

Geometry of Resource Interaction

A Minimalist Approach

Marco Solieri ^{*†}

Université Paris 13, Sorbonne Paris Cité – Laboratoire d’Informatique de Paris-Nord, CNRS
Villetaneuse, France

Università di Bologna – Dipartimento di Informatica, Scienza e Ingegneria, INRIA
Bologna, Italy
`ms@xt3.it`

Abstract

The Resource λ -calculus (RC) is a variation of the λ -calculus where arguments can be superposed and must be linearly used. Hence it is a model for non-deterministic and linear programming languages, and the target language of Taylor expansion of λ -terms. In a strictly typed restriction of RC, we study the notion of path persistency, and we define a Geometry of Interaction that characterises it. The GoI is also invariant under reduction and able to count the addends of normal forms.

Introduction

Geometry of Interaction The dynamics of β -reduction or cut elimination can be described in a purely geometric way – studying paths in the graphs that represent terms or proofs, and looking at those which are *persistent*, i.e. that have a residual path in any reduct. The quest for an effective semantical characterisation of persistence separately produced three notions of paths: *legality*, formulated by topological conditions about symmetries on some cycles [2]; *consistency*, expressed similarly to a token-machine execution [10] and developed to study the optimal reduction; and *regularity*, defined by a dynamic algebra [9, 5]. The notions are equivalent [1], and their common core idea —describing computation by local and asynchronous conditions on routing of paths— inspired the design of efficient parallel abstract machines [11, 13, for instance]. More recently, the Geometry of Interaction (GoI) approach has been fruitfully employed for semantical investigations which characterised quantitative properties of programs, mainly the complexity of their execution time [3].

Taylor expansion of λ -terms and the Resource Calculus Linear Logic’s decomposition of the intuitionistic implication unveiled the relation between the algebraic concept of linearity to the computational property of a function argument to be used exactly once. Such a decomposition was then applied not only at the level of types, but also at the level of terms, in particular extending the λ -calculus with differential constructors and linear combination of ordinary terms [7]. This construction allows to consider the complete Taylor expansion of a term, i.e. the infinite sum of all the approximations of the reduction of a term, which was thus shown to commute with computation of Böhm trees. The ideal target language for the expansion was isolated as the Resource λ -calculus (RC), which is a promotion-free restriction of the Differential λ -calculus [8]. Taylor expansion originated various investigations on quantitative semantics, using the concept of power series for describing program evaluation, and has been applied in various non-standard models of computation [4, 12].

^{*}The author is deeply grateful to Michele Pagani and Stefano Guerrini for their advice.

[†]Supported by the ANR project ANR-2010-BLAN-021301 LOGOI.

Aim and results How can the two aforementioned semantics approaches interact? What is the relation between the GoI’s execution formula and the Taylor expansion of β -reduction? We present the first steps towards this direction. After having concisely introduced RC (§1), we consider the Resource Nets (RNs), that are the type-restricted translation of resource terms into Differential Interaction Nets (§2). We then study the appropriate notion of paths (§3), extending the notion of persistency to paths in RNs dealing with the fact that the reduction of a term t is a sum of nets $t_1 + \dots + t_n$. In particular, we observe that every path of t_i has to be a residual of some path in t , and that the reduction strongly normalises. Thus, we say a path of t to be persistent whenever it has a residual in at least one of the addends of the reduct of t . Restricting the calculus to the constant type, whose only inhabitant is the value $\star$, we have $t \rightarrow \star + \dots + \star$. Now there is only one persistent path of $\star$, the trivial one, therefore we prove that persistent paths of t are as many as persistent paths of its normal form (Th. 1). Furthermore, we define a suitable GoI for RC, in order to characterise persistency (§4). We define the notion of regularity by $\mathbf{r}\mathfrak{L}^*$, an appropriate adjustment of the Dynamic Algebra, where exponentials (! and ?) become a sort of n -ary multiplicatives (resp. $\otimes$ and $\wp$), whose premises are not ordered. Morally, they are the sum of the multiplicatives we obtain by considering all the $n!$ permutations of their premises. We show our algebra is invariant under reduction (Th. 2), from which we obtain the equivalence of persistence to regularity (Th. 3) and also that the number of addends in a normal form is equal to the number of regular paths (Cor. 1).

Related works In a very closely related work by De Falco [6], a GoI construction for DINs is formulated. Besides the similarities in the technical setting of DINs, our formulation turns out to be simpler and more effective, mainly thanks to: (1) the restriction to closed and ground-typed resource nets, (2) the associative syntax we adopted for exponential links, and (3) the stronger notion of path we use. The first simplifies the shape of paths being persistent, because it implies that they are palindrome—they go from the root to the $\star$ and back to the root—and unique in every normal net/term. The second simplifies the management of the exponential links, because it ensures associativity and delimits their dynamics in only one pair of links, while in [6] this property was completely lost and the system more verbose. De Falco uses binary exponential links and introduces a syntactical embedding of the sum in nets by mean of binary links of named sums, and then recover associativity with an equivalence on nets. Compared to ours, their choice results in a drastically more complex GoI construction, even though the paper hints at the extensibility with promotion (corresponding to the full Differential λ -calculus) or even additives. The third ingredient allows us to consider the full reduction, i.e. including the annihilating rule, while in [6] a “weak” variant is studied, where this kind of redexes are frozen, and the GoI only characterises the corresponding notion of “weak-persistence”. Indeed, we restrict to paths that cross every exponential in the net (we prove it is always in case of persistency), thus whenever $t \rightarrow 0$, a path necessarily crosses the annihilating redex and the dynamic algebra is able to detect it.

1 Resource calculus

The Resource Calculus is, on one hand, a linear and thus finitary restriction of the λ -calculus: an argument $[s]$ must be used by an application $t[s]$ exactly once, i.e. it cannot be duplicated nor erased, so every reduction enjoys strong normalisation. On the other hand, it adds non-determinism to the λ -calculus, because the argument is a finite multiset of ordinary terms. The reduct is then defined as the superposition of all the possible ways of substituting each of the

arguments, i.e. a sum. When arguments provided to a function are insufficient or excess the function's request, i.e. the number of variable occurrences, then the computation is deadlocked and the application reduce to 0. We shall omit the “resource” qualification in the terminology.

Definition 1 (Syntax). Let $\mathbb{V}$ be the grammar of a denumerable set of variable symbols $x, y, z, \dots$. Then, the set Δ of the *simple terms* and the set $\Delta^!$ of *simple polyterms* are inductively and mutually generated by the following grammars.

$$\text{Simple terms: } \mathbb{M} ::= \star \mid \mathbb{V} \mid \lambda \mathbb{V}.\mathbb{M} \mid \mathbb{M} \mathbb{B} \quad \text{Simple polyterms: } \mathbb{B} ::= 1 \mid [\mathbb{M}] \mid \mathbb{B} \cdot \mathbb{B} \quad (1)$$

Where: $\star$ is the constant dummy value, brackets delimit multisets, $\cdot$ is the multiset union (associative and commutative), 1 is the empty multiset (neutral element of $\cdot$). So that $([x] \cdot 1) \cdot [y] = [x, y]$. Simple terms are denoted by the lowercase letters of the latin alphabet around t , polyterms in uppercase letters around T . The set $\mathbf{N}(\Delta)$ of *terms* (resp. the set $\mathbf{N}(\Delta^!)$ of *polyterms*) is the set of finite formal sums of simple terms (resp. polyterms) over the semiring $\mathbf{N}$ of natural numbers. We also assume all syntactic constructors of simple terms and polyterms to be extended to sums by (bi-) linearity. E.g. $(\lambda x.(2x + y))[z + 4u]$ is a notational convention for $2(\lambda x.x)[z] + 8(\lambda x.x)[u] + (\lambda x.y)[z] + 4(\lambda x.y)[u]$.

Definition 2 (Reduction). A redex is a simple term in the form: $(\lambda x.s)T$. Let the free occurrences of x in s be $\{x_1, \dots, x_m\}$. The *reduction* is the relation $\rightarrow$ between polyterms obtained by the context closure and the linear extension to sum of the following elementary reduction rule.

$$(\lambda x.s) [t_1, \dots, t_n] \rightarrow \begin{cases} \sum_{\sigma_n \in S_n} s \{t_1/x_{\sigma_n(1)}, \dots, t_n/x_{\sigma_n(n)}\} & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases} \quad (2)$$

Where S_n denotes the set of permutations of the first n naturals, and $\{t/x\}$ is the usual capture-avoiding substitution. If $t \rightarrow^* t' \not\rightarrow$, where $\rightarrow^*$ is the reflexive transitive closure of $\rightarrow$, we write $\text{NF}(t) = t'$.

Example 1. Let $I = I' = \lambda x.x$ and also let $t = \lambda f.f_1[f_2[\star]]$. Then $t[I, I] \rightarrow f_1[f_2[\star]]\{I/f_1, I'/f_2\} + f_1[f_2[\star]]\{I/f_2, I'/f_1\}$ that is $I[I'[\star]] + I'[I[\star]]$, normalising to $I'[\star] + I[\star] \rightarrow 2\star$. Note also a case of annihilation in $t[I] \rightarrow 0$. Finally, observe that if $s = (\lambda x.\star)T \rightarrow \star$ then T must be 1 (otherwise $s \rightarrow 0$).

2 Resource nets

A resource net is a graphical representation of a typed term by means of a syntax borrowed from Linear Logic proof nets, where n -ary $?$ links have a symmetrical dual. The exponential modality is however deprived of promotion, so that it merely represents superposition of proofs and contexts.

2.1 Pre-nets

Definition 3 (Links). Given a denumerable set of symbols called *vertices*, a *link* is a triple (P, K, C) , where: P is a sequence of vertices, called premises; K is a kind, i.e. an element in the set $\{\star, \multimap, \multimap^!, \multimap^?, \multimap^!\}$; C is a singleton of a vertex, called conclusion, disjoint from P . A link $l = ((u_1, \dots, u_n), \kappa, \{v\})$ will be denoted as $\langle u_1, \dots, u_n \ (\kappa) \ v \rangle$, or depicted as in Fig. 1. In the

graphical representations, vertices of a link shall be placed following the usual convention for graphs of λ -calculus (outs on the top, and ins on the bottom); the arrow line shall be used to distinguish the conclusion of a link. The *polarity* of a vertex is an element in $\{\text{in}, \text{out}\}$ and we say they are opposite, and the *arity* of a link is the length of its premises' sequence; both are determined by the link's kind, as shown in Fig. 1. When $v \in P(l) \cup C(l)$ for some vertex v and link l , we write that v is linked by l , or $v \in l$. The exponential links $!$ and $?$ whose arity is 0 are respectively called co-weakening and weakening.

Definition 4 (Types). A *type*, or formula, is a word of the grammar given by $\mathbb{T} ::= \star \mid \mathbb{E} \multimap \mathbb{T}$ and $\mathbb{E} ::= !\mathbb{T}$, where $\star$ is the only *ground* type. A *typing function* $\mathcal{T}$ is a map from vertices to types such that, if A, B are types, then $\mathcal{T}$ respects the following constraints. Constant: $\langle (\star) \star \rangle$. Linear implications: $\langle A, B \multimap A \multimap B \rangle$ and $\langle A, B \multimap A \multimap B \rangle$. Exponentials: $\langle A, \dots, A (!) !A \rangle$ and $\langle A, \dots, A (?) !A \rangle$.

Definition 5 (Pre-nets). A *simple pre-net* $\mathcal{G}$ is a triple $(V, L, \mathcal{T})$, where V is a set of vertices, L is a set of links and $\mathcal{T}$ a typing function on V , such that for every vertex $v \in V$ the followings holds: (1) there are at least one and at most two links l, l' such that $l \ni v \in l'$, and when there is only l , then v is called a *conclusion* of $\mathcal{G}$; (2) the set $C(\mathcal{G})$ of conclusions is non empty and when it is the singleton v , then the $\mathcal{G}$ is called *closed* and v must be *out*; (3) if $l \ni v \in l'$, then l, l' have opposite polarities.

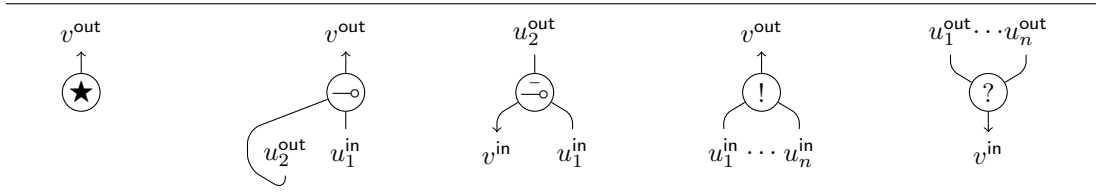
We shall write $V(\mathcal{G})$ to denote the set $V \in \mathcal{G}$. The type of a pre-net $\mathcal{G}$ is the type T associated to its only *out* conclusion, so we write $\mathcal{G} : T$. The *interface* of a simple pre-net $\mathcal{G}$ is the set $I(\mathcal{G})$ of all ordered pairs $(T, \mathbf{p})$ such that for all $v \in C(\mathcal{G})$, v is of type T and has polarity $\mathbf{p}$. A general *pre-net* is a linear combination of simple pre-nets $\mathcal{G}_1 + \dots + \mathcal{G}_n$, where for any $1 \leq i, j \leq n$, we have: $V(\mathcal{G}_i) \cap V(\mathcal{G}_j) = \emptyset$ and $I(\mathcal{G}_i) = I(\mathcal{G}_{i+1})$. We shall simply use 0 to denote each of the empty sums of pre-nets having the same interface I , for every interface I .

2.2 Term translation and net reduction

As the usual translation of the λ -calculus into MELLL proof nets, the $\multimap$ -link is used for translating λ -abstraction, the $\multimap$ -link for application, and the $?$ -link for contracting together all the occurrences of the same variable. In addition, we use $!$ -link for polyterm and formal sum of net for $\dots$ formal sum of terms.

Definition 6 (Term translation). Given a simple term t , the *translation* $\llbracket t \rrbracket$ is a pre-net having one *out* conclusion and a possibly empty set of *in* conclusions. The translation is defined in Fig. 2 where: the final step only adds a $?$ -link on every occurrence of a free variable x , for all free variables of t ; and the actual work is performed by the $\langle t \rangle$, by induction on the syntax of t . Moreover a sum of simple terms is translated to the sum of their translation, i.e.: $\llbracket t_1 + \dots + t_k \rrbracket = \llbracket t_1 \rrbracket + \dots + \llbracket t_k \rrbracket$.

Figure 1 Links: kind, arity and polarity



Note that a net translation is always defined for simple terms while it is not for general terms, because of possible incompatibility in the interfaces of translated addends.

Definition 7 (Resource permutations). Given a simple pre-net $\mathcal{G}$, a *resource permutation* $\sigma_{\mathcal{G}}$ is a total function from the set of !-links in $\mathcal{N}$ to $\bigcup_n S_n$ such that if a link l has arity m , then $\sigma_{\mathcal{N}}(l)$ is an element σ_m of S_m . We shall also write σ_l for $\sigma_{\mathcal{G}}(l)$ and denote the set of resource permutation of $\mathcal{G}$ as $S_{\mathcal{G}}$.

Definition 8 (Resource net reduction). The *redex* of a cut w in a simple pre-net is the pair of links having w as conclusion. The *simple reduction* $\rightarrow$ is the graph-rewriting relation from simple pre-nets to pre-nets defined by the following elementary reduction steps.

$$\mathcal{G}, \langle u, v \text{ (}\multimap\text{)} w \rangle, \langle u', v' \text{ (}\multimap\text{)} w \rangle \rightarrow \mathcal{G}[v \equiv v', u \equiv u'] \quad (3)$$

$$\mathcal{G}, \langle v_1, \dots, v_n \text{ (!)} w \rangle, \langle u_1, \dots, u_m \text{ (?)} w \rangle \rightarrow \begin{cases} \sum_{\sigma_n \in S_n} \mathcal{G}_{\sigma_n}[v_1 \equiv v_{\sigma_n(1)}, \dots, v_n \equiv v_{\sigma_n(n)}] & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases} \quad (4)$$

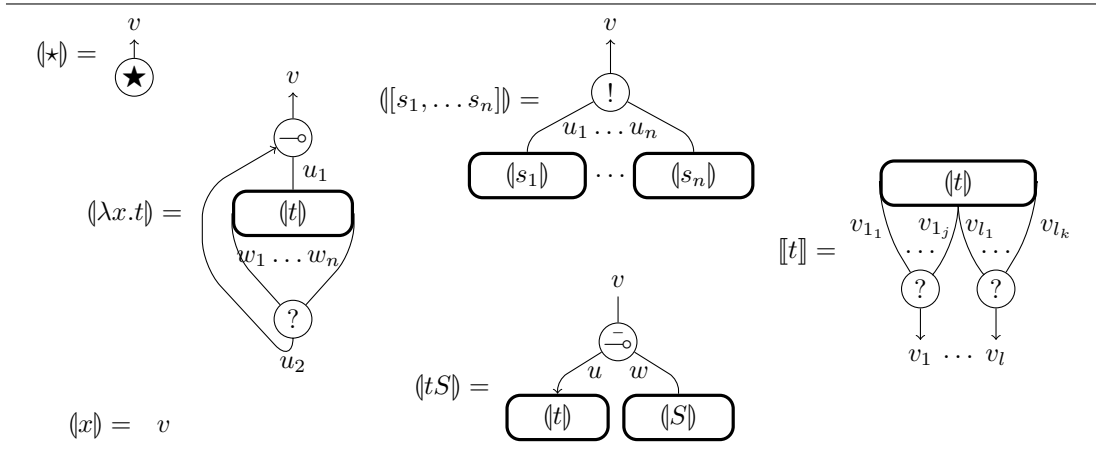
Where $\mathcal{G}_i[v \equiv u]$ denotes the i -th copy of the pre-net $\mathcal{G}$, where the vertices v, u have been equated. In such a case, we say then there is a *simple reduction step* $\rho : \mathcal{G} \rightarrow \mathcal{S}$, where $\mathcal{S}$ is a sum of simple pre-nets and is also written as $\rho(\mathcal{G})$. The *reduction* is the extension of the simple reduction to formal sums of simple pre-nets: if $\mathcal{G} \rightarrow \mathcal{S}$, then $\mathcal{G} + \mathcal{S}' \rightarrow \mathcal{S} + \mathcal{S}'$. If $\mathcal{G} \rightarrow^* \mathcal{S}' \not\rightarrow$, we write $\text{NF}(\mathcal{G}) = \mathcal{S}'$.

Definition 9 (Resource nets). Let $t \in \Delta$ and $\llbracket t \rrbracket \rightarrow^* \mathcal{S}$, for a sum $\mathcal{S} = \mathcal{N}_1 + \dots + \mathcal{N}_n$, where each $\mathcal{N}_i$ is a pre-net. Then $\mathcal{N}_i$ is called a *simple resource net* and $\mathcal{S}$ a *resource net*. From now on we shall again avoid to repeat the “resource” naming of nets.

We recall the net reduction can simulate the term reduction and strongly normalises.

Example 2. Consider $\delta = \lambda x. x[x]$ and recall the terms I and t from Ex. 1. First notice $\llbracket \delta \rrbracket$ is not a pre-net, because a typing function on the structure of vertices and links does not exist. Then look at Fig. 3. On the left extremity: $\llbracket I \rrbracket$ is closed and $\llbracket I \rrbracket : !\star \multimap \star$. On the middle left:

Figure 2 Pre-translation and translation of simple terms into simple nets



$\mathcal{N} : \star$ is not a translation of a term, but it is a net, because $\llbracket t[x, y] \rrbracket \rightarrow \mathcal{N}$ by a linear implication step. On the right side: an exponential reduction step involving index permutation from $\mathcal{N}$ into a sum of two normal simple nets. Moreover, the reduct is equal to $\llbracket x[y[\star]] + y[x[\star]] \rrbracket$.

3 Paths

We introduce some basic definitions about the paths, where the most notable characterise the paths where the computation is visible (straightness) in its entirety (maximality and comprehensiveness). This last notion is the only difference with respect to the classic notion of path as formulated in [5].

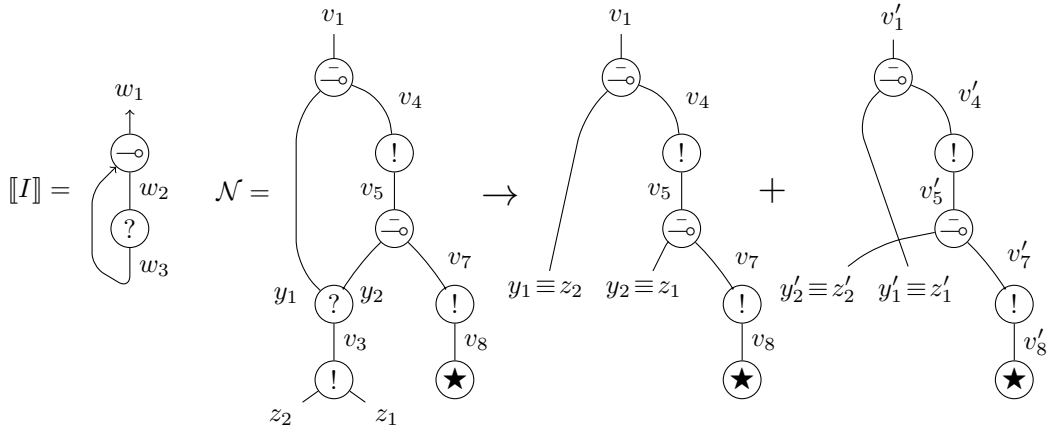
Definition 10 (Path). Given a simple net $\mathcal{N}$, two vertices $u, w \in \mathcal{N}$ are *linked*, or *connected*, if there is a link $l \in \mathcal{N}$ s.t. $u, w \in l$. A *path* $\pi = (v_1, \dots, v_n)$ with $n > 0$ in $\mathcal{N}$ is a sequence of vertices s.t. for all $i < n$, the vertices v_i, v_{i+1} are connected. We call π *trivial* if its length is 1; we say π *unitary* if is 2, so that there is only one link crossed by π .

Moreover, if π crosses consecutively the same link l more than once, then π is called *bouncing*. If l is not a $\star$ -link and π crosses l through v, v' such that $v, v' \in C(l)$ or $v, v' \in P(l)$, then π is *twisting*. When π is both non-bouncing and non-twisting, π is *straight*. Moreover, π is *maximal* if there is no other path $\pi' \in \mathcal{N}$ s.t. $\pi \sqsubseteq \pi'$, where $\sqsubseteq$ is the prefix order on sequences. Also, π is *comprehensive* when it crosses all the premises of all the exponential links. Finally if π is both straight and maximal, the π is an *execution path*. In a net $\mathcal{S}$, we denote with $E(\mathcal{S})$ ($E^+(\mathcal{S})$) the set of execution (and comprehensive) paths in some simple $\mathcal{N}$ addend of $\mathcal{S}$.

Given two paths π, π' we denote the *reversal* of π as π^- , while the *concatenation* of π' to π as $\pi :: \pi'$.

We can now concretely aim to a proper notion path persistence, that intuitively means “having a residual”, so first we inspect and define the action of residual of path. The case of linear implication is straightforward, because the rewriting is local and we only have to ensure that a path does not partially belong to a redex. The case of exponential, instead, is rather more delicate, because the rewriting is global: a simple net rewrites to a sum of simple nets, hence a path may be duplicated in several addends or destroyed. Which addends contain the

Figure 3 Example: nets



residual(s) of a given crossing? The net reduction consists of the sum of all the permutation of the indices of the !-links (cf. Def. 8), thus each addend contains all and only the paths that respect the addend's own permutation, for any crossing of the redex. If a path π is persistent, then, there must be a permutation such that π always crosses the redex respecting the correspondences fixed by the permutation.

Definition 11 (Path residual). Given a net $\mathcal{N}$ and a reduction ρ on a redex $R \in \mathcal{N}$, we say a path $\pi \in \mathcal{N}$ is *long enough* for R when neither its first nor its last vertex is the cut of R . In such a case, there we can express π isolating every *crossing* of R , that is a maximal sub-sequence of π entirely contained in R as: $\pi = \pi_0 :: \chi_1 :: \pi_1 :: \dots :: \chi_k :: \pi_k$, where for any $0 \leq l \leq k$, the subpath χ_l is a crossing for R .

The *path reduction* is a function from paths in $\mathcal{N}$ to sums of paths in $\rho(\mathcal{N})$. The residual of π , written $\rho(\pi)$, is defined according to the reduction rule used by ρ and by extension of the case of $\rho(\chi_l)$.

1. Linear implication cut. If χ_l is as in Eq. 3, then $\rho(\chi_l)$ is defined as follows.

$$\rho((v, w, u)) = (v) \quad \rho((v', w, u')) = (v') \quad (5)$$

$$\rho((v, w, u')) = 0 \quad \rho((v', w, u)) = 0 \quad (6)$$

The residual of the whole π is defined as:

$$\rho(\pi) = \begin{cases} \pi_0 :: \rho(\chi_1) :: \pi_1 :: \dots :: \rho(\chi_k) :: \pi_k & \text{if for any } i, \rho(\chi_i) \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

2. Exponential cut. Let χ_l be as in Eq. 4 and $\sigma_n \in S_n$. First, we define the residual of χ_l w.r.t. σ_n , for every pair of indices $0 \leq i \leq n$, and $0 \leq j \leq m$:

$$\rho^{\sigma_n}(v_i, w, u_j) = \begin{cases} (v_j) & \text{if } n = m, \text{ and } \sigma_n(i) = j \\ 0 & \text{if } n \neq m, \text{ or } \sigma_n(i) \neq j \end{cases} \quad (8)$$

Now, similarly to Eq. 7, we can define the residual of the path π with respect to σ_n :

$$\rho^{\sigma_n}(\pi) = \begin{cases} \pi_0 :: \rho^{\sigma_n}(\chi_1) :: \pi_1 :: \dots :: \rho^{\sigma_n}(\chi_k) :: \pi_k & \text{if for any } l, \rho^{\sigma_n}(\chi_l) \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Finally, we can define the residual of π as the sum of all the residuals, for any σ_n :

$$\rho(\pi) = \sum_{\sigma_n \in S_n} \rho^{\sigma_n}(\pi) \quad (10)$$

If $\rho(\pi) \neq 0$, then π is *persistent to ρ* . If, for every reduction sequence $\rho = (\rho_1, \dots, \rho_m)$, and for every $1 \leq i \leq m$, the path π is persistent to ρ_i , then π is *persistent*.

Example 3. Recall the nets discussed in Ex. 2 and let $\rho(\mathcal{N}) = \mathcal{N} \rightarrow \mathcal{N}_l + \mathcal{N}_r$, resp. be the left and the right addend of Fig. 3. $E(\llbracket I \rrbracket)$ are $\phi = (w_1, w_2, w_3)$ and ϕ^- . $E(\mathcal{N})$ includes: $\pi_1 = (v_1, y_1, v_3, z_1)$ and $\pi_2 = (v_1, y_1, v_3, z_2)$; both are persistent, since: $\text{NF}(\pi_1) = \pi_{1,r} = (v_1, y_1 \equiv z_1)$ and $\text{NF}(\pi_2) = \pi_{2,l} = (v_1, y_1 \equiv z_2)$. Notice they do not belong to the same addend: $\pi_{1,r} \in \mathcal{N}_r$, while $\pi_{2,l} \in \mathcal{N}_l$. A non-persistent path is $\pi_3 = (z_1, v_3, y_1, v_4, v_5, y_2, v_3, z_1)$: the two crossings of the redex belong to two distinct permutations.

Lemma 1. *For any closed $\llbracket t \rrbracket : \star$, every persistent $\pi \in E(\llbracket t \rrbracket)$ is comprehensive.*

Proof sketch. By induction on an anti-sequence of reduction to the normal form $[\star, \dots, \star]$, we prove a stronger thesis: a vertex $v \notin \pi$ if and only if there exist a (co-) weakening l such that $v \in C(l)$. $\square$

An exponential reduction step partitions $E(\mathcal{N})$ accordingly to the addends it creates in the reduct.

Theorem 1. *For any closed $\llbracket t \rrbracket : \star$, every reduction step ρ induces a bijection between persistent paths in $E(\mathcal{N})$ and persistent paths $E(\rho(\mathcal{N}))$.*

Proof sketch. Every non-annihilating n -ary exponential reduction step of a redex R in a simple net $\mathcal{N}$ induces a bijection between the $n!$ permutations of S_n in the R and the reduct $\mathcal{N} \rightarrow \mathcal{N}_1 \dots \mathcal{N}_{n!}$. But $\pi \in E(\mathcal{N})$ is also comprehensive by Lem. 1, so π must respect a unique permutation $\sigma_{\mathcal{N}} \in S_{\mathcal{N}}$, which contains a unique $\sigma_n \in S_n$, and this makes also the exponential reduction rules evidently bijective. $\square$

4 Execution

We define a weight assignment path, we give a Dynamic Algebra for them, and define the Execution Formula of a net using them. The equational theory on weights accurately computes path reduction. We continue to follow the spirit of the GoI for MELL as given in [5], where exponentials without promotion can be characterised similarly to a sort of n -ary multiplicatives, and whose weights are assigned spanning the space of permutations of their indices.

Definition 12 (Dynamic Algebra). The $\mathbf{rL}^*$ algebra is defined over symbols in $\{0, 1, p, q, e_n, \star\}$, where n is a natural number. A word of its alphabet, called *weight*, is generated by an unary *inversion* operator $(\cdot)^*$ and a binary *concatenation* operator with infix implicit notation. The concatenation operator is a monoid, whose identity element is 1, and whose absorbing element is 0 (cf. Fig. 4a). Moreover, the inversion operator is idempotent and involutive for concatenation (cf. Fig. 4b), and satisfies the neutralisation and two annihilation equations in Fig. 4c.

Definition 13 (Weighting). The *permuted base weighting* is a map w from a unitary straight path $\pi \in \mathcal{N}$ and a resource permutation $\sigma_{\mathcal{N}}$ to a weight of $\mathbf{rL}^*$, written as $w^{\sigma_{\mathcal{N}}}(\pi)$.

$$w^{\sigma_{\mathcal{N}}}((u, v)) = \begin{cases} \star & \text{if there is } \langle (\star) u \rangle \text{ and } u = v \\ p & \text{if there is } \langle u, w(-\circ) v \rangle \text{ or } \langle u, w(\bar{\circ}) v \rangle \\ q & \text{if there is } \langle w, u(-\circ) v \rangle \text{ or } \langle w, u(\bar{\circ}) v \rangle \\ e_i & \text{if there is } \langle u_1, \dots, u_i, \dots, u_n (?) v \rangle \text{ and } u_i = u \\ e_{\sigma_r(i)} & \text{if there is } r = \langle u_1, \dots, u_i, \dots, u_n (!) v \rangle \\ (w^{\sigma_{\mathcal{N}}}(v, u))^* & \text{otherwise.} \end{cases} \quad (19)$$

Figure 4 The $\mathbf{rL}^*$ algebra

(a) Monoid rules	(b) Inversion rules	(c) Concatenation rules
$a(bc) = (ab)c$ (11)	$(a^*)^* = a$ (14)	$aa^* = 1$ (16)
$a1 = 1a = a$ (12)	$(ab)^* = b^*a^*$ (15)	$qp^* = pq^* = 0$ (17)
$a0 = 0a = 0$ (13)		$e_i e_{j \neq i}^* = 0$ (18)

The *permuted weighting* is the lifting of the permuted base weighting to generic straight paths, and the *path weighting* is the sum of all the permuted weights of a path, for any resource permutation:

$$\mathbf{w}^{\sigma_{\mathcal{N}}}(v) = 1 \quad \mathbf{w}^{\sigma_{\mathcal{N}}}((u, v) :: \pi) = w^{\sigma_{\mathcal{N}}}(u, v) \mathbf{w}^{\sigma_{\mathcal{N}}}(\pi) \quad \mathbf{w}(\pi) = \sum_{\sigma_{\mathcal{N}} \in S_{\mathcal{N}}} \mathbf{w}^{\sigma_{\mathcal{N}}}(\pi). \quad (20)$$

Definition 14 (Execution). A path π is *regular* if $\mathbf{w}(\pi) \neq 0$. The *execution* of a net $\mathcal{S}$, is defined as:

$$\mathfrak{E}\mathfrak{x}(\mathcal{N}) = \sum_{\pi \in E^+(\mathcal{N})} \mathbf{w}(\pi). \quad (21)$$

Example 4. Consider the nets in Fig. 3 as described in Ex. 2, and the paths discussed in Ex. 3. Assume to have two simple nets of $\llbracket I \rrbracket_1$ and $\llbracket I \rrbracket_2$ whose vertices are distinguished by subscripts, and let $\mathcal{M}$ be the simple net obtained connecting them with $\mathcal{N}$, so that $w_{1_1} \equiv z_1$ and $w_{1_2} \equiv z_2$. Observe that $\text{NF}(\mathcal{M}) = \langle (\star) v_{1_l} \equiv v_{8_l} \rangle + \langle (\star) v_{1_r} \equiv v_{8_r} \rangle$. Now $\mathbf{w}(\pi_1 :: \phi_1) = pe_1e_2^*p^*e_1q =_{\mathfrak{r}\mathfrak{L}^*} pe_1e_2^*p^*e_1q + pe_1e_1^*p^*e_1q =_{\mathfrak{r}\mathfrak{L}^*} 0 + pp^*e_1q =_{\mathfrak{r}\mathfrak{L}^*} 0 + e_1q$. Verify that $\pi_1 :: \phi_1$ and $\pi_2 :: \phi_2$ are prefixes of the only two paths in $E^+(\mathcal{M})$ that are persistent and that indeed $\mathfrak{E}\mathfrak{x}(\mathcal{M}) =_{\mathfrak{r}\mathfrak{L}^*} \star + \star$.

The Dynamic Algebra $\mathfrak{r}\mathfrak{L}^*$ is a suitable denotational semantic for ground typed nets—it is invariant under reduction.

Theorem 2. For any closed net $\mathcal{S} : \star$ and reduction ρ , $\mathfrak{E}\mathfrak{x}(\mathcal{S}) =_{\mathfrak{r}\mathfrak{L}^*} \mathfrak{E}\mathfrak{x}(\rho(\mathcal{S}))$.

Proof sketch. An inspection of the definition of path reduction (Def. 11) allows to verify that every step ρ is simulated by the equations of $\mathfrak{r}\mathfrak{L}^*$ so that for all $\pi \in E^+(\mathcal{N})$, $\mathbf{w}(\pi) = \mathbf{w}(\rho(\pi))$. We can extend it by induction on the length of π and then, by Th. 1, we extend also to $E^+(\mathcal{N})$ and conclude. $\square$

The traditional theorem for GoI constructions is then immediate, and we can also obtain a quantitative information about the number of addends that a given term normalise to.

Theorem 3. For any closed net $\mathcal{S} : \star$, a path $\pi \in E^+(\mathcal{S})$ is persistent if and only if π is regular.

Corollary 1. For any $\llbracket t \rrbracket : \star$, the regular paths in $\mathfrak{E}\mathfrak{x}(\llbracket t \rrbracket)$ are as many as the addends in $\text{NF}(t)$.

Proof sketch. Every $t_i \in \text{NF}(t)$ is $\star$, and the cardinality of $E^+(\llbracket \star \rrbracket)$ is 1. But cardinality of $E^+(\text{NF}(\llbracket t \rrbracket))$ is preserved by expansion also in $E^+(\llbracket t \rrbracket)$ (Th. 1), where regularity is properly characterised (Th. 3). $\square$

Conclusion

We studied the notion of persistency in a restriction of the Resource Calculus (RC) and we defined a proper Geometry of Interaction construction that: is invariant under reduction, characterises persistency, and also counts addends of normal forms. Directions of ongoing and future investigation primarily include extensions of the present minimalist formulation for RC. We would like to consider a resource variant of PCF, where the restriction to ground types is innocuous and where ordinary PCF terms can be Taylor-expanded. We would also like to study how paths behave in the presence of promotion, where duplication is allowed. Indeed we know that, in such a case, the shape of persistent crossings of exponential redexes does not necessarily respect the definition we gave by mean of fixed permutations (sum does not commute with promotion). Subsequently, we aim to understand connections of path reduction and execution with Taylor expansion and other closely related concepts, such as head reduction and Böhm trees.

References

- [1] Andrea Asperti, Vincent Danos, Cosimo Laneve, and Laurent Regnier. Paths in the lambda-calculus. In *Logic in Computer Science, 1994. LICS'94. Proceedings., Symposium on*, pages 426–436. IEEE, 1994.
- [2] Andrea Asperti and Cosimo Laneve. Paths, computations and labels in the lambda-calculus. *Theoretical Computer Science*, 142(2):277–297, 1995.
- [3] Ugo Dal Lago. Context semantics, linear logic, and computational complexity. *ACM Transactions on Computational Logic (TOCL)*, 10(4):25:1–25:32, August 2009.
- [4] Vincent Danos and Thomas Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Information and Computation*, 209(6):966–991, 2011.
- [5] Vincent Danos and Laurent Regnier. Proof-nets and the Hilbert space. In *Advances in Linear Logic*, pages 307–328, New York, NY, USA, 1995. Cambridge University Press.
- [6] Marc de Falco. The geometry of interaction of differential interaction nets. In *Logic in Computer Science, 2008. LICS'08. 23rd Annual IEEE Symposium on*, pages 465–475. IEEE, June 2008.
- [7] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1):1–41, 2003.
- [8] Thomas Ehrhard and Laurent Regnier. Böhm Trees, Krivine’s Machine and the Taylor Expansion of Lambda-Terms. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and JohnV. Tucker, editors, *Logical Approaches to Computational Barriers*, volume 3988 of *Lecture Notes in Computer Science*, pages 186–197. Springer Berlin Heidelberg, 2006.
- [9] Jean-Yves Girard. Geometry of interaction I: Interpretation of System F. *Studies in Logic and the Foundations of Mathematics*, 127:221–260, 1989.
- [10] Georges Gonthier, Martin Abadi, and Jean-Jacques Lévy. The geometry of optimal lambda reduction. In *Proceedings of the 19th ACM SIGPLAN SIGACT symposium on Principles of programming languages*, POPL ’92, pages 15–26. ACM, 1992.
- [11] Ian Mackie. The geometry of interaction machine. In *Proceedings of the 22nd ACM SIGPLAN SIGACT symposium on Principles of programming languages*, pages 198–208. ACM, 1995.
- [12] Michele Pagani, Peter Selinger, and Benoit Valiron. Applying Quantitative Semantics to Higher-Order Quantum Computing. In P. Sewell, editor, *The 41th Annual ACM SIGPLAN SIGACT Symposium on Principles of Programming Languages, POPL14, San Diego, USA*. ACM, 2014.
- [13] Jorge Sousa Pinto. Parallel Implementation Models for the Lambda-Calculus Using the Geometry of Interaction. In Samson Abramsky, editor, *Typed Lambda Calculi and Applications*, volume 2044 of *Lecture Notes in Computer Science*, pages 385–399. Springer Berlin Heidelberg, 2001.