversary, H.J. Ohlbach and U. Reyle (eds.). Kluwer Academic Publishers, June 1999. pp. 173-–250.

[21] de QUEIROZ, R. J. G. B. de OLIVEIRA, A. G. Natural deduction for equality: The missing entity. Advances in Natural Deduction - A Celebration of Dag Prawitz's Work, p.63-91. Springer,2014.

[22] de QUEIROZ, R. J. G. B. and de OLIVEIRA, A. G. Propositional Equality, Identity Types and Reversible Rewriting Sequences as Homotopies. Talk given at Workshop de Lógica, Universidade Federal do Ceará, Fortaleza, CE. 2014.

[23] de QUEIROZ, R. J. G. B. and de OLIVEIRA, A. G. and Gabbay, D. M. The Functional Interpretation of Logical Deduction. World Scientific, 2011.

[24] RAMOS, A.F. and de QUEIROZ, R. J. G. B. and de OLIVEIRA, A. G. On the identity type as the type of computational paths. Logic Journal of the IGPL,vol.25 (4), p.562-584, 2017.

[25] RAMOS, A.F. Explicit Computational Paths in Type Theory. PhD Thesis, 82018. Depto. de Informática, Universidade Federal de Pernambuco, Recife, Brazil,August, 2018.

[26] RAMOS, A.F. and de QUEIROZ, R. J. G. B. and de OLIVEIRA, A. G. and de VERAS, T. M. L. Explicit Computational Paths. South American Journal of Logic Vol. 4, n. 2, pp. 441–484, 2018 ISSN: 2446-6719.

[27] de VERAS, T.M.L., RAMOS, A.F., de QUEIROZ, R.J.G.B., and de OLIVEIRA, A.G. Computational Paths - A Weak Groupoid. Journal of Logic and Computation, https://doi.org/10.1093/logcom/exad071 Published: 24 November 2023

[28] The Univalent Foundations Program. Homotopy Type Theory: Univalent Foundations of Mathematics, Institute for Advanced Study, 2013. https://homotopytypetheory.org/book.

[29] VOEVODSKY, V. to ESCARDO, M., In the vein of bringing to public record things that Vladimir said, here is a short interview. http://plato.stanford.edu/archives/win2014/entries/category-theory/, interview published in google discussion group of Homotopy Type Theory, by Martin H. Escardo On Thursday, 12 October 2017 20:24:26, in response to an interview by Martin H. Escardó with Vladimir Voevodsky on 2015.

[30] VOEVODSKY, V. Univalent Foundations and Set Theory, 2014. Lecture at IAS, Princeton, New Jersey, Mar 2014.

Tiago Mendonça Lucena de Veras
Department of Mathematics
Rural Federal University of Pernambuco (UFRPE)
R. Manuel de Medeiros, Dois Irmãos, CEP 52171-900, Recife, PE, Brazil
E-mail: tiago.veras@ufrpe.br

Arthur F. Ramos
UFPE Informatics Center
Federal University of Pernambuco (UFPE)
Av.Jorn. Aníbal Fernandes s/n, Cidade Universitária, CEP 50740-560, Recife, PE, Brazil
E-mail: afr@cin.ufpe.b

Ruy J. G. B. de Queiroz
UFPE Informatics Center
Federal University of Pernambuco (UFPE)
Av.Jorn. Aníbal Fernandes s/n, Cidade Universitária, CEP 50740-560, Recife, PE, Brazil
E-mail: ruy@cin.ufpe.br

Anjolina G. de Oliveira
UFPE Informatics Center
Federal University of Pernambuco (UFPE)
Av.Jorn. Aníbal Fernandes s/n, Cidade Universitária, CEP 50740-560, Recife, PE, Brazil
E-mail: ago@cin.ufpe.br