Under consideration for publication in Math. Struct. in Comp. Science

Reversible Combinatory Logic

Alessandra Di Pierro[1], Chris Hankin[2] and Herbert Wiklicky[2†]

[1] *Dipartimento di Informatica, Universitá di Pisa*
Largo Bruno Pontecorvo, 3, 56127 Pisa, Italy
dipierro@di.unipi.it
[2] *Department of Computing, Imperial College London*
180 Queen's Gate, London SW7 2AZ, United Kingdom
{clh,herbert}@doc.ic.ac.uk

Received 1 December 2005; Revised 23 March 2006

The λ-calculus is destructive: its main computational mechanism – beta reduction – destroys the redex and makes it thus impossible to replay the computational steps. Combinatory logic is a variant of the λ-calculus which maintains irreversibility. 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. (Mundici and Sieg, 1995), 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 (Kitaev et al., 2000; Nielsen and Chuang, 2000), various issues in systems biology (Danos and Krivine, 2004; Phillips and Ulidowski, 2005), and the need for minimal energy loss (Vitanyi, 2005) make reversible computation once again interesting. Quantum Computation has been the motivation for van Tonder (van Tonder, 2004) who presents a reversible applied lambda calculus with quantum constants; his operational semantics provided the inspiration for the operational semantics of our reversible version of Combinatory Logic. On the other hand, the set of combinators that we consider here have also been studied by Abramsky (Abramsky, 2001;

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

Abramsky et al., 2002), 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 (Di Pierro et al., 2005). 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.

In this paper we introduce an extension of the classical Combinatory Logic **CL** and its associated notion of reduction, which incorporates information useful to reconstruct the redex from the contractum. Intuitively, we can reverse a computation if we keep information about its 'history', i.e. information about the reduction steps that have been performed during the computation. Our extension is based on this intuition and in particular on the notion of a 'history term', which consists in a sequence of variables and appropriately annotated combinators. This notion and in general the information that has to be recorded as a history is strictly dependent on the nature and structure of the original calculus; for example, in Van Tonder's λ-calculus (van Tonder, 2004) the history keeps track only of the substitutions which take place in each β-reduction step.

Reversibility naturally introduces a notion of symmetry into computation and is therefore strongly related to the theory of groups; these are considered by most mathematicians as being virtually synonymous of symmetry (Weinstein, 1996). However, the notion of automorphism associated to group is in some sense too "trivial" to characterise the symmetry involved in a reversible computation. According to R. Brown (Brown, 1987), this has motivated the extension of the theory of groups to the theory of *groupoids*. A groupoid can be informally described as a group with many objects, where objects can be thought of as the start and end points of computational processes. While group theory only allows us to characterise processes which start from one point and (possibly after a number of steps) come back to the same point, 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. Thus, the algebraic structure of groupoids naturally reflects the structure of reversible processes which may traverse a number of states, and is therefore more suitable for our purposes.

Based on this idea, we show that computations in the reversible **CL** can be modelled as elements of the groupoid associated to the reduction relation. This corresponds to the action of a group on the set of reversible terms, the group being determined by the history terms.

This last characterisation allows us to show that the reversible **CL** is universal for classical reversible computation, in the sense that all reversible computations can be represented as a reversible **CL** reduction and vice versa. Moreover, as every reversible reduction corresponds to a permutation, that is a unitary operator, reversible **CL** represents a high-level, though extremely inefficient way, to embed classical (irreversible) computation in quantum computation.

2. Combinatory Logic

Combinatory Logic (Curry and Feys, 1958; Hindley et al., 1972; Hindley and Seldin, 1986) (**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 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:

1 all variables and constants are **CL**-terms,
2 if X and Y are **CL**-terms, then (XY) is a **CL** term.

Following Barendregt (Barendregt, 1984), in the following we will use the symbol $\equiv$ to denote syntactic equivalence. 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 could be λ-defined as follows (Curry and Feys, 1958, p379):

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

Importantly, we can use B, W and C to implement the common combinator S (Curry and Feys, 1958, p155):

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

In order to generate equalities provable in this calculus we use a notion of reduction similar to the *weak reduction* for the SK-calculus (Barendregt, 1984). This is defined as the smallest extension of the relation on **CL**-terms induced by the basic operators which is compatible with application.

Definition 2 (Reduction in CL). The reduction relation $\longrightarrow$ on **CL**-terms is defined by the following rules:

1 $\mathsf{K}XY \longrightarrow X$,
2 $\mathsf{W}XY \longrightarrow XYY$,
3 $\mathsf{C}XYZ \longrightarrow XZY$,
4 $\mathsf{B}XYZ \longrightarrow X(YZ)$,
5 $X \longrightarrow X'$ implies $XY \longrightarrow X'Y$,
6 $X \longrightarrow X'$ implies $YX \longrightarrow YX'$,

We will denote by $\twoheadrightarrow$ the reflexive transitive closure of $\rightarrow$. Following Barendregt (Barendregt, 1984) we will denote by $=$ the least equivalence relation extending $\twoheadrightarrow$. This relation coincides with the set of all equalities which are provable in **CL** (Barendregt, 1984, Prop 7.2.2).

The relation between the λ-calculus and **CL** is a standard result (Barendregt, 1984). 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 well known that in presence of a rule for extensionality the two theories λ-calculus and **CL** (which are in general not equivalent) become equivalent (Barendregt, 1984, Def 7.3.14).

2.1. *Invertible Terms*

The assumption of extensionality is also essential in the investigation of invertibility, as shown in (Dezani-Ciancaglini, 1976; Bergstra and Klop, 1980) in the context of λ-calculus.

Within the theory **CL+ext** that is **CL** extended with the rule (Barendregt, 1984, Def 7.1.10):

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

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 can be defined, 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 (Curry and Feys, 1958, 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

$$\mathsf{I} \cdot \mathsf{I} = \mathsf{B}\mathsf{I}\mathsf{I} = \mathsf{I}.$$

However, in calculi without extensionality this might be about the only example of an invertible term. According to (Barendregt, 1984, 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 (Dezani-Ciancaglini, 1976) and Bergstra and Klop (Bergstra and Klop, 1980) 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. (Barendregt, 1984, Ch 21).

Contrary to the classical approach we will define a calculus which is reversible in the sense that all reductions in the calculus can be expanded in a unique way to get the same derivation but in the opposite direction. 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.

3. Reversible Combinatory Logic

Providing a mechanism for recording 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 which record the reduction steps S (forward execution) and their expansion steps $\overline{S}$ (backward execution). We refer to H as the *history term*.

Definition 3 (Reversible Combinatory Logic Terms). A term in **rCL** is a pair $\langle M \mid H\rangle$, where M is a classical **CL** term and H has the following syntax:

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

where T is a classical **CL**-term, $n, m \in \mathbb{N}$ and $\overline{\overline{S}}$ is defined as S.

If $H \equiv S_1 : S_2 : \ldots : S_n$ then we will denote by $\overline{H}$ the term $\overline{S_n} : \overline{S_{n-1}} : \ldots : \overline{S_1}$. We identify the terms $\overline{\overline{H}}$ and H, i.e. $\overline{\overline{H}} = H$. We denote by $\mathcal{H}$ the set of all history terms modulo this equivalence. It is easy to see that by construction the set of histories $\mathcal{H}$ forms a group with respect to the composition operation ":" by defining the neutral element of the group as the empty history ε and the inverse of H by $\overline{H}$, i.e. $H : \overline{H} = \overline{H} : H = \varepsilon$. 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. $T\mathsf{K}_n^m + l \equiv T\mathsf{K}_{n+l}^m$ and by $H + l$ a position shift applied to a whole history, i.e. $H + l \equiv S_1 + l : S_2 + l : \ldots : S_k + l$.

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

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

Definition 4 (Reduction in rCL). The reversible reduction relation $\rightarrowtail$ is defined by the following rules

Forward rules

1 $\langle \mathsf{K}XY \mid \rangle \rightarrowtail \langle X \mid Y\mathsf{K}_0^{len(X)} \rangle$,

2 $\langle \mathsf{W}XY \mid \rangle \rightarrowtail \langle XYY \mid \mathsf{W}_0^{len(X)} \rangle$,

3 $\langle \mathsf{C}XYZ \mid \rangle \rightarrowtail \langle XZY \mid \mathsf{C}_0^{len(X)} \rangle$,

4 $\langle \mathsf{B}XYZ \mid \rangle \rightarrowtail \langle X(YZ) \mid \mathsf{B}_0^{len(X)} \rangle$,

Backward rules

1 $\langle X \mid \rangle \rightarrowtail \langle \mathsf{K}XY \mid \overline{Y\mathsf{K}}_0^{len(X)} \rangle$,

2 $\langle XYY \mid \rangle \rightarrowtail \langle \mathsf{W}XY \mid \overline{\mathsf{W}}_0^{len(X)} \rangle$,

3 $\langle XZY \mid \rangle \rightarrowtail \langle \mathsf{C}XYZ \mid \overline{\mathsf{C}}_0^{len(X)} \rangle$,

4 $\langle X(YZ) \mid \rangle \rightarrowtail \langle \mathsf{B}XYZ \mid \overline{\mathsf{B}}_0^{len(X)} \rangle$,

Structural rules

1 $\langle X \mid \rangle \rightarrowtail \langle X' \mid H' \rangle$ implies $\langle XY \mid \rangle \rightarrowtail \langle X'Y \mid H' \rangle$,

2 $\langle X \mid \rangle \rightarrowtail \langle X' \mid H' \rangle$ implies $\langle YX \mid \rangle \rightarrowtail \langle YX' \mid H' + len(Y) \rangle$,

3 $\langle X \mid \rangle \rightarrowtail \langle X' \mid H' \rangle$ implies $\langle X \mid H \rangle \rightarrowtail \langle X' \mid H : H' \rangle$.

We will refer to the backward rule i as the symmetric of the forward rule i and vice versa, for $i = 1, 2, 3, 4$. The structural rules guarantee the compatibility of the reduction relation with the composition operation on the proper terms (rules 1 and 2), and with the composition of history terms (rule 3).

We call the relation $\rightarrowtail$ on **rCL** defined by the forward, backward and structural rules in Definition 4 the *forward reduction*, and we denote by $\twoheadrightarrow$ the reflexive and transitive closure of $\rightarrowtail$. The relation $\rightarrowtail$ is a proper relation, i.e. not a function; reductions in **rCL** are therefore *non-deterministic*. However, the converse transition relation $\leftarrowtail$, defined as

$$\langle P_2 \mid H_2 \rangle \leftarrowtail \langle P_1 \mid H_1 \rangle \text{ iff } \langle P_1 \mid H_1 \rangle \rightarrowtail \langle P_2 \mid H_2 \rangle,$$

and referred to as *backward reduction*, is *deterministic*. This allows us to reconstruct the reduction sequences uniquely despite the non-deterministic nature of $\rightarrowtail$. We will formally show this in Proposition 1 whose proof will highlight the fundamental role played by the pair of integers (m, n) occurring in the history terms in making the backward reduction deterministic. Since $\leftarrowtail$ is defined only on those terms $\langle P_2 \mid H_2 \rangle$ for which there is a forward reduction, it is a partial function.

In the following we will use interchangeably the words 'reduction', 'reduction sequence', 'computational path', 'computation'.

Proposition 1. The relation $\leftarrowtail$ is a partial function.

Proof. We have to show for every **rCL** term $\langle P_2 \mid H_2 \rangle$: either there is no term $\langle P_1 \mid H_1 \rangle$ different from $\langle P_2 \mid H_2 \rangle$ with $\langle P_2 \mid H_2 \rangle \leftarrowtail \langle P_1 \mid H_1 \rangle$ or if there exists such a term $\langle P_1 \mid H_1 \rangle$, then the term is uniquely determined.

Suppose that $\langle P_2 \mid H_2\rangle \leftarrowtail \langle P_1 \mid H_1\rangle$. Then by definition of $\leftarrowtail$ the structure of $\langle P_1 \mid H_1\rangle$ is determined in a unique way by the last step S in the history $H_2 \equiv H_2' : S$. We show this by analysing the structure of S. By Definition 3 this can be of one of the following forms:

$$T\mathsf{K}^m_n \text{ or } \mathsf{W}^m_n \text{ or } \mathsf{B}^m_n \text{ or } \mathsf{C}^m_n,$$

reflecting the four forward rules of Definition 4, or

$$\overline{T\mathsf{K}^m_n} \text{ or } \overline{\mathsf{W}^m_n} \text{ or } \overline{\mathsf{B}^m_n} \text{ or } \overline{\mathsf{C}^m_n},$$

reflecting the four backward rules of Definition 4.

We will show only one case as in all the other seven cases the structure of the proof is exactly the same and can be easily reproduced.

Consider the case $S \equiv T\mathsf{K}^m_n$.

Assume that $H_2 \equiv H_2' : T\mathsf{K}^m_n$ for some (possibly empty) H_2', then the reduction $\langle P_1 \mid H_1\rangle \rightarrowtail \langle P_2 \mid H_2\rangle$ consists in an application of the forward rule 1. This means that the classical term P_2 has the form $P_2' X P_2''$, where P_2' and P_2'' can be omitted, $m = len(X)$ and $n = len(P_2')$, or if P_2' has been omitted $n = 0$. This forces the term $\langle P_1 \mid H_1\rangle$ to have the structure

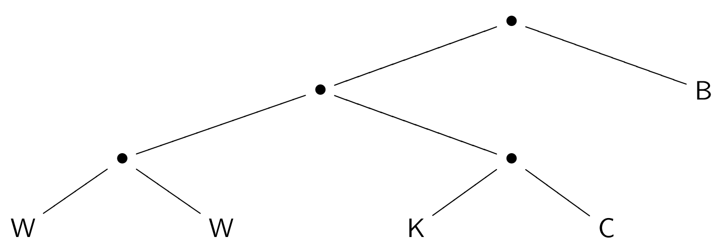
$$P_1 \equiv P_2'(\mathsf{K}XT)P_2'' \text{ and } H_1 \equiv H_2'.$$

Note that the term P_2'' can occur because of structural rule 1. Moreover, the case where P_2' is present corresponds to applying structural rule 2 starting from the basic axiom with $n = 0$ reflecting the basic case where no position shift is needed (or equivalently P_2' is missing). Finally, the case where H_2' is not empty is covered by structural rule 3.

□

If we represent classical **CL** terms as binary trees then the last step S of the history $H_2' : S$ specifies a simple tree transformation. This transformation together with removing S from the history implements the reverse relation $\leftarrowtail$. To do this we first have to decompose the tree representation of P_2 according to the information provided by the sub- and superscripts m and n and then transform the tree.

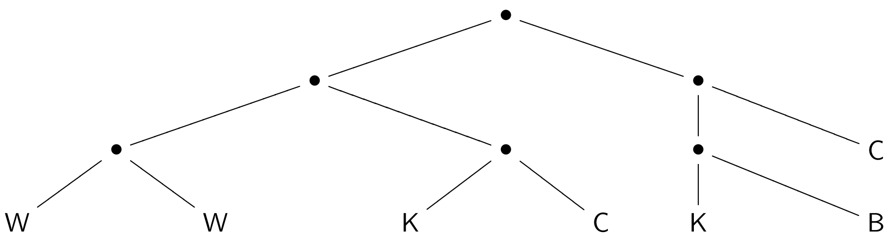
Example 1. Consider the classical term $P_2 \equiv (((\mathsf{WW})(\mathsf{KC}))\mathsf{B})$. This term can be represented by the binary tree:



Let us assume that the history contains only one step, e.g. $H_2 \equiv \mathsf{CK}^1_4$, then we can (re)construct the sub-tree corresponding to P_2' by isolating the sub-tree on the first

4 leaves, and the one corresponding to X as the (degenerate) sub-tree with just one leaf, such that $P_2 = P'_2 X P''_2$: in our case we get the sub-trees corresponding to $P'_2 \equiv ((\mathsf{WW})(\mathsf{KC}))$ and $X \equiv \mathsf{B}$ while P''_2 is omitted. Based on this decomposition we can (re)construct the tree $P_1 \equiv P'_2(\mathsf{K}XT)P''_2 \equiv (((\mathsf{WW})(\mathsf{KC}))(\mathsf{KBC}))$.

In terms of tree-transformations, we just have to replace the subtree $X \equiv \mathsf{B}$ by the subtree representing $((\mathsf{K}X)T) \equiv (\mathsf{KBC})$;



It is important to note that the decomposition of P_2 according to the data provided by n and m might not always be possible. If we had taken, for example, $H_2 \equiv \mathsf{CK}^1_3$ it would have been impossible to isolate a complete sub-tree with the first three leaves W, W and K. This would represent a case where the reverse computation is not defined.

An important property of **rCL** is that reductions always increase the length of the history term. We can interpret accumulation of steps in the history as reflecting the *progress of time*.

Proposition 2. Let $T = \langle M \mid H' \rangle$ and $T' = \langle N \mid H'' \rangle$ be two **rCL** terms. If $T \rightarrowtail\!\!\!\twoheadrightarrow T'$, then there exists $H \in \mathcal{H}$ such that $H'' = H' : H$.

Proof. Straightforward by induction on the length of the reduction $T \rightarrowtail\!\!\!\twoheadrightarrow T'$. □

The identification of histories in $\mathcal{H}$ via the equivalence $H : \overline{H} = \varepsilon$ allows us to eliminate "computational loops", that is the cyclic application of a certain sequence of rules. In this way we can return to the start of a computation by undoing all its steps (in reverse order) as in the following two simple reductions:

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

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

The following example shows that without the position references it would be impossible to reconstruct or retrace a given computational path.

Example 2. Consider the two reductions for terms $\langle \mathsf{K(CW)C} \mid \rangle$ and $\langle \mathsf{KCCW} \mid \rangle$, respectively

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

It is therefore impossible to tell where $\langle \mathsf{CW} \mid \mathsf{CK} \rangle$ came from. However, by adding the 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 (e.g. $n = 0$ indicates left-most reduction) as in the following example.

Example 3. Let us consider the classical term $\mathsf{W}(\mathsf{B}XYZ)\mathsf{K}$. It has two possible reduction paths which are reflected in the history terms:

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

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

Note that fixing a strategy in a reduction effectively rules out the use of the structural rule 2 in the reduction.

The retracing of a computational path, i.e. the reverse reduction relation $\leftarrowtail$, is naturally implemented within the transition relation $\rightarrowtail$.

Lemma 1. If $\langle P_2 \mid H_2 \rangle \leftarrowtail \langle P_1 \mid H_1 \rangle$ then there exists a history H with $H \equiv H_1$ such that $\langle P_2 \mid H_2 \rangle \rightarrowtail\!\!\!\!\rightarrow \langle P_1 \mid H \rangle$.

Proof. If $\langle P_2 \mid H_2 \rangle \leftarrowtail \langle P_1 \mid H_1 \rangle$ then $\langle P_1 \mid H_1 \rangle \rightarrowtail \langle P_2 \mid H_2 \rangle$ (by definition of the converse relation). Assume that $H_1 = S_1 : \ldots : S_{n-1}$ and $H_2 = S_1 : \ldots : S_{n-1} : S_n$. Take $H = S_1 : \ldots : S_{n-1} : S_n : \overline{S}_n$, then

$$\langle P_2 \mid S_1 : \ldots : S_{n-1} : S_n \rangle \rightarrowtail \langle P_1 \mid S_1 : \ldots : S_{n-1} : S_n : \overline{S}_n \rangle$$

as for every forward rule there is a symmetric backward rule in Definition 4, and obviously $S_1 : \ldots : S_{n-1} \equiv S_1 : \ldots : S_{n-1} : S_n : \overline{S}_n$. □

3.1. *Embedding* **CL** *in* **rCL**

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

Proposition 3. For every $M \in$ **CL** we have:

If $M \twoheadrightarrow N$ then for all $H \in \mathcal{H}$ there exists $H' \in \mathcal{H} : \langle M \mid H \rangle \rightarrowtail\!\!\!\!\rightarrow \langle N \mid H' \rangle$.

Proof. By hypothesis there exists a classical reduction

$$M = N_0 \longrightarrow N_1 \ldots \longrightarrow N_i \ldots \longrightarrow N_n = N$$

for some $n \geq 1$, where for all $0 \leq i \leq n$, $N_i \longrightarrow N_{i+1}$ is an instance of one of the rules $1-4$ of Definition 2.

By replacing each reduction step by the corresponding reversible forward reduction step obtained by the rules in Definition 4 we get

$$\langle M \mid \ \rangle \rightarrowtail\!\!\!\!\rightarrow \langle N \mid H'' \rangle,$$

with H'' the history term produced in the reversible forward reduction. For any $H \in \mathcal{H}$

we can now apply the structural rule 3 in Definition 4 to get

$$\langle M \mid H\rangle \rightarrowtail\!\!\!\twoheadrightarrow \langle N \mid H : H''\rangle.$$

Then take $H' = H : H''$. □

The reverse of the proposition above does not hold, as shown by the following example.

Example 4. Consider the **CL** term $M = \mathsf{KCBBB}$ and its corresponding **rCL** term $\langle \mathsf{KCBBB} \mid \rangle$. The following is a possible reversible reduction for this term:

$$\langle \mathsf{KCBBB} \mid \rangle \rightarrowtail \langle \mathsf{CBB} \mid \mathsf{BK}^1_0\rangle \rightarrowtail \langle \mathsf{WCB} \mid \mathsf{BK}^1_0 : \overline{\mathsf{W}}^1_0\rangle$$

The first step is a forward rule 1 the second step is by backward rule 2. We therefore have the situation:

$$\langle M \mid H\rangle \rightarrowtail\!\!\!\twoheadrightarrow \langle N \mid H'\rangle$$

with $M = \mathsf{KCBBB}$, $N = \mathsf{WCB}$, $H = \varepsilon$ and $H' = \mathsf{BK}^1_0 : \overline{\mathsf{W}}^1_0$.

However, neither of the two classical reductions

$$\mathsf{KCBBB} \twoheadrightarrow \mathsf{WCB} \text{ or } \mathsf{WCB} \twoheadrightarrow \mathsf{KCBBB}$$

are possible as classically the two terms KCBBB and WCB reduce as follows:

$$\mathsf{KCBBB} \rightarrow \mathsf{CBB}$$

and

$$\mathsf{WCB} \rightarrow \mathsf{CBB}.$$

3.2. *Invertible* **CL** *terms and* **rCL** *reduction*

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 of a classical 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 2. Let X be a classical **CL** term, and let $H \in \mathcal{H}$. Then

$$\langle X \mid \rangle \rightarrowtail\!\!\!\twoheadrightarrow \langle X' \mid H\rangle \text{ iff } \langle X' \mid \rangle \rightarrowtail\!\!\!\twoheadrightarrow \langle X \mid \overline{H}\rangle.$$

Proof. Provided that $\langle X \mid \rangle \rightarrowtail\!\!\!\twoheadrightarrow \langle X' \mid H\rangle$ we have by structural rule 3

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

and thus by replacing in this derivation each rule by its symmetric rule

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

□

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. 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

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

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

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

Proof. By Lemma 2 and the hypothesis $\langle MN_1 \mid \rangle \rightarrowtail\!\!\twoheadrightarrow \langle N_2 \mid H \rangle$ we have

$$\langle N_2 \mid \rangle \rightarrowtail\!\!\twoheadrightarrow \langle MN_1 \mid \overline{H} \rangle.$$

By structural rule 2 this reduction holds in any context, e.g. M^{-1}, and by applying backward rule 4 and structural rule 3 we get:

$$\begin{array}{rcl} \langle M^{-1}N_2 \mid \rangle & \rightarrowtail\!\!\twoheadrightarrow & \langle M^{-1}(MN_1) \mid \overline{H} + len(M^{-1}) \rangle \\ & \rightarrowtail\!\!\twoheadrightarrow & \langle \mathsf{B}M^{-1}MN_1 \mid \overline{H} + len(M^{-1}) + 1 : \overline{\mathsf{B}}_0^1 \rangle \\ & = & \langle (M^{-1} \cdot M)N_1 \mid \overline{H} + len(M^{-1}) + 1 : \overline{\mathsf{B}}_0^1 \rangle \\ & = & \langle \mathsf{I}N_1 \mid \overline{H} + len(M^{-1}) + 1 : \overline{\mathsf{B}}_0^1 \rangle \\ & \rightarrowtail\!\!\twoheadrightarrow & \langle N_1 \mid H' \rangle, \end{array}$$

where the last reduction is by Proposition 3. □

In this proof the existence of H' is guaranteed by Proposition 3 which allows us to translate classical equivalence into reversible equivalence: From $P = P'$ and $PM \twoheadrightarrow N$, we can conclude classically – by exploiting extensionality – $P'M \twoheadrightarrow N$ for any M. Thanks to Proposition 3 this can be translated into a similar statement in **rCL**: $\langle PM \mid \rangle \rightarrowtail\!\!\twoheadrightarrow \langle N \mid H \rangle$ implies that there exists H' such that for any $P = P'$ and any M, $\langle P'M \mid \rangle \rightarrowtail\!\!\twoheadrightarrow \langle N \mid H' \rangle$. Thus in the proof above, H' can be constructed as in the proof for Proposition 3.

Note that since there are many different representations of the identity, e.g. $\mathsf{I} \equiv \mathsf{WK}$ or $\mathsf{I} \equiv \mathsf{SKK} \equiv \mathsf{B(B(BW)C)(BB)KK}$, the derivations of $\langle \mathsf{WK}M \mid \rangle$ and $\langle \mathsf{B(B(BW)C)(BB)KK}M \mid \rangle$ will result in **rCL** terms $\langle M \mid H \rangle$ and $\langle M \mid H' \rangle$ with the same proper term M but with completely different histories H and H'. We therefore can say nothing about the concrete nature of H' in the previous proposition.

4. The Groupoid Structure of Reversible Computations

A *groupoid* can be succinctly defined as a small category in which every morphism is an isomorphism (Brown, 1987). This algebraic structure introduced by Brandt (Brandt, 1926) (for further details see e.g. (Renault, 1980; Weinstein, 1996; Ramsay and Renault, 2001; Brown, 1987)) naturally reflects the operational meaning of term reduction and its reverse process. In fact, the reduction relation $\rightarrowtail\!\!\twoheadrightarrow$ defines a reversible computation as an isomorphism between **rCL** terms.

In this section we will develop this analogy in full detail. We will refer to a definition of groupoid as in (Brown, 1987).

Definition 5. 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 (product) $(g, h) \mapsto g \cdot h = gh$, and a function i from $\mathcal{B}$ to $\mathcal{G}$ satisfying the following conditions:

1 gh is defined whenever $\beta(g) = \alpha(h)$, and in this case $\alpha(gh) = \alpha(g)$ and $\beta(gh) = \beta(h)$.
2 The product is associative: if gh and hk are defined then so are $(gh)k$ and $g(hk)$ and they are equal.
3 For each $b \in \mathcal{B}$, $i(b)$ is the identity morphism: $\alpha(i(b)) = \beta(i(b)) = b$.
4 Each $g \in \mathcal{G}$ has an inverse g^{-1} satisfying $g^{-1}g = i(\beta(g))$, $gg^{-1} = i(\alpha(g))$, $\alpha(g^{-1}) = \beta(g)$ and $\beta(g^{-1}) = \alpha(g)$.

An element $g \in \mathcal{G}$ is often written as an arrow $g : \alpha(g) \to \beta(g)$.

Groups are particular cases of groupoids, namely those where the base $\mathcal{B}$ contains only a single element. In this case, we get a universal identity, left and right inverse of any $g \in \mathcal{G}$ coincide, and the composition is defined for any two elements g and h.

Example 5 (Groups). Any group $(G, \bullet)$ with identity e and typical elements g, h etc. defines a groupoid $\mathcal{G}$ in the following way: Take $\mathcal{G} = G$ and as base any one element set $\mathcal{B} = \{*\}$; define $\alpha(g) = *$ and $\beta(g) = *$ for all $g \in G$. The group operation "$\bullet$" is translated in the obvious way into the groupoid operation "$\cdot$" via $g \cdot h = g \bullet h$. In particular, composition is defined in this situation for any two elements g and h in $\mathcal{G} = G$ as $\alpha(g) = * = \beta(h)$. We also get a universal identity $e \in \mathcal{G} = G$, and the inverse $g^{-1} \in \mathcal{G}$ and $g^{-1} \in G$ coincide.

The prototypical example of a groupoid which is not a group is the 'path space' groupoid.

Example 6 (Paths). Consider any (finite) directed graph $\Gamma = (E, V)$ and denote by $s(e)$ and $d(e)$ the source and the destination vertex of an edge $e \in E$. Let P be the set of finite paths on Γ, i.e. the finite sequences $\pi = e_0 e_1 \ldots e_n$ of edges $e_i \in E$ such that two successive edges share a common vertex, i.e. $d(e_i) = s(e_{i+1})$ for $i = 0, \ldots, n-1$. We denote the path of length zero with ε. We interpret an *undirected* graph as a directed graph where every edge $e \in E$ also has a *reverse* edge $e^* \in E$ such that $s(e) = d(e^*)$ and $d(e) = s(e^*)$. We call an edge $e_v \in E$ with $s(e) = d(e) = v$ a *self-loop*. We furthermore define an equivalence relation between paths which have the same start and end points, i.e. $\pi_1 \sim \pi_2$ with $\pi_1 = e^1_0 e^1_1 \ldots e^1_n$ and $\pi_2 = e^2_0 e^2_1 \ldots e^2_m$ iff $s(e^1_0) = s(e^2_0)$ and $d(e^1_n) = d(e^2_m)$. As usual we denote the set of equivalence classes by $P/_\sim$.

We can then define a groupoid structure on the equivalence classes of paths on any undirected graph Γ as follows: Take $\mathcal{G} = P/_\sim$ and $\mathcal{B} = V$, i.e. all vertices of Γ. Furthermore, define $\alpha(\pi) = s(e_0)$ and $\beta(\pi) = d(e_n)$ for any path $\pi = e_0 e_1 \ldots e_n \in P$. We can "compose" any two paths $\pi_1 = e^1_0 e^1_1 \ldots e^1_n$ and $\pi_2 = e^2_0 e^2_1 \ldots e^2_m$ if and only if the ending and beginning match, i.e. iff $\beta(\pi_1) = d(e^1_n) = s(e^2_0) = \alpha(\pi_2)$; in which case we obtain the path $\pi_1 \cdot \pi_2 = e^1_0 e^1_1 \ldots e^1_n e^2_0 e^2_1 \ldots e^2_m$. Clearly this product is associative. The empty path ε (or equivalently a self-loop e_v) defines the identity $i(v)$ on any vertex v. As in an undirected graph every edge has a reverse we can define the inverse of $\pi = e_0 e_1 \ldots e_n$ as $\pi^{-1} = e^*_n \ldots e^*_1 e^*_0$.

These two examples clearly illustrate the main difference between groups and groupoids:

While composition in groups is always defined we have in groupoids a "matching" condition which has to be fulfilled. In this sense groupoids are groups with "typing". Moreover, the fact that in a group there is a single base element makes the notion of a reverse path in this structure quite restrictive: it only includes paths which start from point $*$ and come back to point $*$ itself. In a groupoid such a notion can be defined between different start and end points as long as the path can be "retraced" or "reverted".

A groupoid model for our reversible **CL** allows us therefore to include reversible reductions like the one in Example 3:

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

where the end and start points differ and yet the computation can be retraced backward. This kind of computation would be excluded from a model based on a group structure; this would only allow us to include reversible reduction like

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

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

that is 'computational loops' where the same sequence of steps are first done and then undone in reverse order (cf. Section 3).

In our **rCL** model for reversible computations, we cannot talk about the inverse of a term M per se; since $\langle MN_1 \mid \rangle$ and $\langle MN_2 \mid \rangle$ will in general reduce to different terms $\langle M_1 \mid H_1 \rangle$ and $\langle M_2 \mid H_2 \rangle$, an "inverse" of M would depend on the context. Instead we have for every computational path (represented by a history H_1 or H_2) an inverse computational path (represented essentially by $\overline{H}_1$ or $\overline{H}_2$). If we consider invertible terms in **CL**, we can ignore the context: If a term M has an inverse term M^{-1} then any execution of M^{-1} will undo the effects of the execution of M in any context. This means that we only need a dummy context N such that $M^{-1} \cdot MN = N$ which correspond to a one-element base, i.e. a group instead of a general groupoid.

In other words, while groups are convenient and natural for investigating invertible terms (cf. (Dezani-Ciancaglini, 1976; Bergstra and Klop, 1980)), reversible computation requires to reason about (reversible) paths with matching conditions and multiple base points. Groupoids are therefore the natural generalisation of groups which allow us to do this.

Besides being a generalisation of groups, groupoids can also be seen as a generalisation of other mathematical structures, such as *group actions* and *equivalence relations*, as shown in Figure 1 (Ramsay and Renault, 2001).

The reduction relation $\rightarrowtail\!\!\!\rightarrow$ establishes an equivalence relation on the **rCL** terms. We can therefore define a model for **rCL** by taking the corresponding groupoid. According to (Brown, 1987), this is given by $\mathcal{G} = \mathcal{G}(\mathcal{T}, \rightarrowtail\!\!\!\rightarrow)$, where $\mathcal{T}$ is the set of all **rCL** terms and α, β, the identity i and the product operation are defined as follows:

— $\mathcal{G} \subseteq \mathcal{T} \times \mathcal{T}$ with $(T, T') \in \mathcal{G}$ iff $T \rightarrowtail\!\!\!\rightarrow T'$.
— $\mathcal{B} = \mathcal{T}$
— $\alpha((T, T')) = T$ and $\beta((T, T')) = T'$
— $(T, T') \cdot (T', T'') = (T, T'')$

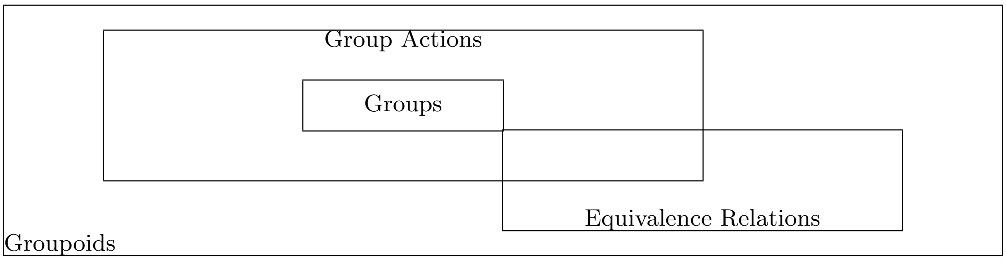


Fig. 1. Groups, Group Actions and Equivalence Relations

— $i(T) = (T, T)$
— $(T, T')^{-1} = (T', T)$.

4.1. *The Actor Groupoid*

We now show that the groupoid $\mathcal{G}(\mathcal{T}, \rightarrowtail\!\!\twoheadrightarrow)$ of reversible computations on **rCL** defined above, can also be introduced via the action of the history group $\mathcal{H}$ on the set of **rCL** terms. Intuitively, this means that each history term determines a permutation on **rCL** corresponding to a reversible computation, and vice versa.

Given a group G with identity e 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 $\pi(e) = \mathrm{id}$, where *id* is the identity automorphism, 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 $\beta((x, g, y)) = y$
— $(x, g, y) \cdot (y, h, z) = (x, hg, z)$
— $(x, g, y)^{-1} = (y, g^{-1}, x)$.

This construction is due to Ehresmann (Ehresmann, 1957) and is sometimes called *actor* groupoid or *semi-direct product* groupoid.

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

$$\pi(H)(\langle M \mid H' \rangle) = \begin{cases} \langle N \mid H' : H \rangle & \text{if } \langle M \mid H' \rangle \rightarrowtail\!\!\twoheadrightarrow \langle N \mid H' : H \rangle \\ \langle M \mid H' \rangle & \text{otherwise} \end{cases}$$

Proposition 5. For all $H \in \mathcal{H}$, $\pi(H)$ is a permutation on **rCL**.

Proof. Given an enumeration of the **rCL** terms, for any $H \in \mathcal{H}$ the map $\pi(H)$ realises a shift on **rCL** terms. □

It is interesting to note that the structure of the permutation group $\mathrm{Aut}(\mathbf{rCL}) = \{\pi(H) \mid h \in \mathcal{H}\}$ is determined by the structure of the history group. In fact, composition, identity and inverse are defined in $\mathrm{Aut}(\mathbf{rCL})$ as $\pi(H_1)(\pi(H_2)) = \pi(H_2 : H_1)$, $\pi(\varepsilon)$, and $\overline{\pi(H)} = \pi(\overline{H})$, respectively.

It is easy to verify that the groupoid $\mathcal{G}(\mathbf{rCL}, \rightarrowtail\!\!\twoheadrightarrow)$ is identical to the group action groupoid $\mathcal{G}(\mathbf{rCL}, \mathcal{H}, \pi)$ defined above. In fact, we can define a groupoid isomorphism by simply forgetting about the "connecting history".

Proposition 6. The map $\delta : \mathcal{G}(\mathbf{rCL}, \mathcal{H}, \pi) \to \mathcal{G}(\mathbf{rCL}, \rightarrowtail\!\!\!\twoheadrightarrow)$ defined by

$$\delta(\langle T, H, T' \rangle) = \langle T, T' \rangle$$

is a groupoid isomorphism.

Proof. The map δ is a groupoid morphism since it is compatible with the product, head and tail maps of the two groupoids, that is we have that $\delta(g_1 g_2) = \delta(g_1)\delta(g_2)$, $\delta(\alpha(g)) = \alpha(\delta(g))$ and $\delta(\beta(g)) = \beta(\delta(g))$. Thus, we only need to show that it is injective and surjective.

Surjective:

Let $\langle T, T' \rangle \in \mathcal{G}(\mathbf{rCL}, \rightarrowtail\!\!\!\twoheadrightarrow)$, and let $T = \langle M \mid H' \rangle$ and $T' = \langle N \mid H'' \rangle$. Then there exists a reversible reduction $T \rightarrowtail\!\!\!\twoheadrightarrow T'$. By Proposition 2 we have that $H'' = H' : H$ for some $H \in \mathcal{H}$. Therefore $\langle T, H, \pi(H)(T) \rangle = \langle T, H, T' \rangle$ is the element in $\mathcal{G}(\mathbf{rCL}, \mathcal{H}, \pi)$ such that $\delta(\langle T, H, T' \rangle) = \langle T, T' \rangle$.

Injective:

If $\delta(\langle T_1, H_1, T_1' \rangle) = \delta(\langle T_2, H_2, T_2' \rangle)$, then $\langle T_1, T_1' \rangle$ and $\langle T_2, T_2' \rangle$ identify the same element in $\mathcal{G}(\mathbf{rCL}, \rightarrowtail\!\!\!\twoheadrightarrow)$. Thus, by Proposition 2 there exists a history $H \in \mathcal{H}$ such that $\langle M \mid H' \rangle \rightarrowtail\!\!\!\twoheadrightarrow \langle N \mid H' : H \rangle$ with $T_1 = T_2 = \langle M \mid H' \rangle$ and $T_1' = T_2' = \langle N \mid H' : H \rangle$. This implies that $H_1 = H_2$ must hold. Otherwise we would have:

$$T_1' = \pi(H_1)(T_1) = \langle N \mid H' : H_1 \rangle \neq \langle N \mid H' : H_2 \rangle = \pi(H_2)(T_2) = T_2'.$$

□

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 taken an "application-oriented" approach and given prominence to the computation features of the λ-calculus and the related theory of combinatory logic rather than their other important aspects as a foundation of mathematics and in their pure form.

Given the well known relation between **CL** and the λ-calculus we can in principle define a reversible version of the λ-calculus by exploiting the encoding of the λ-calculus in **CL**. However, the variable-freeness of **CL** requires only a relatively simple kind of history term as we can avoid recording details of (multiple) variable substitution, etc.

The definition of the formal semantics of **rCL** does not change the *non-deterministic* nature of classical **CL**: depending on the particular reduction strategy we may get different computational paths starting from the same term. However, as the history term not only records "which" kind of reduction has happened, but also "where", we are able to define a converse transition relation (which is "going back in time") which is *deterministic* and thus allows us to reconstruct reduction sequences uniquely.

We also established a clear distinction between the closely related concepts of *invertibility* of terms and the *reversibility* of computations. A term M, e.g. in **CL**, is invertible if a(nother) term M^{-1} exists which is always able to compensate for the effects of the first one and vice versa. In order to introduce this notion we need concepts like an identity term I and term composition "$\cdot$". A computation is reversible, if it can be "replayed", i.e. it is possible to reconstruct the computational steps given the outcome. While invertible terms form a *group* we need the more general notion of a *groupoid* to describe reversible computations, as we can "compose" two computations only if terminal and initial term coincide. We have shown that the computational paths of our reversible calculus can be seen as the orbits of the history group acting on the space of **rCL** terms. On the other hand, the reduction rules of the **rCL** calculus introduce an equivalence relation on the terms with an associated groupoid. We have shown that the two definitions essentially identify the same groupoid as a model for the computational paths in **rCL**.

Reversibility is an essential requirement for the embedding of classical computation in Quantum Mechanics, as quantum computing devices are essentially represented by *unitary*, that is invertible, transformations. The reversible combinatory logic we have presented offers a universal model for classical reversible computation, in the sense that every classical reversible computation corresponds to a **rCL** reduction and vice versa. In the field of quantum computation this result provides an alternative high-level way to look at reversible classical computation; this is usually described in terms of circuits built out of a particular universal gate, namely the Toffoli gate (Nielsen and Chuang, 2000). The universality of **rCL** for classical reversible computation comes from the fact that, as shown by the actor groupoid model, an **rCL** reduction effectively corresponds to a permutation of the **rCL** terms. However, **rCL** is an extremely wasteful way to provide reversibility and hence of no practical use as a base for any plausible implementation of classical (irreversible) computation in a quantum mechanical setting. For this purpose much more efficient approaches have been devised (Bennet, 1973).

A more promising direction for further work is related to the definition of a model for **rCL** which is more denotational in nature. For this we aim at clarifying the relation between reversible reductions and a particular class of linear operators, namely unitary operators, which may serve as a base for a fixpoint semantics of **rCL** and similar reversible extensions of the λ-calculus as well as for the semantics of more concrete quantum programming languages such as those recently proposed in the literature (Gay, 2005). For this we hope to exploit well-established results on the relation between operator algebras (in particular C^* algebras) and groupoids (Renault, 1980).

References

Abramsky, S. (2001). A structural approach to reversible computation. In Beauquier, D. and Matiyasevich, Y., editors, *LCCS 2001: Proceedings of the International Workshop on Logic and Complexity in Computer Science*, pages 1–16.

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

Barendregt, H. P. (1984). *The Lambda Calculus – Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam – New York – Oxford, revised edition.

Bennet, C. (1973). Logical reversibility of computation. *IBM J. Res. Develop.*, 17:525–532.

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

Brandt, W. (1926). Über eine Verallgemeinerung des Gruppengriffes. *Mathematische Annalen*, 96:360–366.

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

Curry, H. B. and Feys, R. (1958). *Combinatory Logic*. North-Holland, Amsterdam.

Danos, V. and Krivine, J. (2004). Reversible communicating systems. In *Proceedings of CONCUR 2004*, volume 3170 of *LNCS*, pages 292–307. Springer Verlag.

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

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

Ehresmann, C. (1957). Gattungen von lokalen Strukturen. *Jahresber. Deutsch. Math.-Verein*, 60:49–77.

Gay, S. (2005). Quantum programming languages. *Bulletin of the EATCS*, 86.

Hindley, J. R., Lercher, B., and Seldin, J. P. (1972). *Introduction to Combinatory Logic*, volume 7 of *London Mathematical Society Lecture Note Series*. Cambridge University Press.

Hindley, J. R. and Seldin, J. P. (1986). *Introduction to Combinators and λ-Calculus*, volume 1 of *London Mathematical Society Student Texts*. Cambridge University Press.

Kitaev, A., Shen, A., and Vyalyi, M. (2000). *Classical and Quantum Computation*. Cambridge University Press, Cambridge, UK.

Mundici, D. and Sieg, W. (1995). Paper machines. *Philosophica Mathematica*, Series III, 3(1):5–30.

Nielsen, M. and Chuang, I. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK.

Phillips, I. and Ulidowski, I. (2005). Operational semantics of reversibility in process algebra. In Aceto, L. and Gordon, A., editors, *Workshop on Algebraic Process Calculi: The First Twenty Five Years and Beyond (PA '05)*, number NS-05-3 in BRICS Notes Series, pages 200–203, Bertinoro, Forli, Italy.

Ramsay, A. and Renault, J., editors (2001). *Groupoids in Analysis, Geometry, and Physics*, volume 282 of *Contemporary Mathematics*. AMS, Providence, RI.

Renault, J. (1980). *A Groupoid Approach to C*-Algebras*, volume 793 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin – Heidelberg – New York.

van Tonder, A. (2004). A lambda calculus for quantum computation. *SIAM Journal of Computation*, 33(5):1109–1135.

Vitanyi, P. (2005). Time, space, and energy in reversible computing. In *Proceedings of the ACM International Conference on Computing Frontiers*, Ischia, Italy. ACM.

Weinstein, A. (1996). Groupoids: Unifying internal and external symmetry. *Notices of the AMS*, 43(7):744–752.