Directed Virtual Reductions

Vincent Danos
CNRS-Université Paris 7
Paris

Marco Pedicini
CNR-Istituto per le Applicazioni del Calcolo
Roma

Laurent Regnier
CNRS-Institut de Mathématiques de Luminy
Marseille

October 12, 1996

Abstract

This note defines a new graphical local calculus, *directed virtual reductions*. It is designed to compute Girard's execution formula EX, an invariant of closed functional evaluation obtained from the "geometry of interaction" interpretation of λ-calculus [4].

The calculus is obtained by synchronizing another graphical local calculus presented in "local and asynchronous beta-reduction": *virtual reductions* [3]. This synchronization makes it easier to mechanize than general virtual reductions. In undirected virtual reductions the consistency of the computation is insured by an algebraic mechanism called the *bar*. This mechanism in general induces correction terms of any order. The directed virtual reduction has been designed to keep those terms at order one.

A further synchronization, the *combustion strategy* will even wipe out first order correction terms. Applied to sharing graphs, the combustion strategy yields Lamping's optimal graphical calculus as presented in [1]. But more efficient optimal implementations of λ-calculus are expected.

The paper is conceived as a follow-up of [3] and supposes a familiarity with virtual reduction.

1 Split dags

1.1 Digs and dags

A *dig* is a directed graph, a *dag* is a directed acyclic graph, that is a dig with no directed cycle or circuit. The *valence* of a node in a dig is the number of edges of which it is the source; the valence of a dig is the maximal valence of its nodes. A *path* in a dig is an arrow of the free involutive category generated by the dig; a *directed path* in a dig is an arrow of the free category generated by the dig. Beware that paths may take edges in the reverse direction, but directed ones may not. We adopt the following *convention* when denoting concrete paths: $\phi_2\phi_1$ is meant to denote ϕ_1 then ϕ_2, that is the product of paths is more like composition of functions than like concatenation; this allows lighter notations in the proofs. The reversion operation, which is as said an involution is denoted by *. Two paths of a dig are said to be *coincident* if they have the same target but are not prefix one of the other. A path in a dig is said to be *straight* if it contains no path of the form $\phi\phi^\star$, that is if it never bounces back.

1.2 Ims and bims

An *inverse monoid* (see [6]), or for short an *im*, is a monoid with an unary function, the *star*, denoted by $(.)^\star$, with:

$$(uv)^\star = v^\star u^\star \tag{1}$$
$$(u^\star)^\star = u \tag{2}$$
$$uu^\star u = u \tag{3}$$
$$uu^\star vv^\star = vv^\star uu^\star \tag{4}$$

We denote by $\langle u\rangle$ the idempotent $uu^\star$. With this notation the last equation becomes $\langle u\rangle\langle v\rangle = \langle v\rangle\langle u\rangle$ and the one before becomes $\langle u\rangle u = u$. One defines a compatible ordering by $u \leq v$ if $u = \langle u\rangle v$.

A *bar inverse monoid*, or for short a *bim*, is an im with a zero, denoted by 0, and an unary function, the *bar*, denoted by $[.]$, with:

$$[1] = 0 \quad \text{and} \quad [0] = 1 \tag{5}$$
$$u[v] = [uv]u \tag{6}$$

Bim's axioms entail $[u][u] = [u]$, $[u]^\star = [u]$ so $[u]$ is an idempotent, and $[u]u = 0$, $vu = 0$ iff $v[u] = v$. Think of u as a partial one-one map, of $u^\star$ as its partial inverse and of $\langle u \rangle$ as its codomain, then $[u]$ is the complementary of its codomain.

Now we give the construction of the free bim generated by a given im.

So let S be an im, and $\mathbf{Z}[S]$ denote the free contracted algebra over S with coefficients in $\mathbf{Z}$ the ring of integers. In other words $\mathbf{Z}[S]$ is the algebra of maps from S to $\mathbf{Z}$ with finitely many non-zero values. Again in other words $\mathbf{Z}[S]$ is the algebra of linear combinations over S with coefficients in $\mathbf{Z}$.

For any such linear combination, $s = \sum n_i s_i$, define:

$$s^\star = \sum n_i {s_i}^\star, \qquad [s] = 1 - \langle s \rangle = 1 - ss^\star.$$

Define the *complementary closure* of S in $\mathbf{Z}[S]$, denoted by $[S]$, to be the monoid generated in $\mathbf{Z}[S]$ by the union of S and $\{1 - \langle u \rangle,\ u \in S\}$.

Proposition 1 *$[S]$ is an inverse monoid with $(.)^\star$ defined as above.*

The proof is a straightforward calculation.

Define then the *bar closure* of S, denoted by $[S]_\omega$, to be the im obtained by ω iterations of the complementary closure, that is: let $S_0 = S$ and $S_{n+1} = [S_n]$ then $[S]_\omega = \bigcup_{n \geq 0} S_n$.

Proposition 2 *$[S]_\omega$ is a bar inverse monoid with $[.]$ defined as above.*

1.3 The "geometry of interaction" bim

Let $\mathsf{L}^\star$ denote the GoI's monoid, that is the free monoid with a morphism $!(.)$, an involution $(.)^\star$ and a zero, generated by p, q, d, r, s and t and satisfying for any $u \in \mathsf{L}^\star$:

$$\begin{array}{rcll} x^\star y &=& \delta_{xy} & \text{for } x, y = p, q, d, r, s, t, \\ !(u)d &=& du & \\ !(u)x &=& x!(u) & \text{for } x = r, s, \\ !(u)t &=& t!(!(u)) & \end{array}$$

An element, or a *monomial*, u is said *positive* if it can be written with no *'s; a's and b's will range over positive monomials of $\mathsf{L}^\star$.

Orientating the equations above from left to right, one gets a rewriting system which is terminating and confluent. The non-zero normal forms, the so-called *stable forms*, are the terms $ab^\star$ where a and b are positive. The fact that all non-zero terms are equal to such an $ab^\star$ form is referred to as the "$ab^\star$ property".

From this one easily gets that the word problem is decidable and that $\mathsf{L}^\star$ is an inverse monoid.

Every computation, from now on, will take place in the bar closure of $\mathsf{L}^\star$ in $\mathbf{Z}[\mathsf{L}^\star]$, which we denote by $[\mathsf{L}^\star]_\omega$. Since, as said, this is a bim, results of [3], which were stated and proved for any bar inverse monoid, apply with no further ado. Note that $[\mathsf{L}^\star]_\omega$ and $\mathbf{Z}[\mathsf{L}^\star]$ are also decidable by rewriting to stable form.

Set $[b_1, \ldots, b_n] = 1 - b_1 {b_1}^\star - \cdots - b_n {b_n}^\star$; $[b_1, \ldots, b_n]$ is an idempotent iff the b_i's are *orthogonal*, that is $\langle b_i \rangle \langle b_j \rangle = 0$.

Lemma 1 (superposition) *Let a, b and c be positive monomials in $\mathsf{L}^\star$ such that $\langle a \rangle \langle b \rangle$, $\langle b \rangle \langle c \rangle$ and $\langle a \rangle \langle c \rangle \neq 0$, then $\langle a \rangle \langle b \rangle \langle c \rangle \neq 0$.*

One could find a proof by using the $ab^\star$ property. However a very simple one exists because the problem can be rephrased as follows in the *superposition algebra* (a derivative of Girard's *resolution algebra* defined p. 342–346 in [5]): given three first-order terms with no variable occuring twice, if they superpose 2 by 2 then they superpose by 3, which is obvious. The proviso that variables occur only once is necessary: take $f(x, x)$, $f(a, x)$ and $f(x, b)$.

1.4 Weighted digs

Let a *weight* on a dig be a functor W from the dig's involutive category of paths to $[\mathsf{L}^\star]_\omega$. Most of the time, we will simply write ϕ for $\mathrm{W}(\phi)$ to ease the reading of definitions and proofs.

A weighted dig is said to be *split* if any three coincident paths of length one ϕ_1, ϕ_2 and ϕ_3 are such that $\langle \phi_1 \rangle \langle \phi_2 \rangle \langle \phi_3 \rangle = 0$; it is said to be *square-free* if for any straight path ϕ, $\phi\phi = 0$. A weighted dig is said to be a *virtual net* if it is split and square-free. Splitness can be rephrased as: any three paths ϕ_1, ϕ_2 and ϕ_3 such that none is prefix of another are such that $\langle \phi_1 \rangle \langle \phi_2 \rangle \langle \phi_3 \rangle = 0$.

Let α and β be two edges in a weighted dig; α and β are said to be *composable* when they are coincident and:

$$\langle \alpha \rangle \langle \beta \rangle \neq 0;$$

β is said to be a *counter-edge* of α along τ if $\beta \neq \alpha$ and τ is a directed path from α's target to β's one, not ending with β, such that (figure 1):

$$\langle\alpha\rangle\langle\tau^\star\beta\rangle \neq 0.$$

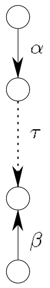

Figure 1: Counter edge and carrier

Equivalently, α and β are composable when $\alpha^\star\beta \neq 0$, and β is a counter-edge of α when $\alpha^\star\tau^\star\beta \neq 0$. Since τ might be empty, a particular case of being a counter-edge of α is simply to be composable with it.

2 Directed Virtual Nets

A *directed virtual net* R is an acyclic virtual net such that for each edge α:

A. $\alpha = [b_1, \ldots, b_n]a$, where a, b_1, $\ldots$, b_n are positive monomials of $\mathsf{L}^\star$. We will denote by α^+ the weight of α without its *filter* $[b_1, \ldots, b_n]$, that is the monomial a.

B. for any $i \neq j$ and for any two counter-edges β_1, β_2 of α along τ_1, τ_2:

$$\begin{array}{lll}
\langle b_i\rangle\langle b_j\rangle & = 0 & \mathbf{0}^R(\alpha; b_i; b_j) \\
\langle b_i\rangle\langle\tau^\star\beta^+\rangle & = 0 & \mathbf{1}^R(\alpha; b_i; \tau^\star\beta) \\
\langle\tau_1{}^\star\beta_1^+\rangle\langle\tau_2{}^\star\beta_2^+\rangle & = 0 & \mathbf{2}^R(\alpha; \tau_1{}^\star\beta_1; \tau_2{}^\star\beta_2)
\end{array}$$

Call "past impacts" the $\langle b_i\rangle$'s and "future impacts" the $\langle\tau^\star\beta^+\rangle$'s, then orthogonality clauses, our name for **B**, say: any two impacts are orthogonal.

3 First synchronization

Let R be a directed virtual net, α and β be two composable edges in R of respective weights $[b_1, \ldots, b_n]a$ and $[a_1, \ldots, a_m]b$.

Let a' and b' be the unique positive stable monomials such that $b^\star a = a'b'^\star$.

As depicted on figure 2, one defines a new directed virtual net R' by:

1. decreasing α to $[b_1, \ldots, b_n, b]a$ and β to $[a_1, \ldots, a_m, a]b$;

2. adding to R two new edges α' and β' linking the same new node to the source of β and α respectively, with weights $\alpha' = a'$ and $\beta' = b'$.

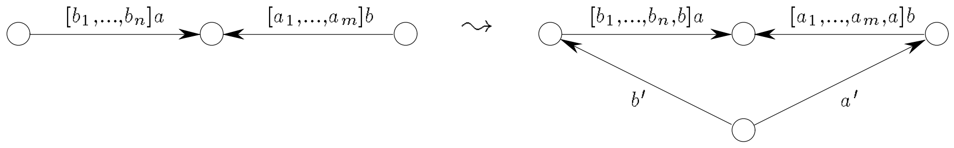

Figure 2: Directed virtual reduction

That R' indeed is a directed virtual net remains to be verified. Let us postpone for a while that verification, and check right now that:

Proposition 3 *Directed virtual reduction of a directed virtual net is a particular case of virtual reduction (as defined in [3]).*

Indeed, the new weight of α under a virtual reduction is:

$$\begin{aligned}(1 - bb^\star[a_1, \ldots, a_m])[b_1, \ldots, b_n]a &= [b_1, \ldots, b_n](a - bb^\star[a_1, \ldots, a_m]a) \\ &= [b_1, \ldots, b_n](a - bb^\star a) \\ &= [b_1, \ldots, b_n](1 - bb^\star)a \\ &= [b_1, \ldots, b_n, b]a\end{aligned}$$

because $\mathbf{1}^R(\beta; a_i; \alpha)$ says $a_i a_i{}^\star a = 0$, $\mathbf{1}^R(\alpha; b_j; \beta)$ says $b_j b_j{}^\star b = 0$, $\mathbf{0}^R(\alpha; b_i; b_j)$ says $[b_1, \ldots, b_n]$ is idempotent and $\mathbf{0}^R(\beta; a_i; a_j)$ says $[a_1, \ldots, a_m]$ is idempotent as well. Thus the new weights of α in both flavors of virtual reduction coincide, and the symmetric computation for the new weights of β would show just the same.[1]

[1]We also see in passing that $\langle\alpha\rangle_{R'} < \langle\alpha\rangle_R$, and more precisely $\langle\alpha\rangle_{R'} = \langle\alpha\rangle_R[b]$, so in this sense α strictly decreases under a directed virtual reduction step.

And using the same clauses we get the weight of $\beta^\star\alpha$:

$$\begin{array}{rcl} b^\star[a_1,\ldots,a_m][b_1,\ldots,b_n]a & = & b^\star[b_1,\ldots,b_n][a_1,\ldots,a_m]a \\ & = & b^\star a, \end{array}$$

or in other words $\beta^\star\alpha = \beta^{+\star}\alpha^+$.

So we deduce that R' is a virtual net, since split and square-free are preserved by virtual reduction as was proved in [3], and directed virtual reduction is just plain virtual reduction, so says the proposition above. *But*, it remains to be checked that it is a *directed* virtual net:

Proposition 4 *The directed virtual reduction of a directed virtual net yields again a directed virtual net.*

The dig R' is clearly a dag where **A** holds. So, it remains to be seen whether **B** also does. To ease the reading of the proof below we shall use subscripts such as ϕ_R or $\langle\phi\rangle_R$ when we need to insist on the fact that some computation takes place in R.

B. So we proceed to verifying **B** in R'. Let then an edge γ be given. By symmetry, it will be enough to consider the following three different cases: $\gamma = \alpha$; γ is none of α, α', β and β'; and $\gamma = \alpha'$. Before inspecting each three cases, we prove a lemma which we will need in the third case.

Lemma 2 (augmentation) *Let R be a directed virtual net and γ_2 be a counter-edge of γ_1 along τ in R, then:*

$$\langle\gamma_1\rangle\langle\tau^\star\gamma_2\rangle = \langle\gamma_1^+\rangle\langle\tau^{+\star}\gamma_2^+\rangle \neq 0,$$

or equivalently:

$$\gamma_1{}^\star\tau^\star\gamma_2 = \gamma_1^{+\star}\tau^{+\star}\gamma_2^+ \neq 0.$$

By induction on τ. If τ is empty, then this is the computation made when proving proposition 1, which says directed virtual reduction is a particular case of virtual reduction. If $\tau = \tau_1\delta$, then $\delta^\star\tau_1{}^\star\gamma_2 \neq 0$, so by induction $\tau^\star\gamma_2 = \tau^{+\star}\gamma_2^+$. We then get: $\langle\gamma_1\rangle\langle\tau^\star\gamma_2\rangle = \langle\gamma_1\rangle\langle\tau^{+\star}\gamma_2^+\rangle$ and by $\mathbf{1}^R(\gamma_1;.;\gamma_2)$ we conclude.

B.1. Consider first the case where $\boxed{\gamma = \alpha}$, and let γ_1 be a counter-edge of α along τ in R'.

First, τ contains none of α' and β' because none are accessible to directed paths, and none of α and β because this would imply a circuit. Hence τ exists in R as well and $\tau_{R'} = \tau_R$.

Second, γ_1 itself is neither α, because τ would be a circuit, nor β because then either τ is again a circuit, or it is empty, but then $\langle\alpha\rangle_{R'}\langle\beta\rangle_{R'} = 0$ and $\gamma_1 = \beta$ would not be a counter-edge of α in R'. Neither could γ_1 be α' or β' because $\beta\tau$ or $\alpha\tau$ would be a circuit.

Therefore $\langle\tau^\star\gamma_1^+\rangle_{R'} = \langle\tau^\star\gamma_1^+\rangle_R$. And from this, all orthogonality clauses for α in R' follow, except for $\mathbf{0}^{R'}(\alpha; b_i; b)$ which was $\mathbf{1}^R(\alpha; b_i; \beta)$, and $\mathbf{1}^{R'}(\alpha; b; \gamma_1)$ which was $\mathbf{2}^R(\alpha; \beta; \gamma_1)$.

B.2. Next, consider the case where $\boxed{\gamma \neq \alpha,\ \alpha',\ \beta \text{ and } \beta'}$ and let γ_1 be a counter-edge of γ along τ in R'. As in **B.1** τ contains none of α' and β' (but now it might contain one of α and β), hence τ exists in R as well and $\tau_{R'} \leq \tau_R$.

If $\underline{\gamma_1 \neq \alpha', \beta'}$: then $\langle\tau^\star\gamma_1^+\rangle_{R'} \leq \langle\tau^\star\gamma_1^+\rangle_R$;[2] if $\underline{\gamma_1 = \alpha'}$: since $\alpha'^+ = \alpha'$ and $\langle\alpha'\rangle_{R'} = \langle\beta^\star\alpha^+\rangle_R$, we get $\langle\tau^\star\gamma_1^+\rangle_{R'} = \langle\tau^\star\alpha'\rangle_{R'} \leq \langle\tau^\star\beta^\star\alpha^+\rangle_R$.[3]

Thus all orthogonality clauses for γ in R' follow from orthogonality clauses for γ in R.

B.3. Last, consider the case where $\boxed{\gamma = \alpha'}$. Note that $\mathbf{2}^{R'}(\alpha'; .; .)$ is the only possible orthogonality clause for α' in R' to check. Let γ_1 be a counter-edge of α' along τ in R'. The tactics consist, roughly speaking, in showing that the impact $\langle\tau^\star\gamma_1\rangle_{R'}$ is equal to its augmentation $\langle\tau^{+\star}\gamma_1^+\rangle_{R'}$ via the augmentation lemma applied in R, where we have it, then to use splitness of R', and to conclude by superposition.

Let us distinguish four subcases in evaluating $\langle\tau^\star\gamma_1\rangle_{R'}$: $\gamma_1 = \beta'$; $\gamma_1 \neq \beta'$, α, β; $\gamma_1 = \alpha$; $\gamma_1 = \beta$.

B.3.a. $\underline{\gamma_1 = \beta'}$: we have $\langle\tau^\star\beta'\rangle_{R'} = \langle\tau^\star\alpha^\star\beta\rangle_R$ because τ contains none of α', β' as usual, nor does it contain α, because there would be a circuit, or β, because there would also be a circuit since the first edge of τ, if there is one, is *not* β. The last point simply because $(\beta\alpha')_{R'} = 0$ by the following

[2] Arguing by augmentation we even have $\langle\tau^\star\gamma_1^+\rangle_{R'} = \langle\tau^\star\gamma_1^+\rangle_R$.

[3] Note that in this case τ may not contain α or β either, since this would imply a circuit, thus $\tau_{R'} = \tau_R$, and in fact $\langle\tau^\star\alpha'\rangle_{R'} = \langle\tau^\star\beta^\star\alpha^+\rangle_R$.

computation:

$$\begin{aligned} [a_1, \ldots, a_n, a]ba' &= ba' - a_1 a_1{}^\star ba' - \cdots - aa^\star ba' \\ &= ba' - 0 - \cdots - ba' \\ &= 0 \end{aligned}$$

since $ba' = bb^\star ab'$, $a_i a_i{}^\star ba' = a_i a_i{}^\star bb^\star ab' = bb^\star a_i a_i{}^\star ab' = 0$, by the usual clauses.

But then by augmentation in R on the first edge of $\alpha\tau$, possibly α itself, we get: $\tau^\star\alpha^\star\beta = \tau^{+\star}\alpha^{+\star}\beta^+ = \tau^{+\star}\beta'\alpha'^\star$, whence: $\langle\tau^\star\beta'\rangle_{R'} = \langle\tau^{+\star}\beta'^+\rangle_{R'}$.

B.3.b. $\underline{\gamma_1 \neq \beta',\ \alpha,\ \beta}$: again we have $\langle\tau^\star\gamma_1\rangle_{R'} = \langle\tau^\star\gamma_1\rangle_R$ provided τ contains none of α', β', α, β. As before, τ does not contain α' or β', nor does it contain β for the same reason as in **B.3.a**. But it might contain α, yet in this case γ_1 is also a counter-edge of α in R along a suffix say τ_1 of τ, and therefore: $\langle\tau^\star\gamma_1\rangle_{R'} = \langle\tau^\star\gamma_1\rangle_R$, thanks to $\mathbf{2}^R(\alpha; \beta; \tau_1{}^\star\gamma_1)$.

If τ is not empty, then again by augmentation in R on the first edge of τ, we get $\langle\tau^\star\gamma_1\rangle_R = \langle\tau^{+\star}\gamma_1^+\rangle_R$, whence: $\langle\tau^\star\gamma_1\rangle_{R'} = \langle\tau^{+\star}\gamma_1^+\rangle_{R'}$.

If τ is empty, we argue differently: $\langle\alpha'\rangle_{R'}\langle\gamma_1\rangle_{R'} = \langle\beta^\star\alpha\rangle_R\langle\gamma_1\rangle_R$ which means α was a counter-edge of γ_1 along β in R and the augmentation lemma in R says $\langle\beta^\star\alpha\rangle_R\langle\gamma_1\rangle_R = \langle\beta^\star\alpha\rangle_R\langle\gamma_1^+\rangle_R$, whence: $\langle\alpha'\rangle\langle\gamma_1\rangle_{R'} = \langle\alpha'\rangle\langle\gamma_1^+\rangle_{R'}$.

B.3.c. $\underline{\gamma_1 = \alpha}$: in this case, τ can't be empty, otherwise β is a loop. Again it does not contain α' or β', nor does it contain β for the same reason as ever, nor α because there would be a circuit. Thus $\tau_{R'} = \tau_R$, and we are in position of using $\mathbf{2}^R(\delta; \tau_1{}^\star\alpha; \tau_1{}^\star\beta)$, where $\tau_1\delta = \tau$ as a path, which says: $\langle\tau_1{}^\star\alpha\rangle\langle\tau_1{}^\star\beta\rangle_R = 0$, so that a fortiori $\langle\tau\rangle\langle\alpha\rangle\langle\beta\rangle_R = 0$. From this follows that: $\langle\tau^\star\alpha\rangle_{R'} = \langle\tau^\star\alpha\rangle_R$. By augmentation on the first edge of τ: $\langle\tau^\star\alpha\rangle_R = \langle\tau^{+\star}\alpha^+\rangle_R$, whence: $\langle\tau^\star\alpha\rangle_{R'} = \langle\tau^{+\star}\alpha^+\rangle_{R'}$. For $\underline{\gamma_1 = \beta}$, just the same treatment as for $\gamma_1 = \alpha$ applies.

Finale. In all cases we got:

$$\langle\alpha'\rangle\langle\tau^\star\gamma_1\rangle_{R'} = \langle\alpha'\rangle\langle\tau^{+\star}\gamma_1^+\rangle_{R'}.$$

Set $\tau = \tau_1$ and let γ_2 be another counter-edge of α' along say τ_2, then by splitness of R', which we may invoke because none our $\tau_i{}^\star\gamma_i$'s can be prefix of the other: $\langle\alpha'\rangle\langle\tau_1{}^\star\gamma_1\rangle\langle\tau_2{}^\star\gamma_2\rangle = 0$, hence $\langle\alpha'\rangle\langle\tau_1^{+\star}\gamma_1^+\rangle\langle\tau_2^{+\star}\gamma_2^+\rangle = 0$, and by the superposition lemma

$$\langle\tau_1^{+\star}\gamma_1^+\rangle\langle\tau_2^{+\star}\gamma_2^+\rangle = 0,$$

what we wanted.

3.1 When computation ends

Proposition 5 (Invariance of EX) *The execution EX is an invariant of directed virtual reduction.*

This because EX is an invariant of virtual reduction and, as stated in proposition 3, directed virtual reduction is a particular instance of virtual reduction.

Proposition 6 (Confluence) *Directed virtual reduction is strongly locally confluent, thus confluent.*

Critical pairs occur when an edge α is composable with two edges β_1 and β_2 and we have already seen that the result of the two compositions is independent of the order of these compositions. Therefore directed virtual reduction is strongly locally confluent (local confluence diagrams are squares) thus confluent.

Proposition 7 (When computation ends) *A directed virtual net where no composition applies anymore is normal, that is EX can be read off from it with no cost.*

The next section presents a particular strategy for directed virtual reduction for which, at any stage of the reduction, the edges crossed by the execution paths are weighted by a positive monomial, with no filter. Thus this is true for the normal form; but in the normal form an execution path can only cross some reverse edges for a while, then change direction once and cross some edges for another while and stop. By definition of normal, no other change of direction may occur so that the path has a weight which is in $ab^\star$ form with no computation at all.

4 Further synchronization

4.1 Combustion strategy

Let's call *full* directed virtual net a directed virtual net in which each edge is either *ghost*, i.e., has no counter edge, or is weighted by a positive monomial. Typically, a proof-net is translated into a full directed virtual net.

From now on the valence of a node will be the number of non ghost edges exiting this node.

Proposition 8 *The* combustion strategy *chooses a node of valence* 0 *and performs all the possible compositions. If R' is obtained from R by the combustion strategy and if R is full then so is R'.*

Suppose $\mathtt{n}$ is a node in R with two entering composable edges α and β. Then $\mathtt{n}$ is of valence 0 otherwise there is an exiting edge γ which has for weight the positive monomial c and this contradicts splitness (because $c^\star c = 1$). Therefore composable edges meet on nodes of valence 0.

Consider a node $\mathtt{n}$ with at least two composable edges and let α be one of them. Perform all the compositions of α with all the β_i's which are composable with it. After that the weight of α is $[b_1, \ldots, b_n]a$ where a is the weight of α and b_i is the weight of β_i before composition so that no edge at $\mathtt{n}$ is composable with α anymore. But $\mathtt{n}$ has valence 0 which means that any exiting edge from $\mathtt{n}$ has no counter edge. Therefore α has no counter edge either and is ghost.

If following the combustion strategy we perform each composition at $\mathtt{n}$ then all the old edges become ghosts and, this is the very property of directed virtual reduction, all the new ones have a weight which is a positive monomial, with *no filter*. So we proved that the reduct of a full net is full.

4.2 Sharing reductions and directed virtual reduction

Let us define a translation from sharing graphs as defined in [1] into directed virtual nets. We orientate edges so that they *exit* the operators by their principal ports and *enter* the operators by their auxiliary ports (for this to be possible one has to add some new nodes, typically split in two an edge linking two principal ports, a so-called *cut*), then label each edge in the following way:

- by $!^n(d)$ if it is the premise of a croissant node at level n, that is if its goal is a croissant node at level n;
- by $!^n(r)$ (resp. $!^n(s)$) if it is the left (resp. right) premise of a fan node at level n;
- by $!^n(t)$ if is the premise of a bracket node at level n;
- by 1 if it is not the premise of any operator (e.g., its goal is one of these new nodes added by the orientation phase),

and finally replace each operator by a simple node and identify nodes which are linked by an edge labeled by 1. When the sharing graph comes from a proof-net of linear logic, the graph obtained in this way is the weighted proof-net.

Proposition 9 *Let R be the translation of a sharing graph G, then R's combustions are isomorphical to the sharing reductions of G.*

Call *cut node* a node where composable edges enter. A cut node corresponds to a redex in the sharing graph. It is then straightforward to check that the combustion of the cut node is isomorphical to the sharing reduction of the corresponding redex.

4.3 An alternative to sharing reductions

In the translation of a linear net, one may, before performing any reduction, amalgamate each maximal oriented path into a single edge, which is then still weighted by a positive monomial. This is a particular case of virtual reduction. The net obtained still satisfies the axioms of directed virtual nets and therefore the combustion strategy applies.

This is an optimization of sharing reduction in which several redexes are fired in one shot. How good this optimization is remains to be tested.

References

[1] Martin Abadi, Georges Gonthier, and Jean-Jacques Lévy. Linear logic without boxes. In *Proceedings of the 7^{th} Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1992.

[2] Andrea Asperti. $\delta\circ!\epsilon = 1$. In Hsiang (Jieh), editor, *Proceedings of the Sixth Rewriting Techniques and Applications Conference*, pages 102–116. Springer-Verlag, 1995.

[3] Vincent Danos and Laurent Regnier. Local and asynchronous beta-reduction. In *Proceedings of the 8^{th} Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1993.

[4] Jean-Yves Girard. Geometry of interaction I: an interpretation of system F. In Ferro & al., editor, *Proceedings of A.S.L. Meetings*. North-Holland, 1988.

[5] Jean-Yves Girard. Geometry of interaction III: acommodating the additives. In Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1995.

[6] Mario Petrich. *Inverse Semigroups.* Pure and Applied Mathematics. John Wiley & sons, 1984.