A Token Machine for Full Geometry of Interaction
(Extended Abstract)

Olivier LAURENT

Institut de Mathématiques de Luminy
163, avenue de Luminy - case 907
13288 MARSEILLE cedex 09 FRANCE
olaurent@iml.univ-mrs.fr

Abstract. We present an extension of the Interaction Abstract Machine (IAM) [10, 4] to full Linear Logic with Girard's Geometry of Interaction (GoI) [6]. We propose a simplified way to interpret the additives and the interaction between additives and exponentials by means of *weights* [7]. We describe the interpretation by a token machine which allows us to recover the usual MELL case by forgetting all the additive information.

The Geometry of Interaction (GoI), introduced by Girard [5], is an interpretation of proofs (programs) by bideterministic automata, turning the global cut elimination steps (β-reduction) into local transitions [10, 4]. Because of its local feature, the GoI has proved to be a useful tool for studying the theory and implementation of optimal reduction of the λ-calculus [8, 2]. It is also strongly connected to some work on games semantics for Linear Logic and PCF ([1, 3] for example). Maybe the most exciting use of the locality of GoI is the current work, aiming at using it for implementing some parallel execution schemes [11].

Although the GoI has been very present in various works, its most popular version only deals with the MELL fragment of Linear Logic (which is sufficient for encoding the λ-calculus though). Girard proposed an extension of GoI to the additives [6], but his solution is quite technical (it makes an important use of an equivalence relation and entails a complex interpretation of the exponentials), which is probably the reason why it had not the same success as the MELL case. A first difficulty is to manage the "non linearity" of the additive cut elimination step (see below). A subtler problem comes from the interaction between additives and exponentials; in particular we will see that the weakening rule becomes very tricky to handle in presence of additives. Note that this is linked to the problem (still open at the time of this writing) of finding a good proof-net syntax for full LL.

In this paper, although we don't claim to give the final word on the GoI for LL, we propose an interpretation with an abstract machine, based on additive *weights* [7]. This greatly simplifies Girard's interpretation since we don't have to work up to isomorphism. We hope that this will prove to be a determining step towards the extension to additives of optimal reduction, games, ...

Additive linearity. A naive look at the usual $\&/\oplus$ cut elimination step of MALL shows that this reduction step is not so "linear": the sub-proof π_2 is completely erased.

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma, A} \quad \dfrac{\pi_2}{\vdash \Gamma, B}}{\vdash \Gamma, A \,\&\, B}\,\& \quad \dfrac{\dfrac{\pi_3}{\vdash \Delta, A^\perp}}{\vdash \Delta, A^\perp \oplus B^\perp}\,\oplus_1}{\vdash \Gamma, \Delta}\,cut \;\rightarrow\; \dfrac{\dfrac{\pi_1}{\vdash \Gamma, A} \quad \dfrac{\pi_3}{\vdash \Delta, A^\perp}}{\vdash \Gamma, \Delta}\,cut$$

This is too drastic to be interpreted by the very local approach of GoI. To "linearize" this cut elimination step, Girard introduced the $\flat$-rules [6], in the calculus $\mathsf{LL}^\flat$, which allow to keep the proof π_2 after reduction but marked with a $\flat$ symbol.

Soundness. By the modification of the additive reduction in $\mathsf{LL}^\flat$, we get the preservation of the GoI by $\flat$-reduction. Although an $\mathsf{LL}^\flat$-normal form is not an LL-normal form (it still has some $\flat$-rules), one can easily extract the latter from the former. This extraction procedure erases parts of the proof that had been memorized along the $\flat$-reduction thus it doesn't respect the GoI interpretation. This is why we will only show a soundness result for LL-proofs of $\vdash 1 \oplus 1$.

Proofs of $\vdash 1 \oplus 1$ give an encoding of booleans since there are exactly two normal proofs of this sequent in LL. The restriction to these boolean results is very drastic but sufficient, from a computational point of view, to distinguish different results (see [5] for a longer discussion).

Without clearly decomposing LL cut elimination into these two steps ($\flat$-reduction and extraction), the study of the modifications of the GoI interpretation during LL-reduction would be very complicated and the results very difficult to express and to prove. Moreover this precise analysis allows us to introduce a parallel version of the automaton which leads to simpler results in both steps (Propositions 1 and 2). This parallel approach may probably also be used to define GoI for the system LLP [9] for classical logic which contains generalized structural rules.

Additives and weakening. The other main technical (and complicated) point is the interaction between additives and exponentials, in particular the interaction with weakening (or $\perp$). According to its erasing behavior, the usual interpretation of a weakened formula is empty. In an additive setting, this idea leads to an inconsistency:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_1}{\vdash 1 \oplus 1, \perp}\,\perp \quad \dfrac{\dfrac{\dfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_2}{\vdash 1 \oplus 1, \perp}\,\perp}{\vdash 1 \oplus 1, \perp \,\&\, \perp}\,\& \quad \dfrac{\dfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_i}{\vdash 1 \oplus 1}\,cut$$

there is no way to know if the $\oplus_1$ proof is "attach" to the left or to the right part of the &, if $\perp$ is empty. Thus the GoI interpretation of this proof doesn't depend on the value of i which is crucial since it determinates the boolean corresponding to the normal form. To solve this problem, we have to modify the

weakening rule by attaching the weakened formula to a formula in the context (which corresponds to encoding $\bot$ by $\exists\alpha(\alpha \otimes \alpha^\bot)$, see [6]) ensuring that an explicit information in the GoI interpretation indicates which $\oplus$ is in the left and in the right.

Sequent calculus vs. proof-nets. The idea of GoI comes from the geometric representation of proofs given by proof-nets [7]. However, the technology of proof-nets for additives is not completely satisfactory, in particular because there is no good cut elimination procedure. Moreover using proof-nets would require a definition of $\flat$-proof-nets. For these reasons, we will interpret proofs in sequent calculus and prove our results for this interpretation but it is easy to define the interpretation of proof-nets while not talking about cut elimination.

The presentation is done in three distinct steps: first the MALL case, then we add the constants and eventually we obtain the full case by adding the exponentials. In this way, it is easier to see the modularity of the construction and to see which part of the interpretation corresponds to which subpart of Linear Logic. By forgetting the adequate constructions, we can easily obtain GoI for various fragments of LL, in particular we recover the usual IAM [4] for MELL.

1 Sequent Calculus MALL$^\flat$

To give the interpretation of proofs, we have to be very precise about the distinct occurrences of formulas. This is why we introduce annotations with indexes in the rules of the sequent calculus.

1.1 Usual MALL Sequent Calculus

$$\frac{}{\vdash A, A^\bot}\, ax \qquad \frac{\vdash \Gamma_1, A \qquad \vdash A^\bot, \Delta_1}{\vdash \Gamma, \Delta}\, cut$$

$$\frac{\vdash \Gamma_1, A \qquad \vdash \Delta_1, B}{\vdash \Gamma, \Delta, A \otimes B} \otimes \qquad \frac{\vdash \Gamma_1, A, B}{\vdash \Gamma, A \parr B} \parr$$

$$\frac{\vdash \Gamma_1, A \qquad \vdash \Gamma_2, B}{\vdash \Gamma, A \,\&\, B} \,\& \qquad \frac{\vdash \Gamma_1, A}{\vdash \Gamma, A \oplus B} \oplus_1 \qquad \frac{\vdash \Gamma_1, B}{\vdash \Gamma, A \oplus B} \oplus_2$$

1.2 $\flat$-Rules

We have to introduce a new symbol $\flat$, for marking some "partial" sequents in proofs, this is not a formula and thus no connective can be applied on it.

$$\frac{}{\vdash \Gamma, \flat}\, \flat \qquad \frac{\vdash \Gamma_1, \Delta_1 \qquad \vdash \Gamma_2, \flat \qquad \vdash \Delta_2, \flat}{\vdash \Gamma, \Delta}\, s^\flat$$

The two $\flat$-premises of the $s^\flat$-rule are used to memorize some sub-proofs through the additive reduction step (see Sect. 1.4).

A proof of a sequent containing the symbol $\flat$ is a kind of partial proof where some sub-proof is missing.

Definition 1 (Weight). *Given a set of* elementary weights*, i.e., boolean variables, a* basic weight *is an elementary weight p or a negation of an elementary weight $\bar{p}$ and a* weight *is a product (conjunction) of basic weights.*

As a convention, we use 1 for the empty product and 0 for a product where p and $\bar{p}$ appear. We also replace $p.p$ by p and $\bar{\bar{p}}$ by p. With this convention we say that the weight w depends on *p when p or $\bar{p}$ appears in w.*

We use the notations $w(p)$ (resp. $w(\bar{p})$) if p (resp. $\bar{p}$) appears in w and $w(\not{p})$ if w doesn't depend on p. The product of weights is denoted by $w.w'$.

We will consider weighted proofs, i.e., with a basic weight associated to each &-rule and to each $s^\flat$-rule. These two kinds of rules are called *sided rules*. For a &-rule, the sub-proof of the left (resp. right) premise is called its left (resp. right) side and for a $s^\flat$-rule the sub-proof of $\vdash \Gamma_1, \Delta_1$ is the left side and the sub-proofs of $\vdash \Gamma_2, \flat$ and $\vdash \Delta_2, \flat$ are the right side.

A weight describes a choice for the &-rules of one of their two premises. It corresponds to the notion of *additive slice* [7], that is the multiplicative proofs obtained by projecting each & on one of its sides.

Definition 2 (Correct weighting). *A weighted proof has a* correct weighting *when two sided rules have a basic weight corresponding to the same elementary weight only if they are in the left side and in the right side of a same sided rule (i.e., an elementary weight never appears twice in the same additive slice of a proof).*

1.3 The $\natural$-Translation

We are only interested in proofs of LL sequents (without $\flat$), $\flat$-rules are used as an intermediary step for the interpretation. This is why in the sequel we will consider only $\mathsf{LL}^\flat$ proofs of LL sequents.

There exists an easy way to transform such an $\mathsf{LL}^\flat$ proof π into an LL one $\pi^\natural$ called the $\natural$*-translation*: for LL-rules just change nothing and for each $s^\flat$-rule erase the right side and connect the left side to the conclusion.

1.4 Cut Elimination

For the $\mathsf{LL}^\flat$ sequent calculus, the cut elimination procedure is the usual one except for the additive step:

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, A} \quad \dfrac{\pi_2}{\vdash \Gamma_2, B}}{\vdash \Gamma_3, A \,\&\, B}\,\& \quad \dfrac{\dfrac{\pi_3}{\vdash \Delta_1, A^\perp}}{\vdash \Delta_2, A^\perp \oplus B^\perp}\,\oplus_1}{\vdash \Gamma, \Delta}\,cut$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, A} \quad \dfrac{\pi_3^1}{\vdash \Delta_1^1, A^\perp}}{\vdash \Gamma_3, \Delta_2}\,cut \quad \dfrac{\dfrac{\pi_2}{\vdash \Gamma_2, B} \quad \dfrac{}{\vdash B^\perp, \flat}\,\flat}{\vdash \Gamma_4, \flat}\,cut \quad \dfrac{\dfrac{\pi_3^2}{\vdash \Delta_1^2, A^\perp} \quad \dfrac{}{\vdash A, \flat}\,\flat}{\vdash \Delta_3, \flat}\,cut}{\vdash \Gamma, \Delta}\,s^\flat$$

For such a cut elimination step between a &-rule and a $\oplus_i$-rule, we can define a canonical weighting for the new proof from the one on the initial proof by associating to the $s^\flat$-rule the basic weight p if $i = 1$ and $\bar{p}$ if $i = 2$ where p is the basic weight of the &-rule.

Due to this modified reduction step, MALL is a sub-system of MALL$^\flat$ which is not stable by reduction.

Remark 1. This new additive step is now "really" linear if we consider sub-proofs with their additive weight: before reduction we have $p.\pi_1 + \bar{p}.\pi_2 + \pi_3$ and after $p.\pi_1 + \bar{p}.\pi_2 + p.\pi_3^1 + \bar{p}.\pi_3^2$. Notice that the $\flat$-premises of the $s^\flat$-rule are crucial for this purpose: one for $\bar{p}.\pi_2$ and the other one for $\bar{p}.\pi_3$.

We also have to define new commutative steps for the $s^\flat$-rule:

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, \Delta_1, C_1} \quad \dfrac{\pi_2}{\vdash \Gamma_2, C_2, \flat} \quad \dfrac{\pi_3}{\vdash \Delta_2, \flat}}{\vdash \Gamma_3, \Delta_3, C}\, s^\flat \quad \dfrac{\pi_4}{\vdash \Sigma_1, C^\perp}}{\vdash \Gamma, \Delta, \Sigma}\, cut$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, \Delta_1, C_1} \quad \dfrac{\pi_4^1}{\vdash \Sigma_1, C_1^\perp}}{\vdash \Gamma_3, \Delta_3, \Sigma_3}\, cut \quad \dfrac{\dfrac{\pi_2}{\vdash \Gamma_2, C_2, \flat} \quad \dfrac{\pi_4^2}{\vdash \Sigma_2, C_2^\perp}}{\vdash \Gamma_4, \Sigma_4, \flat}\, cut \quad \dfrac{\pi_3}{\vdash \Delta_2, \flat}}{\vdash \Gamma, \Delta, \Sigma}\, s^\flat$$

and the corresponding one for a cut on a formula in Δ.

For the other cut elimination steps, the new weighting is easy to define; when a sub-proof is duplicated, we preserve the same basic weights in the two copies.

We will now always consider proofs with correct weightings, noting that correctness is preserved by reduction.

Definition 3 (Quasi-normal form). *A proof in* LL$^\flat$ *is said to be in* quasi-normal form *if it cannot be reduced by any step described above.*

Remark 2. A proof in quasi-normal form contains only cuts in which at least one of the two occurrences of the cut formula has been introduced by a $\flat$ axiom rule and used only in *cut*-rules.

It is possible to define a general cut elimination procedure as in [6] for LL$^\flat$, but it would be more complicated and useless because we can remark that the $\natural$-translation of a proof of an LL sequent in quasi-normal form is a normal proof in LL.

2 The Interaction Abstract Machine

We now define the Interaction Abstract Machine (IAM) for MALL$^\flat$. Forgetting the additive informations gives back the multiplicative IAM [4].

2.1 Tokens and Machine

Definition 4 (Token). *For the multiplicative-additive case, a* token *is a tuple* (m, a, w) *where* m *and* a *are stacks (*ε *will denote the empty stack) built on letters* $\{g, d\}$ *(Girard's notations corresponding to the french* gauche *and* droite*) and* w *is a weight.*

Definition 5 (Abstract machine). *A* state *of the* machine M_π *associated to the proof* π *of* $\mathsf{LL}^\flat$ *is* $F^\uparrow(m, a, w)$ *or* $F^\downarrow(m, a, w)$ *or* $\emptyset$ *where* F *is an occurrence of a formula appearing in the proof and the arrow indicates if the token* (m, a, w) *is going upwards or downwards.* $\emptyset$ *means that the machine stops.*

The transitions of M_π *through the rules of* π *are described in Figs. 1 and 2.* Γ *(resp.* Δ*) is used for one of the formulas of the multiset* Γ *(resp.* Δ*), but the same before and after the transition. If the result of a transition contains a weight* $w = 0$*, we consider it as* $\emptyset$*.*

Remark 3. In Fig. 2, changing the transition in the case $A \,\&\, B^\uparrow(m, g.a, w(\not{p}))$ (resp. $A \,\&\, B^\uparrow(m, d.a, w(\not{p}))$) into $A \,\&\, B^\uparrow(m, g.a, w(\not{p})) \to A^\uparrow(m, a, w.p)$ (resp. $A \,\&\, B^\uparrow(m, d.a, w(\not{p})) \to B^\uparrow(m, a, w.\bar{p})$) would make no difference since this p (resp. $\bar{p}$) information is also added when going down through the &-rule.

2.2 Properties of the Machine

Definition 6 (Partial function on tokens). *Let* π *be a proof and* A *one of its conclusions, we define the partial function* f_π *by:*

$$f_\pi(A, (m, a, w)) = \begin{cases} (B, (m', a', w')) & \text{if the computation on } A^\uparrow(m, a, w) \text{ ends} \\ & \text{by } B^\downarrow(m', a', w') \text{ whith } B \text{ conclusion of } \pi \\ \uparrow & \text{otherwise} \end{cases}$$

The partial function f_π is undefined in two cases: either if the machine stops inside the proof or if the execution doesn't terminate.

Lemma 1. *Let* π *be a proof, and* w' *and* w_0 *two weights s.t.* $w'.w_0 \neq 0$.
$f_\pi(A, (m, a, w)) = (B, (m', a', w')) \Rightarrow f_\pi(A, (m, a, w.w_0)) = (B, (m', a', w'.w_0))$

Theorem 1 (Soundness). *If* π *is a proof in* $\mathsf{MALL}^\flat$ *whose quasi-normal form is* π_0 *then for each pair formula-token* j*:*

- *if* $f_\pi(j) = \uparrow$ *then* $f_{\pi_0}(j) = \uparrow$
- *if* $f_{\pi_0}(j) = j'$ *then* $f_\pi(j) = j'$
- *if* $f_\pi(A, (m, a, w)) = (B, (m', a', w'))$ *and* $f_{\pi_0}(A, (m, a, w)) = \uparrow$ *then there exists* w_0 *s.t.* $f_{\pi_0}(A, (m, a, w.w_0)) = (B, (m', a', w'.w_0))$ *with* $w'.w_0 \neq 0$.

Moreover, if the execution in M_π *is infinite, it is infinite in* M_{π_0}.

	ax		cut
$A^{\uparrow}(m,a,w)$	$\rightarrow A^{\perp\downarrow}(m,a,w)$	$A^{\downarrow}(m,a,w)$	$\rightarrow A^{\perp\uparrow}(m,a,w)$
$A^{\perp\uparrow}(m,a,w)$	$\rightarrow A^{\downarrow}(m,a,w)$	$A^{\perp\downarrow}(m,a,w)$	$\rightarrow A^{\uparrow}(m,a,w)$
		$\Gamma^{\uparrow}(m,a,w)$	$\rightarrow \Gamma_1^{\uparrow}(m,a,w)$
		$\Delta^{\uparrow}(m,a,w)$	$\rightarrow \Delta_1^{\uparrow}(m,a,w)$
		$\Gamma_1^{\downarrow}(m,a,w)$	$\rightarrow \Gamma^{\downarrow}(m,a,w)$
	$\otimes$	$\Delta_1^{\downarrow}(m,a,w)$	$\rightarrow \Delta^{\downarrow}(m,a,w)$
$A \otimes B^{\uparrow}(g.m,a,w)$	$\rightarrow A^{\uparrow}(m,a,w)$		
$A \otimes B^{\uparrow}(d.m,a,w)$	$\rightarrow B^{\uparrow}(m,a,w)$		$\parr$
$A \otimes B^{\uparrow}(\varepsilon,a,w)$	$\rightarrow \emptyset$	$A \parr B^{\uparrow}(g.m,a,w)$	$\rightarrow A^{\uparrow}(m,a,w)$
$A^{\downarrow}(m,a,w)$	$\rightarrow A \otimes B^{\downarrow}(g.m,a,w)$	$A \parr B^{\uparrow}(d.m,a,w)$	$\rightarrow B^{\uparrow}(m,a,w)$
$B^{\downarrow}(m,a,w)$	$\rightarrow A \otimes B^{\downarrow}(d.m,a,w)$	$A \parr B^{\uparrow}(\varepsilon,a,w)$	$\rightarrow \emptyset$
$\Gamma^{\uparrow}(m,a,w)$	$\rightarrow \Gamma_1^{\uparrow}(m,a,w)$	$A^{\downarrow}(m,a,w)$	$\rightarrow A \parr B^{\downarrow}(g.m,a,w)$
$\Delta^{\uparrow}(m,a,w)$	$\rightarrow \Delta_1^{\uparrow}(m,a,w)$	$B^{\downarrow}(m,a,w)$	$\rightarrow A \parr B^{\downarrow}(d.m,a,w)$
$\Gamma_1^{\downarrow}(m,a,w)$	$\rightarrow \Gamma^{\downarrow}(m,a,w)$	$\Gamma^{\uparrow}(m,a,w)$	$\rightarrow \Gamma_1^{\uparrow}(m,a,w)$
$\Delta_1^{\downarrow}(m,a,w)$	$\rightarrow \Delta^{\downarrow}(m,a,w)$	$\Gamma_1^{\downarrow}(m,a,w)$	$\rightarrow \Gamma^{\downarrow}(m,a,w)$

Fig. 1. Identity and multiplicative groups.

	$\&$		$\oplus_1$
$A \,\&\, B^{\uparrow}(m,g.a,w(p))$	$\rightarrow A^{\uparrow}(m,a,w(p))$	$A \oplus B^{\uparrow}(m,g.a,w)$	$\rightarrow A^{\uparrow}(m,a,w)$
$A \,\&\, B^{\uparrow}(m,g.a,w(\not p))$	$\rightarrow A^{\uparrow}(m,a,w(\not p))$	$A \oplus B^{\uparrow}(m,d.a,w)$	$\rightarrow \emptyset$
$A \,\&\, B^{\uparrow}(m,g.a,w(\bar p))$	$\rightarrow \emptyset$	$A \oplus B^{\uparrow}(m,\varepsilon,w)$	$\rightarrow \emptyset$
$A \,\&\, B^{\uparrow}(m,d.a,w(\bar p))$	$\rightarrow B^{\uparrow}(m,a,w(\bar p))$	$A^{\downarrow}(m,a,w)$	$\rightarrow A \oplus B^{\downarrow}(m,g.a,w)$
$A \,\&\, B^{\uparrow}(m,d.a,w(\not p))$	$\rightarrow B^{\uparrow}(m,a,w(\not p))$	$\Gamma^{\uparrow}(m,a,w)$	$\rightarrow \Gamma_1^{\uparrow}(m,a,w)$
$A \,\&\, B^{\uparrow}(m,d.a,w(p))$	$\rightarrow \emptyset$	$\Gamma_1^{\downarrow}(m,a,w)$	$\rightarrow \Gamma^{\downarrow}(m,a,w)$
$A \,\&\, B^{\uparrow}(m,\varepsilon,w)$	$\rightarrow \emptyset$		
$A^{\downarrow}(m,a,w)$	$\rightarrow A \,\&\, B^{\downarrow}(m,g.a,w.p)$		
$B^{\downarrow}(m,a,w)$	$\rightarrow A \,\&\, B^{\downarrow}(m,d.a,w.\bar p)$		
$\Gamma^{\uparrow}(m,a,w(p))$	$\rightarrow \Gamma_1^{\uparrow}(m,a,w(p))$		$s^{\flat}$
$\Gamma^{\uparrow}(m,a,w(\bar p))$	$\rightarrow \Gamma_2^{\uparrow}(m,a,w(\bar p))$	$\Gamma^{\uparrow}(m,a,w(p))$	$\rightarrow \Gamma_1^{\uparrow}(m,a,w(p))$
$\Gamma^{\uparrow}(m,a,w(\not p))$	$\rightarrow \emptyset$	$\Gamma^{\uparrow}(m,a,w(\bar p))$	$\rightarrow \Gamma_2^{\uparrow}(m,a,w(\bar p))$
$\Gamma_1^{\downarrow}(m,a,w)$	$\rightarrow \Gamma^{\downarrow}(m,a,w.p)$	$\Gamma^{\uparrow}(m,a,w(\not p))$	$\rightarrow \emptyset$
$\Gamma_2^{\downarrow}(m,a,w)$	$\rightarrow \Gamma^{\downarrow}(m,a,w.\bar p)$	$\Delta^{\uparrow}(m,a,w(p))$	$\rightarrow \Delta_1^{\uparrow}(m,a,w(p))$
		$\Delta^{\uparrow}(m,a,w(\bar p))$	$\rightarrow \Delta_2^{\uparrow}(m,a,w(\bar p))$
		$\Delta^{\uparrow}(m,a,w(\not p))$	$\rightarrow \emptyset$
		$\Gamma_1^{\downarrow}(m,a,w)$	$\rightarrow \Gamma^{\downarrow}(m,a,w.p)$
	$\flat$	$\Gamma_2^{\downarrow}(m,a,w)$	$\rightarrow \Gamma^{\downarrow}(m,a,w.\bar p)$
$\Gamma^{\uparrow}(m,a,w)$	$\rightarrow \emptyset$	$\Delta_1^{\downarrow}(m,a,w)$	$\rightarrow \Delta^{\downarrow}(m,a,w.p)$
		$\Delta_2^{\downarrow}(m,a,w)$	$\rightarrow \Delta^{\downarrow}(m,a,w.\bar p)$

Fig. 2. Additive and $\flat$ groups. (p is the basic weight associated to the &-rule or to the $s^{\flat}$-rule and the $\oplus_2$ is easy to define from $\oplus_1$)

The introduction of the weight w_0 corresponds to the transformation of π_3 into $p.\pi_3 + \bar{p}.\pi_3$ during additive cut elimination (see Remark 1). Before reduction we don't need any information about p to go in π_3 but after reduction we have to know if we go to $p.\pi_3$ or to $\bar{p}.\pi_3$.

Proof. We have to prove that for each step of cut elimination the theorem is true and then by an easy induction on the length of a normalization we obtain the result.

We suppose that the cut-rule which we are eliminating is the last rule of the proof π and we obtain a proof π'. If it is not the case, we just have to remark that adding the same new rules at the end of π and π' is correct with respect to the interpretation.

We only consider the case of the additive cut elimination step (figure in Sect. 1.4) which is the most important one, the others are left to the reader.

We use the notation $j = (\Gamma, t)$ or $s = \Gamma^{\uparrow}(m, a, w)$ to say that the formula we are talking about is in the multiset Γ (idem for $\Delta, \dots$). Moreover $f(\Gamma, t) = (\Gamma, t')$ doesn't necessarily mean that the formula is the same before and after the computation.

Let p be the basic weight associated to the &-rule. We study the different possible cases for j:

- if $j = (\Gamma, (m, a, w(p)))$, we look at the sequence $s_1, s_2, \dots$ (resp. $s'_1, s'_2, \dots$) of the states $F^{\uparrow}(m, a, w)$ in the conclusions of the sub-proofs π_1, π_2 and π_3 (resp. π_1, π_2, π_3^1 and π_3^2) during the computation of M_π (resp. $M_{\pi'}$) on the state associated to j. In fact these states will always be in the conclusions of π_1 and π_3 (resp. π_1 and π_3^1) with s_1 in Γ_1, more precisely:
  - if $s_{2i+1} = F^{\uparrow}(m, a, w)$ with $F = \Gamma_1$ or A, $s_{2i+2} = A^{\perp\uparrow}(m', a', w')$ with $f_{\pi_1}(F, (m, a, w)) = (A, (m', a', w'))$ or s_{2i+2} doesn't exist;
  - if $s_{2i} = A^{\perp\uparrow}(m, a, w)$, $s_{2i+1} = A^{\uparrow}(m', a', w')$ with $f_{\pi_3}(A^{\perp}, (m, a, w)) = (A^{\perp}, (m', a', w'))$ or s_{2i+1} doesn't exist.

  The same facts occur for the s'_i by replacing π_3 with π_3^1 so we have $\forall i, s_i = s'_i$. If $s_1, s_2, \dots$ is infinite, $s'_1, s'_2, \dots$ too. There are two different reasons for $s_1, s_2, \dots$ to be finite, if s_n is the last state and is in the conclusions of π_k: either the evaluation of the corresponding machine M_{π_k} is infinite (or undefined at a step) on s_n and the same thing occurs in π' or f_{π_k} on this state gives a result j' in the context and $f_\pi(j) = j'$ (idem in π').
- if $j = (\Gamma, (m, a, w(\bar{p})))$, either $f_{\pi_2}(j) = (\Gamma, t')$ and $f_\pi(j) = (\Gamma, t') = f_{\pi'}(j)$ or $f_{\pi_2}(j) = (B, t')$ and $f_\pi(j) = \uparrow = f_{\pi'}(j)$;
- if $j = (\Gamma, (m, a, w(\not{p})))$, $f_\pi(j) = \uparrow$ and $f_{\pi'}(j) = \uparrow$;
- if $j = (\Delta, (m, a, w(p)))$, similar to the $(\Gamma, (m, a, w(p)))$ case;
- if $j = (\Delta, (m, a, w(\bar{p})))$, either $f_{\pi_3}(j) = (\Delta, t')$ and $f_\pi(j) = (\Delta, t') = f_{\pi'}(j)$ or $f_{\pi_3}(j) = (A^{\perp}, t')$ and $f_\pi(j) = \uparrow = f_{\pi'}(j)$;
- if $j = (\Delta, (m, a, w(\not{p})))$, $f_{\pi'}(j) = \uparrow$ but $f_\pi(j)$ may be defined with $f_\pi(j) = (F, (m', a', w'))$ in this situation we have by Lemma 1 (noting that $w'.p \neq 0$ for a correct weighting) and by applying the case $j = (\Delta, (m, a, w(p)))$, $f_{\pi'}(\Delta, (m, a, w.p)) = (F, (m', a', w'.p)) = f_\pi(\Delta, (m, a, w.p))$. This case is

very important because it is characteristic of the fact that f_π and $f_{\pi'}$ may differ. □

Corollary 1. *If w is a weight s.t. for all elementary weight p of π, $p \in w$ or $\bar{p} \in w$ then $f_\pi(F,(m,a,w)) = f_{\pi_0}(F,(m,a,w))$.*

Theorem 2 (Termination). *Let π be a proof, A a conclusion of π and t a token, the execution of the machine on $A^\uparrow(t)$ terminates.*

Proof. Let π_0 be a quasi-normal form of π. By Theorem 1, we have to prove that the execution of the machine associated to π_0 on $A^\uparrow(t)$ terminates.

In π_0, if the execution never uses the transition of a cut formula, either it stops in a transition or it goes up to an axiom and then down to a conclusion so it terminates. Moreover, by the definition of a quasi-normal form, the *cut*-rules appearing in π_0 are of the form:

$$\dfrac{\dfrac{\vdots}{\vdash \Gamma, A} \qquad \dfrac{\dfrac{}{\vdash A^\perp, \flat}\,\flat}{\vdots}\,cut}{\vdash \Gamma, \flat}\,cut$$

and if the execution uses the transition on A in such a *cut*-rule, it stops in the $\flat$-rule. Thus the evaluation is always finite in a quasi-normal form. □

2.3 The Parallel IAM

In order to complete some transitions on which the IAM stops, we can introduce a parallel version of the machine for which states are formal sums of states of the IAM with 0 for the empty sum. To define the parallel machine M_π^p associated to a proof π, we modify some particular transitions and we replace $\emptyset$ by 0:

$$\begin{array}{l} \overset{\&}{} \\ \Gamma^\uparrow(m,a,w(\not{p})) \to \Gamma_1^\uparrow(m,a,w.p) + \Gamma_2^\uparrow(m,a,w.\bar{p}) \end{array}$$

$$\begin{array}{l} \overset{s^\flat}{} \\ \Gamma^\uparrow(m,a,w(\not{p})) \to \Gamma_1^\uparrow(m,a,w.p) + \Gamma_2^\uparrow(m,a,w.\bar{p}) \\ \Delta^\uparrow(m,a,w(\not{p})) \to \Delta_1^\uparrow(m,a,w.p) + \Delta_2^\uparrow(m,a,w.\bar{p}) \end{array}$$

We denote by f_π^p the partial function associated to this machine and defined like f_π (Definition 6). To simplify the results (formal sums of pairs formula-token), we use the following rewriting rule:

$$(A,(m,a,w.p)) + (A,(m,a,w.\bar{p})) \quad \to \quad (A,(m,a,w))$$

Proposition 1 (Parallel soundness). *If π is a proof whose quasi-normal form is π_0 then $f_\pi^p = f_{\pi_0}^p$.*

When a weight information is missing, the parallel machine tries all the possibilities thus it doesn't need any starting information. This is why the requirement of an additional weight w_0 in Theorem 1 disappears.

3 Adding the Constants

3.1 Rules and Machine

$$\frac{}{\vdash 1}\,1 \qquad \frac{\vdash \Gamma_1, A_1}{\vdash \Gamma, A, \bot}\,\bot \qquad \frac{\vdash \Gamma_1, \flat}{\vdash \Gamma, \top}\,\top$$

As explained in the introduction, we have to modify the $\bot$-rule by distinguishing a particular formula in the context.

We can extend the $\natural$-translation without any loss of its properties by replacing each $\top$-rule by the usual one $\dfrac{}{\vdash \Gamma, \top}$ and by erasing everything above it.

For the multiplicative constants, the cut elimination is as usual. For the additive constants, we obtain:

$$\dfrac{\dfrac{\vdash \Gamma_2, A_1, \flat}{\vdash \Gamma_1, A, \top_1}\,\top \qquad \vdash \Delta_1, A^\perp}{\vdash \Gamma, \Delta, \top}\,cut \quad \rightarrow \quad \dfrac{\vdash \Gamma_2, A, \flat \qquad \vdash \Delta_2, A^\perp}{\dfrac{\vdash \Gamma_1, \Delta_1, \flat}{\vdash \Gamma, \Delta, \top}\,\top}\,cut$$

We extend the notion of token by using the letters $\{g, d, \Uparrow, \Downarrow\}$ for the multiplicative stack and we add new transitions for the added rules (Fig. 3).

	1		$\bot$	
$1^\uparrow(m,a,w)$	$\rightarrow 1^\downarrow(m,a,w)$	$A^\uparrow(m,a,w)$	$\rightarrow \bot^\downarrow(\Uparrow.m,a,w)$	
		$A_1^\downarrow(m,a,w)$	$\rightarrow \bot^\downarrow(\Downarrow.m,a,w)$	
		$\bot^\uparrow(\Uparrow.m,a,w)$	$\rightarrow A_1^\uparrow(m,a,w)$	
	$\top$	$\bot^\uparrow(\Downarrow.m,a,w)$	$\rightarrow A^\downarrow(m,a,w)$	
$\top^\uparrow(m,a,w)$	$\rightarrow \top^\downarrow(m,a,w)$	$\bot^\uparrow(m,a,w)$	$\rightarrow \emptyset$	$m \neq \Uparrow.m', \Downarrow.m'$
$\Gamma^\uparrow(m,a,w)$	$\rightarrow \Gamma_1^\uparrow(m,a,w)$	$\Gamma^\uparrow(m,a,w)$	$\rightarrow \Gamma_1^\uparrow(m,a,w)$	
$\Gamma_1^\downarrow(m,a,w)$	$\rightarrow \Gamma^\downarrow(m,a,w)$	$\Gamma_1^\downarrow(m,a,w)$	$\rightarrow \Gamma^\downarrow(m,a,w)$	

Fig. 3. Constant group.

Theorem 1.a (Soundness *continued*). *The Theorem 1 is still true in* $\mathsf{MALL}^\flat$ *with constants.*

3.2 Computation of Booleans

We want to compute results for the usual cut elimination procedure of LL. As already explained, we have to restrict ourselves to the particular case of proofs of $\vdash 1 \oplus 1$ that give a notion of booleans.

Lemma 2. *If* π *is a proof of* $\vdash 1$, *there exists* w *s.t.* $f_\pi(1,(\varepsilon,\varepsilon,w)) = (1,(\varepsilon,\varepsilon,w))$.

Lemma 3. *If* π *is a proof of* $\vdash 1 \oplus 1, \flat$ *then, for any* j, $f_\pi(j) = \uparrow$.

Theorem 3. *If* π *is a proof of* $\vdash 1 \oplus 1$ *whose quasi-normal form is* π_0 *then* $\pi_0^\natural = \dfrac{\dfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_i$ *and*

– *either there exists* w *s.t.* $f_\pi(1 \oplus 1, (\varepsilon, g, w)) = (1 \oplus 1, (\varepsilon, g, w))$ *and* $i = 1$
– *or there exists* w *s.t.* $f_\pi(1 \oplus 1, (\varepsilon, d, w)) = (1 \oplus 1, (\varepsilon, d, w))$ *and* $i = 2$.

Proof. We suppose that $i = 1$ and we make an induction on π_0.

– If the last rule is $\oplus_k$ it must be a $\oplus_1$ by normalization; we apply the Lemma 2 to the premise which gives us a weight w s.t. $f_\pi(1 \oplus 1, (\varepsilon, g, w)) = (1 \oplus 1, (\varepsilon, g, w))$. Moreover for any weight w', $f_\pi(1 \oplus 1, (\varepsilon, d, w')) = \uparrow$.
– If the last rule is $s^\flat$, let p be its basic weight. We can apply the induction hypothesis to the sub-proof π_0' of $\vdash 1 \oplus 1$ and we obtain a weight w s.t. $f_{\pi_0'}(1 \oplus 1, (\varepsilon, g, w)) = (1 \oplus 1, (\varepsilon, g, w))$ so $f_{\pi_0}(1 \oplus 1, (\varepsilon, g, w.p)) = (1 \oplus 1, (\varepsilon, g, w.p))$. Moreover for any weight w', $f_{\pi_0'}(1 \oplus 1, (\varepsilon, d, w')) = \uparrow$ thus, by Lemma 3, we also have $f_{\pi_0}(1 \oplus 1, (\varepsilon, d, w')) = \uparrow$.

Finally we conclude by Theorem 1. □

We cannot assume that the weight w is empty for the evaluation of f_π because, for some proofs, $f_\pi(A, (m, a, 1)) = \uparrow$ for any A, m and a (see the proof of $\vdash 1 \oplus 1$ in the introduction, for example).

The parallel machine gives a solution for this problem since it doesn't require any initial weight information. The weight may be built dynamically thus starting with 1 is sufficient.

Proposition 2. *If* π *is a proof of* $\vdash 1 \oplus 1$ *whose quasi-normal form is* π_0 *then* $\pi_0^\natural = \dfrac{\dfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1} \oplus_i$ *and either* $f_\pi^p(1 \oplus 1, (\varepsilon, g, 1)) \neq 0$ *and* $i = 1$ *or* $f_\pi^p(1 \oplus 1, (\varepsilon, d, 1)) \neq 0$ *and* $i = 2$.

4 Exponentials

We have now to generalize some of our definitions of Sect. 1 to deal with the following exponential rules. The interpretation is the one defined by Danos and Regnier [4], accommodated with the additives and extended to the $?w$-rule.

4.1 Sequent Calculus

$$\frac{\vdash \Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad \frac{\vdash \Gamma_1, A}{\vdash \Gamma, ?A}\,?d$$

$$\frac{\vdash \Gamma_1, B_1}{\vdash \Gamma, B, ?A}\,?w \qquad \frac{\vdash \Gamma_1, ?A_1, ?A_2}{\vdash \Gamma, ?A}\,?c \qquad \frac{\vdash \Gamma_1, ??A}{\vdash \Gamma, ?A}\,??$$

The formula B in the context of the $?w$-rule is used for the same purpose as in the $\bot$-rule. We use a *functorial* promotion and a *digging* rule (??-rule) instead of the usual promotion because it allows us to decompose precisely the GoI.

Definition 7 (Weight, *Definition 1 continued*). *A* copy address *c is a word built on the letters $\{g, d\}$.*

A basic weight *is now a pair of an elementary weight p (or its negation $\bar{p}$) and a copy address c, and is denoted by p_c ($\bar{p}_c$).*

In order to deal with the erasing of sub-proofs by the weakening cut elimination step, we will only consider proofs with no ? in conclusions. To prove the preservation of the interpretation by reduction, we can restrict cut elimination to the particular strategy reducing only exponential cuts with no context in the !-rule.

In the $?c$ cut elimination step, we obtain two copies π_1^1 and π_1^2 of the proof π_1 of $\vdash !A$. In π_1^1 (resp. π_1^2), we replace all the basic weights p_c by $p_{g.c}$ (resp. $p_{d.c}$).

4.2 Extending the Machine

Definition 8 (Exponential informations).

- Exponential signatures σ *and* exponential stacks s *are defined by:*

$$\sigma ::= \square \mid g.\sigma \mid d.\sigma \mid \ulcorner\sigma\urcorner.\sigma \mid [s]^{\Uparrow} \mid [s]^{\Downarrow}$$
$$s ::= \varepsilon \mid \sigma.s$$

  We will use the notation $[s]$ *to talk about both* $[s]^{\Uparrow}$ *and* $[s]^{\Downarrow}$.
- *The* copy address $\widetilde{\sigma}$ *of an exponential signature* σ *is defined by:* $\widetilde{\square} = \varepsilon$, $\widetilde{[s]} = \varepsilon$, $\widetilde{g.\sigma} = g.\widetilde{\sigma}$, $\widetilde{d.\sigma} = d.\widetilde{\sigma}$ *and* $\widetilde{\ulcorner\sigma'\urcorner.\sigma} = \widetilde{\sigma'}.\widetilde{\sigma}$.
- *The* copy address *of an exponential stack is:* $\widetilde{\varepsilon} = \varepsilon$ *and* $\widetilde{\sigma.s} = \widetilde{\sigma}.\widetilde{s}$.
- *We define the predicate* weak() *on signatures by:*
  - $weak(\square) = false$ *and* $weak([s]) = true$
  - $weak(g.\sigma) = weak(\sigma)$ *and* $weak(d.\sigma) = weak(\sigma)$
  - $weak(\ulcorner\sigma'\urcorner.\sigma) = weak(\sigma')$

The $weak()$ predicate tells if the leaf of the exponential branch described by σ is a $?w$-rule.

Definition 9 (Token, *Definition 4 continued*). *For the full case, a* token *is a tuple* (m, a, w, b, s) *where* b *and* s *are exponential stacks. Moreover the language of* m *is extended to* $\{g, d, \Uparrow, \Downarrow, |\}$ *and the language of* a *is extended to* $\{g, d, |\}$ *(*$\Uparrow$, $\Downarrow$ *and* $|$ *are only used for* $\bot$ *and* $?w$*).*

Definition 10 (Type of a token). *The* type *of a token* (m, a, w, b, s) *in a formula of a proof is the pair* $(|b| - d, |s| - n)$ *where* $|.|$ *is the length of a stack,* d *is the depth of the formula in the proof (i.e., the number of !-rules below it) and* n *is the number of exponential connectives in the scope of which the subformula described by* m *and* a *is (without looking at the right of any* $|$, $\Uparrow$ *or* $\Downarrow$ *symbol).*

In the transitions of the machine defined in Fig. 2, we replace everywhere p by $p_{\widetilde{b.c}}$ since we have to take into account the stack b and to look at the dependency with respect to $p_{\widetilde{b.c}}$, for example:

$$A \,\&\, B^{\uparrow}(m, g.a, w(p_{\widetilde{b.c}})) \to A^{\uparrow}(m, a, w(p_{\widetilde{b.c}}))$$

For the constants (Sect. 3.1), we have to refine the $\bot$-transitions (Fig. 4).

The new transitions of the token machine for exponential rules are described in Fig. 5. Some transitions are implicit to simplify the description: if no transition appears for a state $F^{\updownarrow}(m, a, w, b, s)$, it just corresponds to the transition $F^{\updownarrow}(m, a, w, b, s) \to \emptyset$.

$$
\begin{array}{lll}
 & \bot & \\
A^{\uparrow}(m,a,w,b,s) & \to \bot^{\downarrow}(\Uparrow.m,|.a,w,b,s) & \\
A_1^{\downarrow}(m,a,w,b,s) & \to \bot^{\downarrow}(\Downarrow.m,|.a,w,b,s) & \\
\bot^{\uparrow}(\Uparrow.m,|.a,w,b,s) & \to A_1^{\uparrow}(m,a,w,b,s) & \\
\bot^{\uparrow}(\Downarrow.m,|.a,w,b,s) & \to A^{\downarrow}(m,a,w,b,s) & \\
\bot^{\uparrow}(m,a,w,b,s) & \to \emptyset & m \neq \Uparrow.m', \Downarrow.m' \text{ or } a \neq |.a' \\
\Gamma^{\uparrow}(m,a,w,b,s) & \to \Gamma_1^{\uparrow}(m,a,w,b,s) & \\
\Gamma_1^{\downarrow}(m,a,w,b,s) & \to \Gamma^{\downarrow}(m,a,w,b,s) &
\end{array}
$$

Fig. 4. $\bot$-transitions (exponential case).

Lemma 4. *If the type of the starting token is (p,q) with $q \geq 0$, at any step of the execution the type of the token is (p,q') with $q' \geq 0$.*

Lemma 5. *$f_\pi(F,(m,a,w,b,s)) = (F',(m',a',w',b,s'))$, and if $\widetilde{b} = \widetilde{b_1}$ then $f_\pi(F,(m,a,w,b_1,s)) = (F',(m',a',w',b_1,s'))$.*

Theorem 1.b (Soundness *continued*). *The Theorem 1 is still true in $\mathsf{LL}^!$ for a proof without any ? in its conclusions and a token of type (p,q) with $p \geq 0$ and $q \geq 0$.*

Proof. We keep the same notations as in the proof of Theorem 1. We will look for each exponential cut at the sequence $s_1, s_2, \ldots$ (resp. $s'_1, s'_2, \ldots$) of the states $F^{\uparrow}(m,a,w,b,s)$ in the conclusions of the sub-proofs π_1 and π_2 during the computation of M_π (resp. $M_{\pi'}$) on the state associated to j. With the notations given below, we can remark that s_{2i} (resp. s_{2i+1}) will always be in the conclusions of π_1 (resp. π_2) and also for s'_{2i} and s'_{2i+1}.

We only prove the digging case, the others are left to the reader.

Digging cut. In this case, we cannot prove $\forall i, s_i = s'_i$ but only the weaker result $\forall i, s_i = F^{\uparrow}(m,a,w,b,s) \iff s'_i = F^{\uparrow}(m,a,w,b',s)$ with $\widetilde{b} = \widetilde{b'}$. Lemma 5 proves that it doesn't really matter.

$$
\dfrac{\dfrac{\dfrac{\pi_1}{\vdash A}}{\vdash\, !A}\,! \qquad \dfrac{\dfrac{\pi_2}{\vdash \Gamma_1, ??A^{\perp}}}{\vdash \Gamma_2, ?A^{\perp}}\,??}{\vdash \Gamma}\,cut
\quad\to\quad
\dfrac{\dfrac{\dfrac{\dfrac{\pi_1}{\vdash A}}{\vdash\, !A}\,!}{\vdash\, !!A}\,! \qquad \dfrac{\pi_2}{\vdash \Gamma_1, ??A^{\perp}}}{\vdash \Gamma}\,cut
$$

If s_{2i+1} exists then:

- either s_{2i+2} doesn't exist because f_{π_2} is not defined on s_{2i+1} or because $f_{\pi_2}(s_{2i+1}) \in \Gamma_1$,
- or $f_{\pi_2}(s_{2i+1}) = (??A^{\perp}, (m,a,w,b,s))$ with $s = \sigma.\sigma'.s'$ (Lemma 4) and $s_{2i+2} = A^{\uparrow}(m,a,w,(\ulcorner\sigma'\urcorner.\sigma).b,s')$.

Remark that the stack b of s_{2i} is always of the shape $(\ulcorner\sigma'\urcorner.\sigma).b'$. If s_{2i} exists then:

$$
\begin{array}{lll}
 & ! & \\
!A^{\uparrow}(m,a,w,b,\sigma.s) & \to A^{\uparrow}(m,a,w,\sigma.b,s) & \neg weak(\sigma) \\
!A^{\uparrow}(m,a,w,b,\sigma.s) & \to !A^{\downarrow}(m,a,w,b,\sigma.s) & weak(\sigma) \\
A^{\downarrow}(m,a,w,\sigma.b,s) & \to !A^{\downarrow}(m,a,w,b,\sigma.s) & \\
?\Gamma^{\uparrow}(m,a,w,b,\sigma.s) & \to \Gamma^{\uparrow}(m,a,w,\sigma.b,s) & \\
\Gamma^{\downarrow}(m,a,w,\sigma.b,s) & \to ?\Gamma^{\downarrow}(m,a,w,b,\sigma.s) &
\end{array}
$$

$$
\begin{array}{lll}
 & ?\,d & \\
?A^{\uparrow}(m,a,w,b,\Box.s) & \to A^{\uparrow}(m,a,w,b,s) & \\
?A^{\uparrow}(m,a,w,b,s) & \to \emptyset & s \neq \Box.s' \\
A^{\downarrow}(m,a,w,b,s) & \to ?A^{\downarrow}(m,a,w,b,\Box.s) & \\
\Gamma^{\uparrow}(m,a,w,b,s) & \to \Gamma_1^{\uparrow}(m,a,w,b,s) & \\
\Gamma_1^{\downarrow}(m,a,w,b,s) & \to \Gamma^{\downarrow}(m,a,w,b,s) &
\end{array}
$$

$$
\begin{array}{lll}
 & ?? & \\
?A^{\uparrow}(m,a,w,b,(\ulcorner\sigma'\urcorner.\sigma).s) & \to ??A^{\uparrow}(m,a,w,b,\sigma.\sigma'.s) & \\
?A^{\uparrow}(m,a,w,b,s) & \to \emptyset & s \neq (\ulcorner\sigma\urcorner.\sigma').s' \\
??A^{\downarrow}(m,a,w,b,\sigma.\sigma'.s) & \to ?A^{\downarrow}(m,a,w,b,(\ulcorner\sigma'\urcorner.\sigma).s) & \\
\Gamma^{\uparrow}(m,a,w,b,s) & \to \Gamma_1^{\uparrow}(m,a,w,b,s) & \\
\Gamma_1^{\downarrow}(m,a,w,b,s) & \to \Gamma^{\downarrow}(m,a,w,b,s) &
\end{array}
$$

$$
\begin{array}{lll}
 & ?\,c & \\
?A^{\uparrow}(m,a,w,b,(g.\sigma).s) & \to ?{A_1}^{\uparrow}(m,a,w,b,\sigma.s) & \\
?A^{\uparrow}(m,a,w,b,(d.\sigma).s) & \to ?{A_2}^{\uparrow}(m,a,w,b,\sigma.s) & \\
?A^{\uparrow}(m,a,w,b,s) & \to \emptyset & s \neq (g.\sigma).s', (d.\sigma).s' \\
?A_1{}^{\downarrow}(m,a,w,b,\sigma.s) & \to ?A^{\downarrow}(m,a,w,b,(g.\sigma).s) & \\
?A_2{}^{\downarrow}(m,a,w,b,\sigma.s) & \to ?A^{\downarrow}(m,a,w,b,(d.\sigma).s) & \\
\Gamma^{\uparrow}(m,a,w,b,s) & \to \Gamma_1^{\uparrow}(m,a,w,b,s) & \\
\Gamma_1^{\downarrow}(m,a,w,b,s) & \to \Gamma^{\downarrow}(m,a,w,b,s) &
\end{array}
$$

$$
\begin{array}{lll}
 & ?\,w & \\
B^{\uparrow}(m,a,w,b,s) & \to ?A^{\downarrow}(|.m,|.a,w,b,[s]^{\Uparrow}.[\varepsilon]^{\Uparrow^{k-1}}) & \\
B_1^{\downarrow}(m,a,w,b,s) & \to ?A^{\downarrow}(|.m,|.a,w,b,[s]^{\Downarrow}.[\varepsilon]^{\Downarrow^{k-1}}) & \\
?A^{\uparrow}(|.m,|.a,w,b,[s]^{\Uparrow}.s') & \to B_1^{\uparrow}(m,a,w,b,s) & \\
?A^{\uparrow}(|.m,|.a,w,b,[s]^{\Downarrow}.s') & \to B^{\downarrow}(m,a,w,b,s) & \\
?A^{\uparrow}(m,a,w,b,s) & \to \emptyset & s \neq [s'].s'' \text{ or } m \neq |.m' \text{ or } a \neq |.a' \\
\Gamma^{\uparrow}(m,a,w,b,s) & \to \Gamma_1^{\uparrow}(m,a,w,b,s) & \\
\Gamma_1^{\downarrow}(m,a,w,b,s) & \to \Gamma^{\downarrow}(m,a,w,b,s) &
\end{array}
$$

Fig. 5. Exponential group. (in the $?w$-transitions, k is the number of ? and ! in front of $?A$ and $[\varepsilon]^{\Uparrow^{k-1}}$ is used to preserve a correct type of the token)

– either s_{2i+1} doesn't exist because f_{π_1} is not defined on s_{2i},
– or $f_{\pi_1}(s_{2i}) = (A, (m, a, w, b, s))$ and, by Lemma 5, $b = (\ulcorner\sigma'\urcorner.\sigma).b'$ thus $s_{2i+1} = A^{\perp\uparrow}(m, a, w, b', \sigma.\sigma'.s)$.

We also have $s'_{2i+1} = s_{2i+1}$ and if $s_{2i} = A^{\uparrow}(m, a, w, (\ulcorner\sigma'\urcorner.\sigma).b, s)$ then $s'_{2i} = A^{\uparrow}(m, a, w, \sigma'.\sigma.b, s)$ by Lemma 5 with $\widetilde{(\ulcorner\sigma'\urcorner.\sigma)}.b = \widetilde{\sigma'.\sigma.b}$. □

To conclude, we have to note that the Theorem 3, about computation for booleans, is still true in the full case!

Acknowledgements. Thanks to Laurent Regnier for his support and to the referees for their comments about presentation.

References

[1] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF (extended abstract). In M. Hagiya and J. C. Mitchell, editors, *Theoretical Aspects of Computer Software*, volume 789 of *Lecture Notes in Computer Science*, pages 1–15. Springer, April 1994.

[2] Andrea Asperti, Cecilia Giovannetti, and Andrea Naletto. The bologna optimal higher-order machine. *Journal of Functional Programming*, 6(6):763–810, November 1996.

[3] Vincent Danos, Hugo Herbelin, and Laurent Regnier. Games semantics and abstract machines. In *Proceedings of Logic In Computer Science*, New Brunswick, 1996. IEEE Computer Society Press.

[4] Vincent Danos and Laurent Regnier. Reversible, irreversible and optimal λ-machines. In J.-Y. Girard, M. Okada, and A. Scedrov, editors, *Proceedings Linear Logic Tokyo Meeting*, volume 3 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1996.

[5] Jean-Yves Girard. Geometry of interaction I: an interpretation of system F. In Ferro, Bonotto, Valentini, and Zanardo, editors, *Logic Colloquium '88*. North-Holland, 1988.

[6] Jean-Yves Girard. Geometry of interaction III: accommodating the additives. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*, pages 329–389. Cambridge University Press, 1995.

[7] Jean-Yves Girard. Proof-nets: the parallel syntax for proof-theory. In Ursini and Agliano, editors, *Logic and Algebra*, New York, 1996. Marcel Dekker.

[8] Georges Gonthier, Martin Abadi, and Jean-Jacques Lévy. The geometry of optimal lambda reduction. In *Proceedings of Principles of Programming Languages*, pages 15–26. ACM Press, 1992.

[9] Olivier Laurent. Polarized proof-nets and λμ-calculus. To appear in *Theoretical Computer Science*, 2001.

[10] Ian Mackie. The geometry of interaction machine. In *Proceedings of Principles of Programming Languages*, pages 198–208. ACM Press, January 1995.

[11] Marco Pedicini and Francesco Quaglia. A parallel implementation for optimal lambda-calculus reduction. In *Proceedings of Principles and Practice of Declarative Programming*, 2000.