On Reversible Combinatory Logic

Alessandra Di Pierro [1]

Dipartimento di Informatica, Universitá di Pisa, Italy

Chris Hankin and Herbert Wiklicky [1,1]

Department of Computing, Imperial College London, United Kingdom

Abstract

The λ-calculus is destructive: its main computational mechanism – beta reduction – destroys the redex and makes it thus impossible to replay the computational steps. Recently, reversible computational models have been studied mainly in the context of quantum computation, as (without measurements) quantum physics is inherently reversible. However, reversibility also changes fundamentally the semantical framework in which classical computation has to be investigated. We describe an implementation of classical combinatory logic into a reversible calculus for which we present an algebraic model based on a generalisation of the notion of group.

1 Introduction

It has been suggested, e.g. [11], that the standard model for computation as embodied in Turing Machines answers the problem of what constitutes a "computational procedure" in Hilbert's 10th Problem by reference to mental arithmetic as practised in previous times by European school children, accountants and waiters. This "waiter's arithmetic" is non-reversible and destructive. It is open to speculation whether a culture based on reversible computation like an abacus would have developed a different basic computational model. Quantum computation and the need for minimal energy loss make reversible computation once again interesting, see e.g. [15]. This has been the motivation for van Tonder [14] who presents a reversible applied lambda calculus (with quantum constants); his operational semantics provided the inspiration for the operational semantics of our reversible version **rCL** of Combinatory Logic. On the other hand, the set of combinators that we consider here have

[1] The authors are partly funded by the EPSRC project S77066A "Quantitative Analysis of Computational Resources".

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

also been studied by Abramsky [1,2], although with a different motivation, namely the links between reversible calculus and linear logic.

Our main motivation for investigating a reversible version of Combinatory Logic is ultimately the development of a denotational semantics of (probabilistic versions of) the λ-calculus reflecting the operational semantics we introduced in [10]. This kind of semantics is based on linear operator algebras and aims to support a compositional approach to (probabilistic) program analysis. The close relationship between reversibility and certain important classes of linear operators – in particular unitary and normal operators – was the starting point of a deeper investigation of the structure of reversible computation.

Reversibility naturally introduces a notion of symmetry into computation and is therefore strongly related to the theory of groups and their action, which are considered by most mathematicians as being virtually synonymous of symmetry [16]. Starting from the work on invertible lambda terms we could then use the group of permutations for the classical λ-calculus as our mathematical base for investigating compositionality issues in the static analysis of complex systems. However, the notion of automorphism associated to group is in some sense too "trivial" to characterise the symmetry involved in a reversible computation; it turns out that the structure of these objects can be better characterised algebraically by using *groupoids* and not just groups. In fact, the extension from groups to groupoids was formally introduced to describe reversible processes which may traverse a number of states. These cannot be easily captured by using group theory as this only allows us to characterise processes which start from one point and (possibly after a number of steps) come back to the same point. On the contrary, in groupoid theory processes can have different start and end points but they can be composed if and only if the starting point of one process is the end point of the previous one. Intuitively, a groupoid can be thought of as a group with many identities [6]. It is interesting to note that according to Connes, quantum mechanics was discovered by considering the groupoid of quantum transitions rather than the group of symmetry [7].

2 The Groupoid Structure of Reversible Computations

Our approach to reversible computation is based on a particular algebraic model of computation which naturally reflects the operational meaning of term reduction and its reverse process. This model is based on the notion of a *groupoid*. A groupoid, also known as a *virtual group*, is an algebraic structure introduced by Brandt [5] (for further details see e.g. [13,16,12,6]).

Definition 2.1 A *groupoid* with *base* $\mathcal{B}$ is a set $\mathcal{G}$ with mappings α and β from $\mathcal{G}$ onto $\mathcal{B}$, a partially defined binary operation $(g,h) \mapsto g \cdot h = gh$, and an inverse map $g \mapsto g^{-1}$ from $\mathcal{G}$ to $\mathcal{G}$ satisfying the following conditions:

(i) gh is defined whenever $\beta(g) = \alpha(h)$, and in this case $\alpha(gh) = \alpha(g)$ and $\beta(gh) = \beta(h)$.

(ii) if gh and hk are defined then so are $(gh)k$ and $g(hk)$ and they are equal (associativity).

(iii) For each $g \in \mathcal{G}$, there are left- and right-identity elements λ_g and ϱ_g satisfying $\lambda_g g = g = g\varrho_g$.

(iv) Each $g \in \mathcal{G}$ has an inverse g^{-1} satisfying $g^{-1}g = \varrho_g$ and $gg^{-1} = \lambda_g$.

An important property of groupoids is that $\alpha(g)$ and λ_g ($\beta(g)$ and $\rho(g)$) determine one another, that is there is a bijection between the base $\mathcal{B}$ and the set of all the identity elements of $\mathcal{G}$. This implies that identities are essentially unique, or in other words there is only one object which is "composable" on the left (right) with a given object g.

Consider a set of 'computational processes' $\mathcal{C} = \{C_i\}_i$, such as for example those specified via some programs in a formal programming language. Among the elements in $\mathcal{C}$ we would usually expect to find a 'neutral' or 'identity computation', that is intuitively a computational process which does not change any input state, such as for example a **skip** statement in a **while** language. We would also usually be able to define a way to compose two processes sequentially, in the sense that we can feed the result of one computation C_1 into another computation C_2 by obtaining a new one usually denoted by $C_1; C_2$ or $C_2 \circ C_1$. Moreover, In order for the computation be reversible we need a notion of symmetry. These general ideas which essentially identify reversible computation with some kind of (symbolic) dynamical process establish a groupoid structure on $\mathcal{C}$. In order to see this more precisely, consider a calculus with a notion of "reduction" i.e. a transition relation $\rightarrowtail$ between terms $T, \ldots \in \mathcal{T}$, and its transitive and reflexive closure $\twoheadrightarrow$. Define a computational groupoid $\mathcal{G} = \mathcal{G}(\mathcal{T}, \twoheadrightarrow)$ as follows:

- $\mathcal{G} \subseteq \mathcal{T} \times \mathcal{T}$ with $(T, T') \in \mathcal{G}$ iff $T \twoheadrightarrow T'$.
- $\mathcal{B} = \mathcal{T}$
- $\alpha((T, T')) = T$ and $\beta((T, T')) = T'$
- $(T, T') \cdot (T', T'') = (T, T'')$
- $\lambda_{(T,T')} = (T, T)$ and $\varrho_{(T,T')} = (T', T')$
- $(T, T')^{-1} = (T', T)$.

Intuitively, we can reverse a computation for a term $T \in \mathcal{T}$ if we keep information about its 'history', i.e. information about the transition steps that have been performed during the computation. A highly expensive way to make the transition relation $\rightarrowtail$ reversible is to use as history all strings $H \in \mathcal{H} = \mathcal{T}^*$ of terms in $\mathcal{T}$ and replace each transition $T_1 \rightarrowtail T_2$, by $\langle T_1 \mid H \rangle \rightarrowtail \langle T_2 \mid HT_1 \rangle$ for all $H \in \mathcal{H}$. In this way we record the complete history of the previous terms and it is easy to see that computation is now reversible, i.e. we can reverse a derivation path until we reach the initial term.

In general, we might be interested in a more "efficient" way of recording the derivation history of a term. However, it depends on the nature and structure of the original calculus what information the history has to record; for example, in Van Tonder's λ-calculus [14] the history keeps track only of the substitutions which take place in each β-reduction step.

3 Combinatory Logic

Combinatory Logic (**CL**) is a formalism which (similarly to the λ-calculus) was introduced to describe functions and certain primitive ways to combine them to form other functions. With respect to the λ-calculus it has the advantage that is variable free; this allows one to avoid all the technical complications concerned with substitutions and congruence. It has on the other hand the disadvantage of being less intuitive than the λ-notation. For the purpose of this work we have opted for this more involved formalism because it allows for a more agile treatment and definition of our notion of reversible computation.

Definition 3.1 [Combinatory Logic Terms] The set of combinatory logic terms, **CL**-terms, over a finite or infinite set of constants containing K and S and an infinite set of variables is defined inductively as follows:

(i) all variables and constants are **CL**-terms,

(ii) if X and Y are **CL**-terms, then (XY) is a **CL** term.

The two combinators S and K form a common basis for combinatory logic. However, other sets of basic combinators can be defined. We will use the base consisting of four basic operations encoded in the combinators B (implementing bracketing), C (elementary permutations), W (duplication), and K (for deletion) which are defined as follows (cf [8, p379]):

$$\mathsf{K} \equiv \lambda xy.x, \ \mathsf{W} \equiv \lambda xy.xyy, \ \mathsf{C} \equiv \lambda xyz.xzy, \ \mathsf{B} \equiv \lambda xyz.x(yz).$$

Importantly, we can use B, W and C to implement the common combinator S (cf [8, p155]):

$$\mathsf{S} \equiv \mathsf{B}(\mathsf{B}(\mathsf{BW})\mathsf{C})(\mathsf{BB}).$$

In order to generate equalities provable in this calculus we use a notion of reduction similar to the *weak reduction* for the SK-calculus [3]. This is defined as the smallest extension of the relation on **CL**-terms induced by the basic operators which is compatible with application.

Definition 3.2 The reduction relation $\rightarrow$ on **CL**-terms is defined by the following rules:

(i) $\mathsf{K}XY \rightarrow X$,

(ii) $\mathsf{W}XY \rightarrow XYY$,

(iii) $\mathsf{C}XYZ \rightarrow XZY$,

(iv) $\mathsf{B}XYZ \rightarrow X(YZ)$,

(v) $X \rightarrow X'$ implies $XY \rightarrow X'Y$,

(vi) $X \rightarrow X'$ implies $YX \rightarrow YX'$,

We will denote by $\twoheadrightarrow$ the reflexive transitive closure of $\rightarrow$.

The relation between the λ-calculus and **CL** is a standard result (cf. [3, p156]). With reference to the standard base $\{\mathsf{S}, \mathsf{K}\}$ there is a canonical encoding $(\ )_{\mathbf{CL}}$ of λ terms in **CL** terms. It is a well known result that in presence of a rule for extensionality the two theories λ-calculus and **CL**(which are in general not equivalent) become equivalent (cf.[3, Def 7.3.14]).

3.1 Invertible Terms

The assumption of extensionality is also essential in the investigation of invertibility, as shown in [9,4] in the context of λ-calculus.

Within the theory **CL+ext** that is **CL** extended with the rule (cf [3, Def 7.1.10]):

$$Px = P'x \text{ with } x \notin FV(PP') \text{ implies } P = P',$$

we can characterise the *invertible* combinatory logic terms. We first observe that a semi-group structure on the extended theory **CL+ext** is given by defining a composition of terms by means of the B combinator as

$$X \cdot Y = \mathsf{B}XY$$

as for all Z we get $(X \cdot Y)Z = \mathsf{B}XYZ = X(YZ)$. This operation is associative and can be seen as implementing 'sequential' or 'functional composition'. In the λ-calculus it is defined by

$$M \cdot N = \lambda z.M(Nz)$$

for any two λ-terms M, N.

Moreover, we can take the I combinator as the identity; in the λ-calculus this is given, for example, by the term $\lambda x.x$.

Naturally, the question arises which terms of a calculus like **CL+ext** form a group, i.e. for which terms X we have an element X^{-1} (the inverse) such that

$$X \cdot X^{-1} = X^{-1} \cdot X = \mathsf{I}.$$

The classically invertible **CL** terms are all those terms X for which there is a Y such that $\mathsf{B}XY = \mathsf{B}YX = \mathsf{I}$ holds (cf also [8, Sect 5.D.5 and Def 5.D.1]). A very simple example of an invertible term is the identity combinator I which is its own inverse. In fact, we have that $\mathsf{I} \cdot \mathsf{I} = \mathsf{BII} = \mathsf{I}$. However, in calculi without extensionality this might be about the only example of an invertible term. According to [3, Section 21.3] the invertible terms in the λ-calculus (without extensionality) form the trivial group $\{\mathsf{I}\}$. Extensionality is therefore needed to obtain some non-trivial invertible elements. It allows us to show for example that $\mathsf{C} = \mathsf{C}^{-1}$, i.e. C is its own inverse. This is intuitively clear as the combinator C is essentially representing a transposition of its 2nd and 3rd argument and permutations are reversible. Dezani [9] and Bergstra and Klop [4] have studied the problem of how to describe the invertible elements in different calculi and theories. This also resulted in a description of the group of all invertible elements in the $\lambda\eta$-calculus (cf. [3, Ch 21]).

Contrary to the classical approach we will define a calculus which is reversible in the sense that all reductions in the calculus are invertible. The new reversible calculus will be an extension of the **CL+ext** theory, so that all classical **CL+ext** reductions will still be reductions in the new calculus.

4 Reversible Combinatory Logic

Providing a mechanism to record the computational history of a term allows us to define a reversible version of **CL**, which we will call **rCL**.

Formally, we define a reversible combinatory logic term, or **rCL**-term, as a pair $\langle M \mid H\rangle$, where M is a classical **CL**-term, which we refer to as the *proper term*, and H is a list of elements S which record the reduction steps S (forward execution) and their expansion steps $\overline{S}$ (backward execution). We refer to H as the *history term* and define its syntax by

$$H ::= \varepsilon \mid S : H \mid \overline{S} : H$$
$$S ::= T\mathsf{K}_n^m \mid \mathsf{W}_n^m \mid \mathsf{B}_n^m \mid \mathsf{C}_n^m$$

with T a classical **CL**-term and $n, m \in \mathbb{N}$. We denote by $\mathcal{H}$ be the set of all history terms. The meaning of the two numbers n and m is to record the exact point in the term in which the combinator, i.e. its corresponding reduction rule, is applied, and the length of prefix of the reduced term, respectively. This information is important to guarantee a unique replay of all reduction steps. We will often omit ε and use blank to represent the empty history. We will denote by $S + l$ with $l \in \mathbb{N}$ a history step where the position reference is increased by l, e.g. $T\mathsf{K}_n^m + l = T\mathsf{K}_{n+l}^m$ and by $H + l$ a position shift applied to a whole history, i.e. $H + l = S_1 + l : S_2 + l : \ldots : S_k + l$.

Formally, we define the function *len* on classical **CL**-terms by:

$$len(X) = \begin{cases} 1 & \text{if } X \text{ is a constant or variable} \\ n+m & \text{if } X = (YZ) \text{ with } len(Y) = n \text{ and } len(Z) = m. \end{cases}$$

The reversible (forward) reduction relation on **rCL** is defined by:

(i) $\langle \mathsf{K}XY \mid \rangle \rightarrowtail \langle X \mid Y\mathsf{K}_0^{len(X)}\rangle$, (iii) $\langle \mathsf{C}XYZ \mid \rangle \rightarrowtail \langle XZY \mid \mathsf{C}_0^{len(X)}\rangle$,

(ii) $\langle \mathsf{W}XY \mid \rangle \rightarrowtail \langle XYY \mid \mathsf{W}_0^{len(X)}\rangle$, (iv) $\langle \mathsf{B}XYZ \mid \rangle \rightarrowtail \langle X(YZ) \mid \mathsf{B}_0^{len(X)}\rangle$,

The reversible (backward) reduction relation on **rCL** is defined by:

(i) $\langle X \mid \rangle \rightarrowtail \langle \mathsf{K}XY \mid \overline{Y\mathsf{K}}_0^{len(X)}\rangle$, (iii) $\langle XZY \mid \rangle \rightarrowtail \langle \mathsf{C}XYZ \mid \overline{\mathsf{C}}_0^{len(X)}\rangle$,

(ii) $\langle XYY \mid \rangle \rightarrowtail \langle \mathsf{W}XY \mid \overline{\mathsf{W}}_0^{len(X)}\rangle$, (iv) $\langle X(YZ) \mid \rangle \rightarrowtail \langle \mathsf{B}XYZ \mid \overline{\mathsf{B}}_0^{len(X)}\rangle$,

Additionally the assume the following structural rules:

(i) $\langle X \mid \rangle \rightarrowtail \langle X' \mid H'\rangle$ implies $\langle XY \mid \rangle \rightarrowtail \langle X'Y \mid H'\rangle$,

(ii) $\langle X \mid \rangle \rightarrowtail \langle X' \mid H'\rangle$ implies $\langle YX \mid \rangle \rightarrowtail \langle YX' \mid H' + len(Y)\rangle$,

(iii) $\langle X \mid \rangle \rightarrowtail \langle X' \mid H'\rangle$ implies $\langle X \mid H\rangle \rightarrowtail \langle X' \mid H : H'\rangle$.

(iv) $\langle X \mid \overline{H} : H\rangle \rightarrowtail \langle X \mid \rangle$ and $\langle X \mid H : \overline{H}\rangle \rightarrowtail \langle X \mid \rangle$.

The last two rules allows us to go back to the starting point by reversing the history. For example:

$$\langle \mathsf{W} \mid \rangle \rightarrowtail \langle \mathsf{KWB} \mid \overline{\mathsf{BK}}^1_0 \rangle \rightarrowtail \langle \mathsf{W} \mid \overline{\mathsf{BK}}^1_0 : \mathsf{BK}^1_0 \rangle \rightarrowtail \langle \mathsf{W} \mid \rangle, \text{ and}$$
$$\langle \mathsf{KWB} \mid \rangle \rightarrowtail \langle \mathsf{W} \mid \mathsf{BK}^1_0 \rangle \rightarrowtail \langle \mathsf{KWB} \mid \mathsf{BK}^1_0 : \overline{\mathsf{BK}}^1_0 \rangle \rightarrowtail \langle \mathsf{KWB} \mid \rangle$$

This also shows that the histories $\overline{\mathsf{BK}}^1_0$ and BK^1_0 are (right and left) inverses of each other (cf. Section 4.1). We denote by $\twoheadrightarrow$ the reflexive and transitive closure of $\rightarrowtail$.

Example 4.1 Without the position references the following two terms reduce to the same term:

$$\langle \mathsf{K(CW)C} \mid \rangle \rightarrowtail \langle \mathsf{CW} \mid \mathsf{CK} \rangle \text{ and } \langle \mathsf{KCCW} \mid \rangle \rightarrowtail \langle \mathsf{CW} \mid \mathsf{CK} \rangle$$

It is therefore impossible to tell where $\langle \mathsf{CW} \mid \mathsf{CK} \rangle$ came from. However with position information we have

$$\langle \mathsf{K(CW)C} \mid \rangle \rightarrowtail \langle \mathsf{CW} \mid \mathsf{CK}^2_0 \rangle \text{ and } \langle \mathsf{KCCW} \mid \rangle \rightarrowtail \langle \mathsf{CW} \mid \mathsf{CK}^1_0 \rangle$$

The position information also allows us to encode different reduction strategies ($n = 0$ indicates left-most reduction) as in the following example.

Example 4.2 Let us consider the classical term $\mathsf{W(B}XYZ\mathsf{)K}$. It has two possible reduction paths which are reflected in the history terms:

$$\langle \mathsf{W(B}XYZ\mathsf{)K} \mid \rangle \rightarrowtail \langle (\mathsf{B}XYZ)\mathsf{KK} \mid \mathsf{W}^4_0 \rangle \rightarrowtail \langle (X(YZ))\mathsf{KK} \mid \mathsf{W}^4_0 : \mathsf{B}^1_0 \rangle \text{ and}$$
$$\langle \mathsf{W(B}XYZ\mathsf{)K} \mid \rangle \rightarrowtail \langle (\mathsf{W}(X(YZ))\mathsf{K} \mid \mathsf{B}^1_1 \rangle \rightarrowtail \langle (X(YZ))\mathsf{KK} \mid \mathsf{B}^1_1 : \mathsf{W}^3_0 \rangle$$

Classical combinatory logic can be embedded in **rCL** by representing any **CL**-term M with a **rCL**-term T of the form $\langle M \mid \varepsilon \rangle$. We can show that the weak reduction relation for **CL**-terms can be simulated by the reversible reduction relation on **rCL**. This is implied by the following more general result.

Proposition 4.3 *For every $M \in$ **CL** we have:*

$$M \twoheadrightarrow N \text{ or } N \twoheadrightarrow M \text{ iff } \forall H \in \mathcal{H}\ \exists H' \in \mathcal{H}:\ \langle M \mid H \rangle \twoheadrightarrow \langle N \mid H' \rangle.$$

4.1 The History Group $\mathcal{H}$

For a history $H = S_1 : S_2 : \ldots : S_{n-1} : S_n \in \mathcal{H}$, we define its formal inverse

$$\overline{H} = \overline{S_1 : S_2 : \ldots : S_{n-1} : S_n} = \overline{S_n} : \overline{S_{n-1}} : \ldots : \overline{S_2} : \overline{S_1}$$

with the following properties: (i) $\overline{\overline{H}} = H$ and (ii) $H : \overline{H} = \varepsilon$.

It is easy to see that by construction the set of histories $\mathcal{H}$ forms a group with with respect to the composition operation ":".

The inverse of a history and the inverse of a classical **CL** term, if it exists, are closely related. The inverse history can, to a certain degree, simulate the effects of the inverse term. In order to establish this relation, we first show how the group structure of the history terms interacts with the reversible reduction rules introduced before.

Lemma 4.4 *Let X be a classical* **CL** *term, and let $H \in \mathcal{H}$. Then*

$$\langle X \mid \rangle \rightarrowtail\!\!\!\rightarrow \langle X' \mid H\rangle \textit{ iff } \langle X' \mid \rangle \rightarrowtail\!\!\!\rightarrow \langle X \mid \overline{H}\rangle.$$

Proof. As $\overline{H} : H \equiv H : \overline{H} = \varepsilon$, we have

$$\langle X \mid \overline{H}\rangle \rightarrowtail\!\!\!\rightarrow \langle X' \mid \overline{H} : H\rangle \equiv \langle X' \mid \rangle$$

and thus by the reversible backward reduction rules

$$\langle X' \mid \rangle \rightarrowtail\!\!\!\rightarrow \langle X \mid \overline{H}\rangle.$$

□

We can now show that for classical invertible terms M, histories can be used to simulate a reduction for the inverse M^{-1} given a reduction for M.

Proposition 4.5 *Let M be an invertible term in* **CL***. Given a history $H \in \mathcal{H}$ and two* **CL** *terms N_1 and N_2 such that*

$$\langle MN_1 \mid \rangle \rightarrowtail\!\!\!\rightarrow \langle N_2 \mid H\rangle.$$

Then there exist $H', H'' \in \mathcal{H}$ such that

$$\langle M^{-1}N_2 \mid H''\rangle \rightarrowtail\!\!\!\rightarrow \langle N_1 \mid H'\rangle.$$

Proof. By Lemma 4.4 and the hypothesis we have that $\langle N_2 \mid \rangle \rightarrowtail\!\!\!\rightarrow \langle MN_1 \mid \overline{H}\rangle$. Using the history $\overline{H}$ and again Lemma 4.4, we get

$$\langle M^{-1}N_2 \mid \overline{H} + \mathit{len}(M^{-1})\rangle \rightarrowtail\!\!\!\rightarrow \langle M^{-1}MN_1 \mid \rangle.$$

Therefore

$$\langle M^{-1}N_2 \mid \overline{H} + \mathit{len}(M^{-1} + 1 : \mathsf{B}_0^{\mathit{len}(M^{-1}}\rangle \rightarrowtail \langle \mathsf{B}M^{-1}N_2 \mid \overline{H} + \mathit{len}(M^{-1} + 1\rangle$$
$$\rightarrowtail\!\!\!\rightarrow \langle \mathsf{B}M^{-1}MN_1 \mid \rangle \stackrel{\text{def}}{=} \langle (M^{-1} \cdot M)N_1 \mid \rangle \rightarrowtail \langle N_1 \mid H'\rangle.$$

□

4.2 The Groupoid of Reversible Computations

Given a group G and a set X, a *group action* of G on X is defined as a homomorphism π of G into the automorphism group of X, i.e. $\pi(g) \in \mathrm{Aut}(X)$ such that $\pi(e) = \mathrm{id}$, and $\pi(gh)(x) = \pi(g)(\pi(h)(x))$. Given a group action π of G on X we can define a groupoid $\mathcal{G} = \mathcal{G}(X, G, \pi)$ as follows:

- $\mathcal{G} \subseteq X \times G \times X$ with $(x, g, y) \in \mathcal{G}$ iff $\pi(g)(x) = y$.
- $\mathcal{B} = X$
- $\alpha((x, g, y)) = x$ and $\beta((x, g, y)) = y$
- $(x, g, y) \cdot (y, h, z) = (x, hg, z)$
- $\lambda_{(x,g,y)} = (x, e, x)$ and $\varrho_{(x,g,y)} = (y, e, y)$
- $(x, g, y)^{-1} = (y, g^{-1}, x)$.

We show that the set of reversible computations is the groupoid defined by the action of the history group $\mathcal{H}$ on the set of **rCL** terms. Intuitively, this

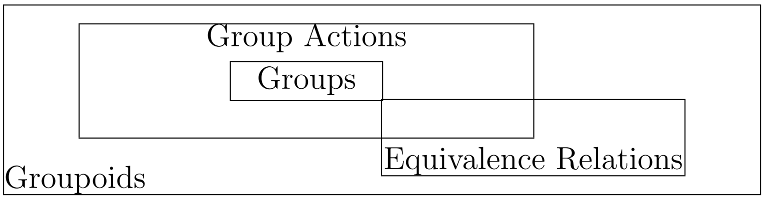

Fig. 1. Groups, Group Actions and Equivalence Relations

means that each history term determines a permutation on **rCL** corresponding to a reversible computation, and vice versa.

Consider the groupoid $\mathcal{G}$ defined by the action π of $\mathcal{H}$ on **rCL** given by

$$\pi(H)(\langle M \mid H' \rangle) = \langle M \mid H' : H \rangle$$

The computational groupoid $\mathcal{G}(\mathbf{rCL}, \rightarrowtail\!\!\twoheadrightarrow)$ constructed as in Section 2 using the transition relation $\rightarrowtail\!\!\twoheadrightarrow$ on **rCL** terms, and the group action groupoid $\mathcal{G}(\mathbf{rCL}, \mathcal{H}, \pi)$ defined above are isomorphic. The isomorphism is given by simply forgetting about the "connecting history".

Proposition 4.6 *The map* $\delta : \mathcal{G}(\mathbf{rCL}, \mathcal{H}, \pi) \rightarrow \mathcal{G}(\mathbf{rCL}, \rightarrowtail\!\!\twoheadrightarrow)$ *which is defined as* $\delta(\langle T, H, T' \rangle) = \langle T, T' \rangle$*, i.e.* $\delta = (\alpha, \beta)$*, is a groupoid isomorphism.*

5 Conclusion

We have introduced a reversible version **rCL** of Combinatory Logic where terms are enriched with a "history" part which allows us to uniquely "replay" every computational step. We have utilised the structure of a *groupoid* to model computation in **rCL**.

Groupoids can bee seen as a generalisation of several mathematical structures, such as *groups*, *group actions* and *equivalence relations*, as shown in Figure 1 (cf. [12]). The last two structures are particularly relevant for our treatment of **rCL**. In fact, the computational paths of a reversible calculus can be seen as the orbits of a group acting on some space, in our case the history group acting on the space of **rCL** terms. On the other hand, the equational theory of a calculus introduces an equivalence relation on the terms. Groupoids are therefore able to accommodate the operational semantics as well as the equational theory of **rCL**.

Further work will concentrate on constructing a denotational semantics for **rCL** based on the groupoid structure presented here. Our aim is in a compositional definition of transition operators which serves as a basis for semantics-based analysis techniques for the λ-calculus. For this we hope to exploit well-established results on the relation between operator algebras (in particular C^* algebras) and groupoids [13]. Furthermore, we believe that reversible combinatory logic can in principle be used for a (maybe highly inefficient) translation of classical into quantum computation.

References

[1] Abramsky, S., *A structural approach to reversible computation*, in: *Proceedings of LCCS 2001*, 2001, pp. 1–16.

[2] Abramsky, S., E. Haghverdi and P. Scott, *Geometry of interaction and linear combinatory algebras*, Mathematical Structures in Computer Science **12** (2002), pp. 625–665.

[3] Barendregt, H. P., "The Lambda Calculus," Studies in Logic and the Foundations of Mathematics **103**, North-Holland, 1984, revised edition.

[4] Bergstra, J. and J. W. Klop, *Invertible terms in the lambda calculus*, Theoretical Computer Science **11** (1980), pp. 19–37.

[5] Brandt, W., *Über eine Verallgemeinerung des Gruppengriffes*, Mathematische Annalen **96** (1926), pp. 360–366.

[6] Brown, R., *From groups to groupoids: a brief survey*, Bull. London Math. Soc. **19** (1987), pp. 113–134.

[7] Connes, A., "Noncommutative Geometry," Academic Press, San Diego, 1994.

[8] Curry, H. B. and R. Feys, "Combinatory Logic," North-Holland, 1958.

[9] Dezani-Ciancaglini, M., *Characterization of normal forms possesing inverse in the* $\lambda - \beta - \eta$*-calculus*, Theoretical Computer Science **2** (1976), pp. 323–337.

[10] Di Pierro, A., C. Hankin and H. Wiklicky, *Probabilistic lambda-calculus and quantitative program analysis*, Journal of Logic and Computation **15** (2005), pp. 159–179.

[11] Mundici, D. and W. Sieg, *Paper machines*, Philosophica Mathematica **Series III, 3** (1995), pp. 5–30.

[12] Ramsay, A. and J. Renault, editors, "Groupoids in Analyis, Geometry, and Physics," Contemporary Mathematics **282**, AMS, Providence, RI, 2001.

[13] Renault, J., "A Groupoid Approach to C*-Algebras," Lecture Notes in Mathematics **793**, Springer Verlag, Berlin – Heidelberg – New York, 1980.

[14] van Tonder, A., *A lambda calculus for quantum computation*, SIAM Journal of Computation **33** (2004), pp. 1109–1135.

[15] Vitanyi, P., *Time, space, and energy in reversible computing*, in: *Proceedings of the ACM International Conference on Computing Frontiers*, 2005.

[16] Weinstein, A., *Groupoids: Unifying internal and external symmetry*, Notices of the AMS **43** (1996), pp. 744–752.