A Semantic Measure of the Execution Time in Linear Logic

D. de Carvalho [a,1], M. Pagani [b,2], L. Tortora de Falco [c]

[a] *INRIA Lorraine – LORIA*

[b] *Laboratoire PPS & Université Paris 7*

[c] *Dipartimento di Filosofia – Università Roma Tre*

Abstract

We give a semantic account of the execution time (i.e. the number of cut elimination steps leading to the normal form) of an untyped $MELL$ net. We first prove that: 1) a net is head-normalizable (i.e. normalizable at depth 0) if and only if its interpretation in the multiset based relational semantics is not empty and 2) a net is normalizable if and only if its *exhaustive* interpretation (a suitable restriction of its interpretation) is not empty. We then give a semantic measure of execution time: we prove that we can compute the number of cut elimination steps leading to a cut free normal form of the net obtained by connecting two cut free nets by means of a cut link, from the interpretations of the two cut free nets. These results are inspired by similar ones obtained by the first author for the untyped lambda-calculus.

Key words: Linear Logic, Denotational Semantics, Computational complexity

---

1 Introduction

Right from the start, Linear Logic (LL, [9]) appeared as a potential logical tool to study computational complexity. The logical status given by the exponentials (the new connectives of LL) to the operations of erasing and copying (corresponding to the structural rules of intuitionistic and classical logic) shed a new light on the duplication process responsible of the "explosion" of the size (and time) during the cut elimination procedure. This is witnessed by the

---

Email addresses: `Daniel.decarvalho@loria.fr` (D. de Carvalho), `pagani@uniroma3.it` (M. Pagani), `tortora@uniroma3.it` (L. Tortora de Falco).

[1] Partially supported by project NO-CoST (ANR, JC05 43380)

[2] Supported by a postdoc fellowship Dip.to Filosofia, Università Roma Tre

contribution given by LL to the wide research area called Implicit Computational Complexity: a true breakthrough with this respect is Girard's Light Linear Logic (LLL, [10]). A careful handling of LL's exponentials allows the author to keep enough control on the duplication process, and to prove that a function f is representable in LLL if and only if f is polytime.
One of the main questions arisen from [10] is the quest of a denotational semantics suitable for light systems (a semantics of proofs in logical terms, or more generally a model). Among the main attempts in this direction we can quote on the one hand [12,1], where the structures (games, coherent spaces) associated with logical formulas[3] are modified so that the principles valid in LL but not in the chosen light system do not hold in the semantics, and on the other hand [11] which deals with a property of the elements of the structures (the interpretations of proofs) characterizing those elements which *can* interpret proofs with bounded complexity.

A different approach to the semantics of bounded time complexity is possible: the basic idea is to measure by semantic means the execution of any program, regardless to its computational complexity. The aim is to compare different computational behaviors and to learn afterwards something on the very nature of bounded time complexity. Following this approach, in [7,5] one of the authors of the present paper could compute the execution time of an untyped λ-term from its interpretation in the Kleisli category of the comonad associated with the finite multisets functor on the category **Rel** of sets and relations. Such an interpretation is the same as the interpretation of the net translating the λ-term in the multiset based relational model of linear logic. The execution time is measured here in terms of elementary steps of the so-called Krivine's machine. Also, [7,5] give a precise relation between an intersection types system introduced by [2] and experiments in the multiset based relational model. Experiments are a tool introduced by Girard in [9] allowing to compute the interpretation of proofs pointwise. An experiment corresponds to a type derivation and the result of an experiment corresponds to a type.[4]

We apply here this approach to Multiplicative and Exponential Linear Logic (MELL), and we show how it is possible to compute the number of steps of cut elimination by semantic means (notice that our measure being the number of cut elimination steps, here is a first difference with [7,5] where Krivine's machine was used to measure execution time). Linear Logic offers a very sharp way to study Gentzen's cut elimination by representing proofs

---

[3] The basic pattern of denotational semantics is to associate with every formula an object of some category and with every proof of the formula a morphism of this category called the interpretation of the proof.

[4] The intersection types system considered in [7,5] lacks idempotency and this fact was crucial in that work. In the present paper, this corresponds to the fact that we use multisets for interpreting exponentials and not sets as in the set based coherent semantics. The use of multisets is essential in our work too.

as graphs with boxes, called *proof-nets* [9]. The peculiarity of proof-nets is to reduce the number of commutative cut elimination steps, which instead abounded in sequent calculi. If π_2 is a proof-net obtained by applying some steps of cut elimination to π_1, the main property of any model is that the interpretation $[\![\pi_1]\!]$ of π_1 is the same as the interpretation $[\![\pi_2]\!]$ of π_2, so that from $[\![\pi_1]\!]$ it is clearly impossible to determine the number of steps leading from π_1 to π_2. Nevertheless, if π_1 and π_2 are two cut free proof-nets connected by means of a cut link, we can wonder:

(1) is it the case that the thus obtained net can be reduced to a cut free one?
(2) if the answer to the previous question is positive, what is the number of cut reduction steps leading from the net with cut to a cut free one?

The main point of the paper is to show that it is possible to answer both these questions by only referring to $[\![\pi_1]\!]$ and $[\![\pi_2]\!]$ [5].

The first question makes sense only in an untyped framework (in the typed case, cut elimination is strongly normalizing, see [9]), and indeed Subsect. 2.1 is devoted to define an untyped version of Girard's proof-nets, based on previous works, mainly [3,15,11,14]. Terui [16] also introduced a calculus corresponding to an untyped and intuitionistic version of proof-nets of Light Affine Logic and [6] addressed the problem of characterizing the (head-)normalizable nets in this restricted setting. We shift here from the intuitionistic to the classical framework. Let us mention here that to improve readability we chose to state and prove our results for proof-nets (i.e. logically correct proof-structures), but correctness (in our framework Def. 3) is rarely used (see also Rk. 37). The cut elimination procedure we define is similar to λ-calculus β-reduction, in the sense that the exponential step (the step (!/?) of Def. 6) is more similar to a step of β-reduction than it usually is. This is essential to prove our results (see the discussion on Fig. 5). We consider in the paper two reduction strategies: head reduction and stratified reduction. The first one consists in reducing the cuts at depth 0 and stop. The second one consists in reducing a cut only when there exists no cut with (strictly) smaller depth. These reduction strategies extend the head (resp. leftmost) reduction of λ-calculus.

Section 2 is devoted to define our version of proof-net (Subsection 2.1) and the model allowing to measure the number of cut elimination steps (Subsection 2.2). In Section 3, we show how experiments provide a counter for head and stratified reduction steps (Lemmas 17, 20). In Section 4 we answer question (1), and in Section 5 we answer question (2).

Let us conclude with a little remark. In [17], the question of injectivity for the relational and coherent semantics of LL is addressed: is it the case that for π

---

[5] The questions (and the answers) are more general than it seems: every proof-net with cuts is the reduct of some proof-net obtained by cutting two cut free proof-nets.

and π' cut free, from $[\![\pi]\!] = [\![\pi']\!]$ one can deduce $\pi = \pi'$? It is conjectured that relational semantics is injective for MELL, and there is still no answer to this question. Given π_1 and π_2, we don't know how to compute the normal form of the net obtained by connecting π_1 and π_2 by means of a cut link from $[\![\pi_1]\!]$ and $[\![\pi_2]\!]$. Indeed, the present paper shows that from $[\![\pi_1]\!]$ and $[\![\pi_2]\!]$ we can at least compute the number of cut elimination steps leading to the normal form.

2 Preliminaries

We introduce the syntax and the model for which we prove our results: the untyped nets and their interpretation in the category **Rel** of sets and relations.

2.1 Untyped nets

After their introduction by Girard in [9], proof-nets have been extensively studied and used as a proof-theoretical tool for several purposes. All this work led to many improvements of the original notion introduced by Girard.
We use here an untyped version of Girard's proof-nets. Danos and Regnier [3,15] introduced and studied "pure proof-nets" that is the exact notion of proof-net corresponding to pure λ-calculus. There has been no real need for a different notion of untyped proof-net until Girard's work on Light Linear Logic [10]: Terui [16] introduces a "light" untyped λ-calculus enjoying strong normalization in polynomial time and encoding all polytime functions. This calculus clearly corresponds to an untyped and intuitionistic version of proof-nets. In the same spirit, an untyped notion of proof-net (called *net*) is introduced in [11] in order to encode polytime computations: the novelty here is the shift from the intuitionistic to the classical framework (see also [14]). This yields cuts which cannot be reduced and called *clashes* (see Def. 5 and Fig. 2).
By following [15,4], we choose here a version of nets where ?-links have $n \geq 0$ premises (these links are often represented by a tree of contractions and weakenings). We also have a $\flat$-node which is our way to represent dereliction. These choices allow a strict correspondence between the number of steps of the cut elimination of a net and its interpretation in **Rel** (see Theorem 36). We will end the subsection with a brief discussion on these choices.

Definition 1 (Ground-structure) *A* ground-structure*, or* g-structure *for short, is a finite (possibly empty) labelled directed acyclic graph whose nodes (also called links) are defined together with an arity and a coarity, i.e. a given number of incident edges called the* premises *of the node and a given number of emergent edges called the* conclusions *of the node. The valid nodes are:*

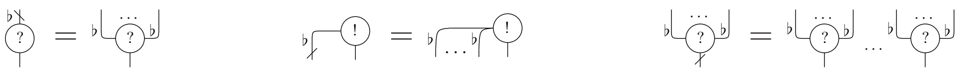
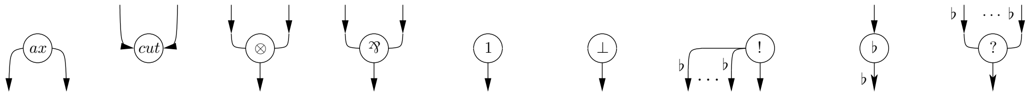


Fig. 1. Some conventions to picture an arbitrary number of nodes/edges



An edge can have or not a $\flat$ *label: an edge with no label (resp. with a* $\flat$ *label) is called* logical *(resp.* structural*). The* $\flat$*-nodes have a logical premise and a structural conclusion, the* $?$*-nodes have* $k \geq 0$ *structural premises and one logical conclusion, the* $!$*-nodes have no premise, exactly one logical conclusion, also called* main *conclusion of the node, and* $k \geq 0$ *structural conclusions, called* auxiliary *conclusions of the node. Premises and conclusions of the nodes* ax*,* cut*,* $\otimes$*,* $\parr$*,* 1*,* $\bot$ *are logical edges. We allow edges with a source but no target, they are called* conclusions *of the g-structure; we consider that a g-structure is given with an order* $(c_1, \ldots, c_n)$ *of its conclusions.*
We denote by $!(\alpha)$ *the set of* $!$*-links of a g-structure* α*.*

When drawing a g-structure we order its conclusions from left to right. Also we represent edges oriented top-down so that we speak of moving upwardly or downwardly in the graph, and of nodes or edges "above" or "under" a given node/edge. In the sequel we will not write explicitly the orientation of the edges. In order to give more concise pictures, when not misleading, we may represent an arbitrary number of $\flat$-edges (possibly zero) as a $\flat$-edge with a diagonal stroke drawn across (see Fig 1). In the same spirit, a ?-link with a diagonal stroke drawn across its conclusion represents an arbitrary number of ?-links, possibly zero (see Fig 1). For an example of this notation see Fig. 3. Given any set X, we denote by $\mathbf{X}$ the set of finite sequences of elements of X, and by $\mathbf{x}$ a generic element of X. For example, a sequence $(c_1, \ldots, c_n)$ of conclusions of a g-structure α may be denoted simply by $\mathbf{c}$.

Definition 2 (Untyped $\flat$-structure) *An* untyped $\flat$-structure*, or simply* $\flat$-structure*,* π of depth 0 *is a g-structure without* $!$*-nodes; in this case, we set* $\mathrm{ground}(\pi) = \pi$*. An* untyped $\flat$-structure π of depth $d+1$ *is a g-structure* α*, denoted by* $\mathrm{ground}(\pi)$*, with a function that assigns to every* $!$*-link* o *of* α *with* $n_o + 1$ *conclusions a* $\flat$*-structure* π^o *of depth at most* d*, called the* box of o*, with* n_o *structural conclusions and exactly one logical conclusion, and a bijection from the set of the* n_o *structural conclusions of the link* o *to the set of the* n_o *structural conclusions of the* $\flat$*-structure* π^o*. Moreover* α *has at least one* $!$*-link with a box of depth* d*.*
The depth of a link l in π *is the number of boxes of* π *containing* l*.*

In the following definition we introduce the untyped nets by means of switching acyclicity. This is a standard notion of correctness which characterizes the structures which are sequentializable in a calculus extended with the mix rule.

Definition 3 (Untyped nets) *A* switching *of a g-structure α is an (undirected) subgraph of α obtained by forgetting the orientation of α's edges, by deleting one of the two premises of each ⅋-node, and for every ?-node l with $n \geq 1$ premises, by erasing all but one premises of l.*
An untyped $\flat$-net, $\flat$-net *for short, is a $\flat$-structure π s.t. every switching of every g-structure of π is an acyclic graph. An* untyped net, net *for short, is a $\flat$-net with no structural conclusion.*

Notice that with every structural edge b of a net is associated exactly one $\flat$-node and one ?-node: we will refer to these nodes as *the $\flat$-node/?-node associated with b*. Observe that the $\flat$-node and the ?-node associated with a given edge might have a different depth.

Concerning the presence of empty nets, notice that the empty net does exist and it has no conclusion. Its presence is required by the cut elimination procedure (Def. 6): the elimination of a cut between a 1-link and a $\perp$-link yields the empty graph, and similarly for a cut between a !-link with no auxiliary conclusion and a 0-ary ?-link. On the other hand, notice also that with a !-link o of a net, it is *never* possible to associate the empty net: o has at least one conclusion and this has also to be the case for the net associated with o.

Definition 4 (Nets size) *The* size $s(\alpha)$ of a g-structure α *is the number of the logical edges of α. The* size $s(\pi)$ of a $\flat$-structure π *is defined by induction on the depth of π, as follows: $s(\pi) = s(\text{ground}(\pi)) + \sum_{o \in !(\text{ground}(\pi))} s(\pi^o)$.*

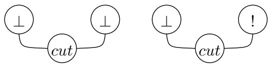


Fig. 2. Two clashes

Since we are in an untyped framework, nets may contain *"pathological"* cuts (see examples in Fig. 2) which are not reducible. These cuts are called *clashes* and their presence is in contrast with what happens in λ-calculus, where the simpler grammar of terms avoid clashes also in an untyped framework.

Definition 5 (Clash) *The two edges premises of a cut-link are* dual *when:*

- *they are conclusions of resp. a $\otimes$-node and of a ⅋-node, or*
- *they are conclusions of resp. a 1-node and of a $\perp$-node, or*
- *they are conclusions of resp. a !-node and of a ?-node.*

A cut-link is a clash, *when the premises of the cut-node are not dual edges and none of the two is the conclusion of an ax-link.*

Definition 6 (Cut elimination) *The cut elimination is defined as in [4]. To eliminate a cut t in a net π means in general to transform π in a net*[6] *$t(\pi)$ by substituting a specific subgraph β of π with a subgraph β' having the*

---

[6] The fact that $t(\pi)$ is indeed a net should be checked, see [15].

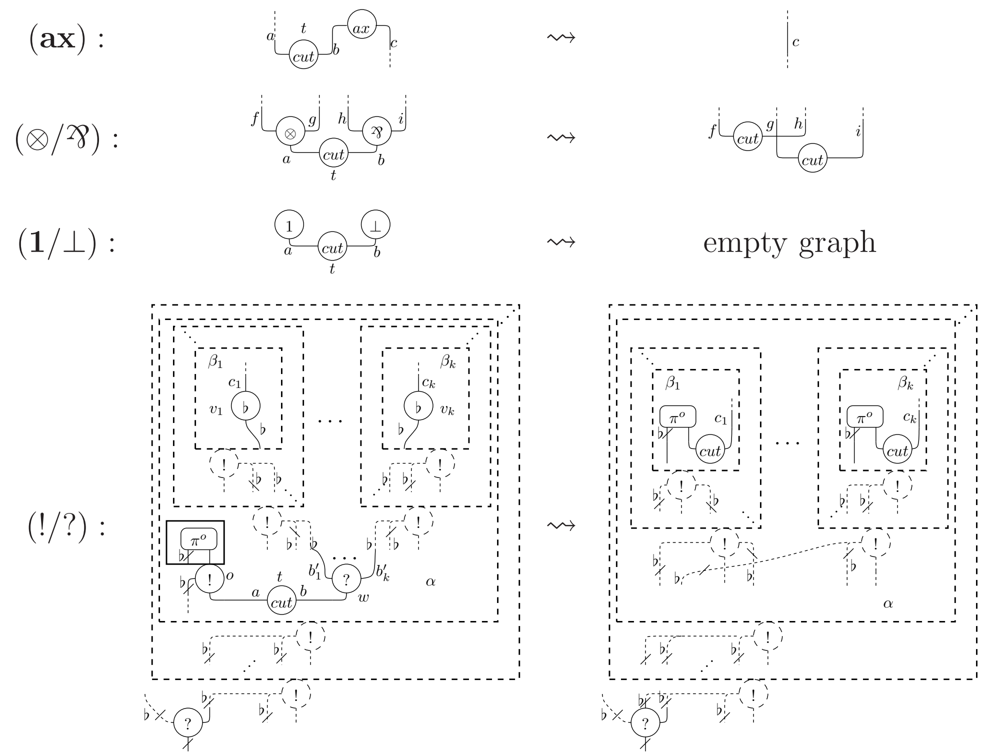


Fig. 3. Cut elimination for nets. In the (!/?) case what happens is that the !-link o dispatches k copies of π^o ($k \geq 0$ being the arity of the ?-node w premise of the cut) inside the !-boxes (if any) containing the ♭-nodes associated with the premises of w; notice also that the reduction duplicates k times the premises of ?-nodes which are associated with the auxiliary conclusions of o.

---

same pending edges (i.e. edges with no target or no source) as β. The subgraphs β and β' depend on the cut t and are described in Fig. 3.

We will also refer to $t(\pi)$ as a one step reduct of π, and to the transformations associated with the different types of cut link as the reduction steps. *We write $\pi \rightsquigarrow \pi'$, when π' is the result of one reduction step.*

A reduction step is said to be a head reduction step *when the reduced cut t has depth 0 in π: we write $\pi \rightsquigarrow_h \pi'$, when π' is the result of one head reduction step. A reduction step $\pi \rightsquigarrow t(\pi)$ is said to be a* stratified reduction step *when for every cut-link (including clashes) t' of π we have $depth(t) \leq depth(t')$: we write $\pi \rightsquigarrow_s \pi'$ when π' is the result of one stratified reduction step.*

We denote by $\rightsquigarrow^*$ (resp. $\rightsquigarrow_h^*$ and $\rightsquigarrow_s^*$) the reflexive and transitive closure of $\rightsquigarrow$ (resp. $\rightsquigarrow_h$ and $\rightsquigarrow_s$). A net π is head-normalizable *(resp.* normalizable*) if there exists a head-cut free (resp. cut free) net π_0 such that $\pi \rightsquigarrow^* \pi_0$.*

A reduction sequence *R from π to π' is a sequence of nets $(\pi_1, \ldots, \pi_n)$, s.t. $R = \pi \rightsquigarrow \pi_1 \rightsquigarrow \ldots \rightsquigarrow \pi_n = \pi'$. A reduction sequence R is an* head reduction *(resp. a* stratified reduction*) when every step of R is an head (resp. a stratified) reduction step.*

Notice that the cut elimination cannot be applied to clashes, and this means that there are nets to which no cut elimination step can be applied, even if they are not cut free (consider for example the nets of Fig. 2).
Notice also that cut elimination is defined on nets and not on general ♭-nets. This is because we want $\rightsquigarrow$ to leave unchanged the number of conclusions of a net: this is true only for the logical conclusions, the structural ones may be changed by the (!/?)-steps. In the sequel, however, we need to speak of *the cut elimination of a box* π^o (which is a ♭-net) associated with a !-link o: in that case we mean the cut elimination of the net obtained by adding to π^o the ?-links of π associated with the structural conclusions of π.

Definition 7 (Ancestor, residue) *Let* $\pi \rightsquigarrow \pi'$*. When an edge* d *(resp. a node* l*) of* π' *comes from a (unique) edge* $\overleftarrow{d}$ *(resp. node* $\overleftarrow{l}$*) of* π*, we say that* $\overleftarrow{d}$ *(resp.* $\overleftarrow{l}$*) is the* ancestor *of* d *(resp.* l*) in* π *and that* d *(resp.* l*) is a* residue *of* $\overleftarrow{d}$ *(resp.* $\overleftarrow{l}$*) in* π'*. If this is not the case, then* d *(resp.* l*) has no ancestor in* π*, and we say it is a* created *edge (resp. node). We indicate, for every type of cut elimination step of Fig. 3, which edges (resp. links) are created in* π' *(meaning that the other edges (resp. nodes) of* π' *are residues of some* π*'s node). We use the notations of Fig. 3:*

- (ax)*: there are no created edges, nor created nodes in* π'*. Remark that* a, b *are erased in* π'*, so that we consider* c *in* π' *as the residue of* c *in* π*;*
- $(\otimes/⅋)$*: there are no created edges, while the two new cut-links between the two left (resp. right) premises of the* ⅋*- and* $\otimes$*-links are created nodes;*
- $(1/\bot)$*: there are no created edges, nor created nodes in* π'*;*
- (!/?)*: every auxiliary conclusion added to the* !*-links containing one copy of* π^o *is a created edge; every cut link between (a copy of)* π^o*'s main conclusion and* c_i *is a created node.*[7]

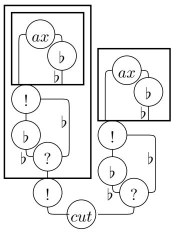


Fig. 4. Example of a non-normalizable net.

Examples. It is well-known that there are non-normalizable *untyped* nets. A famous example is the net corresponding to the untyped λ-term $(\lambda x.xx)(\lambda x.xx)$ (see [3], [15]). We give in Fig. 4 a slight variant (which is not a λ-term), due to Mitsu Okada. The reader can check that this net reduces to itself by one (!/?) step and one (ax) step.

Let us briefly discuss the complexity of our cut elimination, in particular of the (!/?) step. Consider the net π in Fig. 5. Different head reductions start from π, depending on which cut t_i (for $i \leq n$) we choose to reduce. But every such

---

[7] Notice that every !-link of π' which contains a copy of π^o is considered a residue of the corresponding !-link of π, even though it has different auxiliary conclusions. Notice also that the edges/nodes in each copy of π^o are considered residues of the corresponding edges/nodes in π^o.

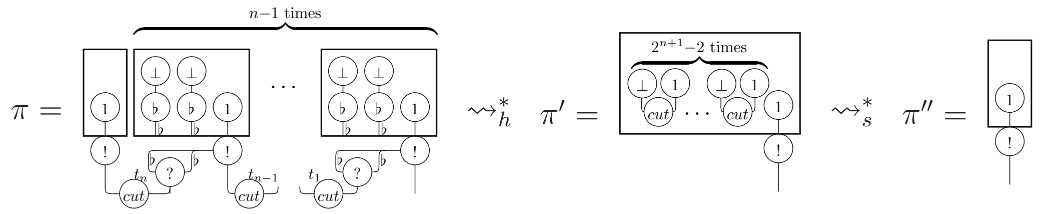


Fig. 5. Example of the "cost" of cut elimination ($n \geq 1$).

reduction eventually reaches the head-cut free net π'. Besides, all head reductions ending in π' have the same length: they consist in n steps, of type (!/?). Indeed it is a general property that two head (resp. stratified) reductions of a net leading to a head-cut (resp. cut) free net always have the same length, as proven in Cor. 29.

This property is peculiar of the syntax we have chosen, which gathers in a unique step (!/?) all the exponential steps of **MELL** (see [4]). In the original syntax of [9] (see also [14]), the (!/?) step splits in (!/?d), (!/?w), (!/?c) and (!/!). A crucial motivation for our choice is that in the syntax of [9], different head reductions leading to the same head-cut free net may have different lengths. For example, the net π of Fig. 5 can be transformed in π' by a different number of exponential steps of [9]. One of the shortest reduction is obtained by reducing the cuts $t_1, \ldots, t_n$ in a decreasing order (w.r.t. the index): reduce t_n and the two created (!/!) cuts, then t_{n-1} and the two created (!/!) cuts, then t_{n-2} and so on. This reduction leads to π' after $3n$ steps at depth 0 (n of type (!/?c) and $2n$ of type (!/!))[8]. On the other hand, by reducing $t_1, \ldots, t_n$ in an increasing order, one gets one of the longest head reductions: reduce t_1 and the two created (!/!) cuts, then focus on t_2 and notice that the t_1 reduction has created two new ?c nodes above t_2, so that to eliminate t_2 and the cuts created by t_2 at depth 0 we need to perform 3 (!/?c) steps and 4 (!/!) steps, then for t_3 we need 7 (!/?c) steps and 8 (!/!) steps, and so on. Eventually, it turns out that the length of this reduction sequence is $\sum_{i=1}^{n}(2^{i+1} - 1) = 2^{n+2} - 4 - n$, thus exponential w.r.t. the size of π.

This example shows that there is no hope to implement this reduction on a Turing Machine (TM) in an efficient (polynomial) way. The point is that the reduction $\pi \rightsquigarrow_h^* \pi'$ has length n in our syntax, while the size of π' is exponential in n and in the size of π. This means that any TM needs a number of steps exponential w.r.t. the length n of $\pi \rightsquigarrow_h^* \pi'$ and the size of π, just to write the result π' of the reduction. This drawback is due exactly to the (!/?) step, which may increase exponentially the size of a net.

However, while (due to the previous example) head-normalization cannot be implemented in an efficient way, the situation might very well be different for the stratified reduction. In our example, in order to reach the cut free net π''

---

[8] In order to reach π' one actually needs also $2n$ (!/?d) steps with depth 1.

from the head-cut free net π' one needs $2^{n+1} - 2$ more stratified steps, which means that (at least in this case) nothing prevents the number of stratified steps leading from π to the cut free π'' to be closely related to the number of steps required by a TM to compute π''.

2.2 Denotational semantics

We define here the interpretation allowing to measure execution time. Our aim is to use the multiset based relational model, but notice that we want to interpret *untyped* nets. In λ-calculus, the shift from typed to untyped semantics essentially relies on the choice of a suitable object D which is reflexive, that is s.t. $D \to D$ is a retract of D (*via* some morphisms). In the **MELL** framework we have more constructions than the intuitionistic arrow, then it is not enough for the object D we look for to enjoy the λ-calculus notion of reflexivity (it must satisfy more properties). Indeed we define an object D (Definition 8) in the category **Rel** of sets and relations in such a way that not only $D^{\perp}$, $D \otimes D$, $D \parr D$, $!D$ and $?D$ are retracts of D, but also that each of these constructs interacts well with the others (*via* some morphisms), thus allowing an interpretation of untyped nets invariant under cut-elimination (Theorem 11). Let us fix a set $\mathcal{A}$ of “atoms”, such that $\mathcal{A}$ does not contain any pair nor any multiset. We also require that $* \notin \mathcal{A}$: these conditions on $\mathcal{A}$ ensure that following Definition 8 we obtain an object D that satisfies the equation $D = \mathcal{A} \oplus \mathcal{A}^{\perp} \oplus 1 \oplus \perp \oplus (D \otimes D) \oplus (D \parr D) \oplus !D \oplus ?D$, where the constructs have the usual interpretations: $\mathcal{A}^{\perp} = \mathcal{A}$, $\otimes$ and $\parr$ are the cartesian product of sets, 1 and $\perp$ are the singleton $\{*\}$, ! and ? are the finite multisets functor, and $\oplus$ is the disjoint union[9].

Definition 8 *We set:*

$$\begin{aligned} D_0 &:= \{+,-\} \times (\mathcal{A} \cup \{*\}) \\ D_{n+1} &:= D_0 \cup (\{+,-\} \times ((D_n \times D_n) \cup \mathcal{M}_{fin}(D_n))) \\ D &:= \bigcup_{n \in \mathbb{N}} D_n \end{aligned}$$

We call the depth of an element $x \in D$ *the least number* $n \in \mathbb{N}$ *s.t.* $x \in D_n$. *We denote the set of finite sequences of* D*'s elements by* $\mathbf{D}$*; a generic element of* $\mathbf{D}$ *is denoted in boldface:* $\mathbf{y} \in \mathbf{D}$.

By definition of D, we have elements of type $(p,(x,y))$ for $p \in \{+,-\}$ and $x, y \in D$: actually, to improve readability we omit in the sequel some parenthesis and we simply write (p,x,y).

---

[9] The previously mentioned conditions guarantee that the following definition of D gives rise indeed to a *disjoint* union.

Definition 9 *Let* $+^\perp = -$ *and* $-^\perp = +$. *We define* $x^\perp$ *for any* $x \in D$, *by induction on* $\mathrm{depth}(x)$*:*

- *for* $a \in \mathcal{A} \cup \{*\}$, $(p, a)^\perp = (p^\perp, a)$*;*
- *else,* $(p, x, y)^\perp = (p^\perp, x^\perp, y^\perp)$, *and* $(p, [x_1, \ldots, x_n])^\perp = (p^\perp, [x_1^\perp, \ldots, x_n^\perp])$.

A key feature is that, for every $x \in D$, one has $x \neq x^\perp$, a property used in the proof of Theorem 21 and also in Definition 18 of *exhaustive* element.

Now, we show how to compute the interpretation of an untyped net directly, without passing through a sequent calculus. This is done by adapting the notion of experiment to our untyped framework. For a net π with n conclusions, we define the *interpretation of* π, denoted by $[\![\pi]\!]$, as a subset of $\parr_{i=1}^n D$, that can be seen as a morphism from 1 to $\parr_{i=1}^n D$. We compute $[\![\pi]\!]$ by means of the *experiments of* π, a notion introduced by Girard in [9] and central in this paper. We define, by induction on the depth of π, an experiment e of π:

Definition 10 (Experiment) [10] *An* experiment e of a $\flat$-net π, *denoted by* $e : \pi$, *is a function which associates with every* $!$*-link* o *of* $\mathrm{ground}(\pi)$ *a multiset* $[e_1^o, ..., e_k^o]$ $(k \geq 0)$ *of experiments of* π^o, *and with every edge* a *of* $\mathrm{ground}(\pi)$ *an element of* D, *s.t. if* a, b, c *are edges of* $\mathrm{ground}(\pi)$ *the following conditions hold (see Fig. 6):*

- *if* a, b *are the conclusions (resp. the premises) of an ax-link (resp. cut-link), then* $e(a) = e(b)^\perp$*;*
- *if* c *is the conclusion of a* 1*-link (resp.* $\perp$*-link), then* $e(c) = (+, *)$ *(resp.* $e(c) = (-, *)$*);*
- *if* c *is the conclusion of a* $\otimes$*-link (resp.* $\parr$*-link) with premises* a, b, *then* $e(c) = (+, e(a), e(b))$ *(resp.* $e(c) = (-, e(a), e(b))$*);*
- *if* c *is the conclusion of a* $\flat$*-link with premise* a, *then* $e(c) = (-, [e(a)])$*;*
- *if* c *is the conclusion of a* $?$*-link with premises* $a_1, \ldots, a_n$, *and for every* $i \leq n$, $e(a_i) = (-, \mu_i)$, *where* μ_i *is a finite multiset of elements of* D, *then* $e(c) = \left(-, \sum_{i \leq n} \mu_i\right)$*; in particular if* c *has no premises, then* $e(c) = (-, [\,])$*;*
- *if* c *is a conclusion of a* $!$*-link* o *of* $\mathrm{ground}(\pi)$, *let* π^o *be the box of* o *and* $e(o) = [e_1^o, \ldots, e_n^o]$. *If* c *is the main conclusion of* o, *let* c^o *be the main conclusion of* π^o, *then* $e(c) = (+, [e_1^o(c^o), \ldots, e_n^o(c^o)])$, *if* c *is an auxiliary conclusion of* o, *let* c^o *be the auxiliary conclusion of* π^o *associated with* c, *and for every* $i \leq n$, *let* $e_i^o(c^o) = (-, \mu_i)$, *then* $e(c) = \left(-, \sum_{i \leq n} \mu_i\right)$.

If $c_1, \ldots, c_n$ *are the conclusions of* π, *then the* result of e, *denoted by* $|e|$, *is the element*[11] $(e(c_1), \ldots, e(c_n))$ *of* $\parr_{i=1}^n D$. *The* interpretation of π *is the set of the results of its experiments:*

---

[10] Remark that the following definition is slightly different from that used in [17], namely e is defined only on the edges of $\mathrm{ground}(\pi)$.

[11] Recall that a g-structure, hence a $\flat$-net, is given together with an order on its conclusions, so the sequence $(e(c_1), \ldots, e(c_n))$ is univocally determined by e and π.

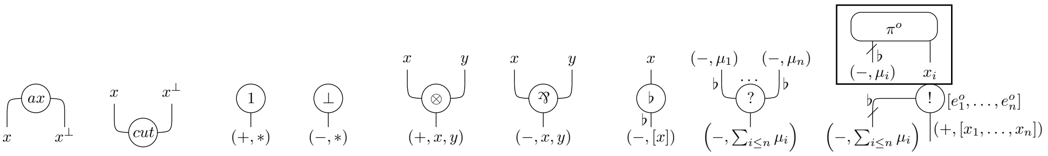


Fig. 6. Experiments of $\flat$-nets

$$[\![\pi]\!] := \{(e(c_1),\ldots,e(c_n))\ ;\ e\ \textit{experiment of}\ \pi\}\quad .$$

If $\mathbf{y}=(e(c_1),\ldots,e(c_n))$ *is the result of an experiment* $e:\pi$, *we denote by* $\mathbf{y}_{c_i}$ *the element* $e(c_i)$, *for every* $i\leq n$. *Generally, if* $\mathbf{d}=(c_{i_1},\ldots,c_{i_k})$ *is a sequence of conclusions of* π, *we note by* $\mathbf{y}_\mathbf{d}$ *the element* $(e(c_{i_1}),\ldots,e(c_{i_k}))$ *of* $\mathbf{D}$.

In general an experiment e of a $\flat$-net π is uniquely determined by its values on the axiom conclusions and on the !-links of ground(π); if moreover π is head-cut free, then any choice of values for the axiom conclusions and for the !-links of ground(π) defines an experiment.

In Fig. 7 we give some examples of experiments: consider the topmost net π of Fig. 7 and its experiment e, one has $|e|=\Big((-,2[x]+2[y])\,,\big(-,2[x^\perp]+2[y^\perp]\big)\Big)$. The interpretation of π is:

$$[\![\pi]\!]=\Big\{\Big((-,[x_1,\ldots,x_4])\,,\big(-,[x_1^\perp,\ldots,x_4^\perp]\big)\Big)\ ;x_i\in D,\text{for } i\leq 4\Big\}\quad .$$

The reader can check that every cut reduct of π (for example the nets π_1,π_2,π_8 of Fig. 7) has the same interpretation as π. Indeed the invariance of $[\![\pi]\!]$ under cut reduction is a key property, stated by the well-known theorem:

Theorem 11 (Soundness) *For every* π,π' *nets: if* $\pi\rightsquigarrow^*\pi'$, *then* $[\![\pi]\!]=[\![\pi']\!]$.

PROOF. A straightforward consequence of Lemma 17 (see [9]). □

The empty net has no conclusion and it has exactly one experiment: the function with empty domain. Thus the interpretation of the empty net is not the empty set, but the singleton of the empty sequence $\{(\,)\}$. By Theorem 11, this means that every net reducing to the empty net is interpreted by $\{(\,)\}$. Clearly there are nets having an empty interpretation, for example take any net with an head-clash: no experiment meets the cut condition of Fig. 6. More interesting examples of nets having an empty interpretation are those nets from which starts an infinite head-reduction sequence (as for example the net of Fig. 4).

The following definition introduces an equivalence relation $\sim$ on the experiments of a $\flat$-net π: intuitively the $\sim$ equivalence classes are made of experiments associating with a given !-link of π multisets of experiments with the same cardinality. This relation, as well as the notion of substitution defined immediately after, will play a role in Subsection 5.2.

Definition 12 *We define an* equivalence $\sim$ *on the set of experiments of a* $\flat$*-net* π*, by induction on* $\mathrm{depth}(\pi)$*. Let* $e, e' : \pi$*, we set* $e \sim e'$ *whenever for every* !*-node* o *of* $\mathrm{ground}(\pi)$*, there is* $m \in \mathbb{N}$*, s.t.:* $e(o) = [e_1, \ldots, e_m]$*,* $e'(o) = [e'_1, \ldots, e'_m]$*, and* $\forall j \leq m$*,* $e_j \sim e'_j$*.*

Notice that whenever π has depth 0, we have $e \sim e'$ for every $e, e' : \pi$. For an example with !-links, recall the experiment $e : \pi$ defined on the topmost net of Fig. 7: the $\sim$-equivalence class of e is the set of all experiments of π which associates a multiset of cardinality 4 with the left !-link and a multiset of cardinality 2 with the right !-link.

Definition 13 (Substitution) *A* substitution *is a function* $\sigma : D \to D$ *induced by a function* $\sigma^{\mathcal{A}} : \mathcal{A} \to D$ *and defined by induction on the depth of* D *elements, as follows (as usual* $p \in \{+, -\}$ *and* $a \in \mathcal{A}$*):*

$$\sigma(p, *) := (p, *) \qquad \sigma(+, a) := \sigma^{\mathcal{A}}(a) \qquad \sigma(-, a) := \sigma^{\mathcal{A}}(a)^{\perp}$$

$$\sigma(p, x, y) := (p, \sigma(x), \sigma(y)) \qquad \sigma(p, [x_1, \ldots, x_n]) := (p, [\sigma(x_1), \ldots, \sigma(x_n)])$$

We denote by $\mathcal{S}$ *the set of substitutions. If* $\mathbf{y} = (x_1, \ldots, x_n) \in \bigparr_{i=1}^{n} D$*, we set* $\sigma(\mathbf{y}) := (\sigma(x_1), \ldots, \sigma(x_n))$*.*

A similar notion of substitution plays a crucial role in [13]. In our setting, a crucial property is that the interpretation of a $\flat$-net is closed by substitution, as the following lemma shows (the proof is an easy induction on $s(\pi)$).

Lemma 14 *Let* π *be a* $\flat$*-net. For every* $e' : \pi$ *and* $\sigma \in \mathcal{S}$*, there is* $e : \pi$ *s.t.* $\sigma(|e'|) = |e|$ *and* $e \sim e'$*.*

3 The size of experiments

Experiments can be thought as objects in between syntax and semantics: by relating them precisely to head and stratified reductions, we make a first step in finding a semantic measure of execution time. The second (and last) step is the shift from experiments to their results, and this is precisely the purpose of Sections 4 and 5.
A central tool of the paper is Lemma 17, called *Key-lemma*, which points out that experiment sizes provide a counter for head and stratified reduction steps: the main achievement of this section is the proof of the Key-lemma.

Definition 15 (Experiment size) *For every* $\flat$*-net* π*, for every* $e : \pi$*, we* *define, by induction on depth(π), the* size of e, *$s(e)$ for short, as follows:*

$$s(e) = s(\text{ground}(\pi)) + \sum_{o \in !(\text{ground}(\pi))} \sum_{e^o \in e(o)} s(e^o) \quad .$$

Notice that the part of $s(e)$ which really depends on e is the number of copies e chooses for the !-links, the rest depends only on the $\flat$-net π. In particular we have the following immediate consequence of Definition 12:

Fact 16 *Let π be a $\flat$-net. For every $e, e' : \pi$ s.t. $e \sim e'$, we have $s(e) = s(e')$.*

Let's now give an example of size computation: recall the experiment $e : \pi$ on the topmost net of Figure 7, and suppose x, y are atomic, i.e. $x, y \in \{+, -\} \times \mathcal{A}$. We have: $s(e^o_1) = s(e^o_2) = s(e^u) = 3$ and then $s(e) = 8 + 18 = 26$.

In [9] p. 61-70, Girard shows that in the coherent semantics we have a notion of residue under cut elimination. Namely, he proves that if $\pi \rightsquigarrow \pi'$, then every experiment $e : \pi$ has a "residue" $\overrightarrow{e} : \pi'$ s.t. $|e| = |\overrightarrow{e}|$, as well as every experiment $e' : \pi'$ has an "ancestor" $\overleftarrow{e'} : \pi$, s.t. $|\overleftarrow{e'}| = |e'|$. This fact has as a consequence the invariance of the interpretation $[\![\pi]\!]$ under cut elimination (here Theorem 11). In the following lemma, we refine Girard's proof in the framework of **Rel**, by pointing out that, in case of head-reduction, not only e and $\overrightarrow{e}$ have the same result but also $s(\overrightarrow{e}) = s(e) - 2$. Such a new "quantitative" insight in the relationship between e and its "residue" $\overrightarrow{e}$ is at the core of our program to study computational properties by semantic means.
Before proving Lemma 17, let us consider an example. Take the experiment $e : \pi$ of Fig. 7 and consider $\pi \rightsquigarrow_h \pi_1$: the labelling of π_1's edges and !-links defines a "residue" $\overrightarrow{e} : \pi_1$ of e (at least according to the construction of residue given by Girard in [9]). The reader can check that $|\overrightarrow{e}| = |e|$ and $s(\overrightarrow{e}) = 6 + 18 = 24 = s(e) - 2$. This example shows that a notion of residue in the relational semantics would be more subtle to define than in the coherent semantics: let $e^{\overrightarrow{u}}_x$ (resp. $e^{\overrightarrow{u}}_y$) be the experiment of the box of π_1 which takes the values $x, x^\perp$ (resp. $y, y^\perp$) on both the axioms in the box, and let $\overrightarrow{e'}$ be the experiment of π_1 which differs from $\overrightarrow{e}$ on the !-link $\overrightarrow{u}$, where we set $\overrightarrow{e'}(\overrightarrow{u}) = [e^{\overrightarrow{u}}_x, e^{\overrightarrow{u}}_y]$. The experiment $\overrightarrow{e'}$ has the same "right" as $\overrightarrow{e}$ to be considered a residue of e (in particular one has $|e| = |\overrightarrow{e}| = |\overrightarrow{e'}|$). Not only an experiment could have several residues but it could also have several ancestors. Indeed, consider $\pi_1 \rightsquigarrow_h \pi_2$ and the experiment $e_2 : \pi_2$ defined by the labelling of π_2 in Fig. 7: both $\overrightarrow{e}$ and $\overrightarrow{e'}$ should be considered ancestors of e_2 (or, said the other way round, e_2 would be the residue of both $\overrightarrow{e}$ and $\overrightarrow{e'}$).
Let us comment a bit this very delicate phenomenon (many ancestors, many residues) by looking more carefully at the case of the different residues $\overrightarrow{e}$ and $\overrightarrow{e'}$ of e. What happens is that we have a multiset of 4 labels of an *ax*-link (the left box of π), and cut elimination requires that we split this multiset in two

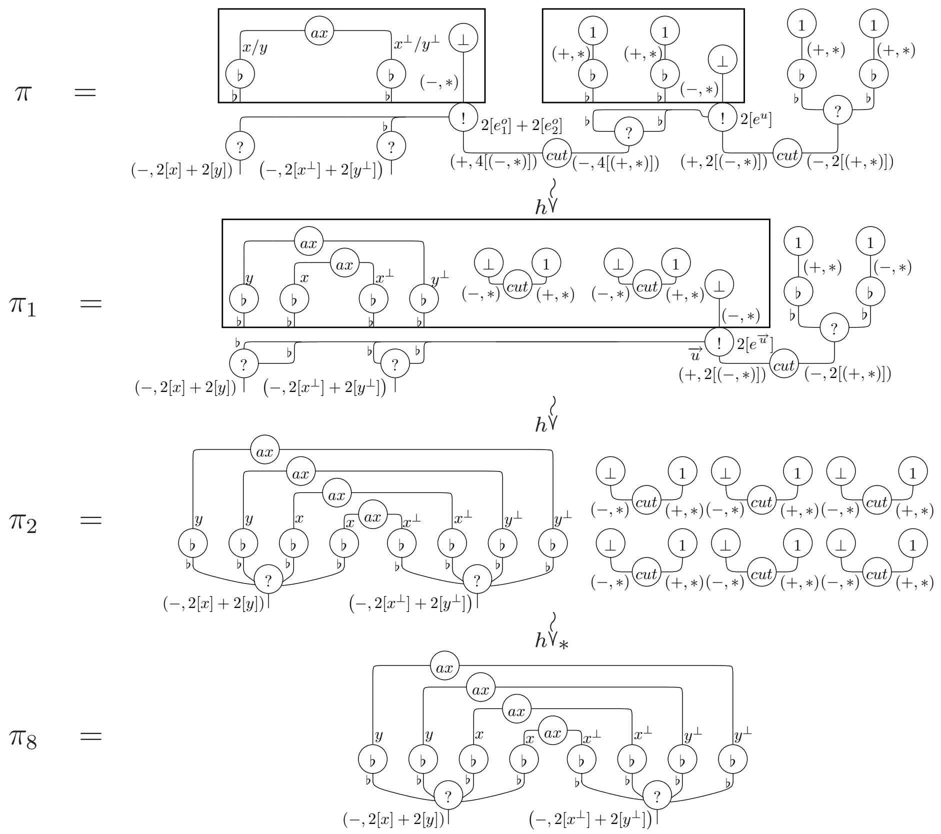


Fig. 7. Example of an experiment $e : \pi$ and its residues under cut elimination. The value of an experiment on an edge or !-link is written as a label of that edge/!-link. Inside the left box of π we use fractions x/y to describe different values of experiments: we write as numerator (resp. denumerator) the values of e_1^o (resp. e_2^o). For simplicity we have omitted the values on the structural edges.

---

multisets, each of which contains 2 labels. In **Rel**, there is no canonical way to operate such a splitting[12].

Lemma 17 (Key-lemma) *Let π, π' be two nets s.t. $\pi \rightsquigarrow_h \pi'$. Then:*

(1) for every $e : \pi$ there is $\overrightarrow{e} : \pi'$ s.t. $|e| = |\overrightarrow{e}|$, and $s(\overrightarrow{e}) = s(e) - 2$;

---

[12] This is in sharp contrast to the case of coherent semantics, where there exists a unique splitting of the original multiset. Actually, that's a way to express the so-called "uniformity" feature of coherent semantics: while a net can have different **Rel** experiments with the same result (as it is the case for $\overrightarrow{e}, \overrightarrow{e'} : \pi_1$), in the framework of Girard's coherence spaces there exists exactly one experiment for every point of the interpretation of a given net, and, consequently, the ancestor and the residue of an experiment are unique. We will have to take account of this in Subsection 5.2, see Proposition 35. Uniformity of coherent semantics was strongly exploited in [17] to prove injectivity of fragments of **LL**. But this is another story…

(2) for every $e' : \pi'$ *there is* $\overleftarrow{e'} : \pi$ *s.t.* $|\overleftarrow{e'}| = |e'|$*, and* $s\left(\overleftarrow{e'}\right) = s(e') + 2$.

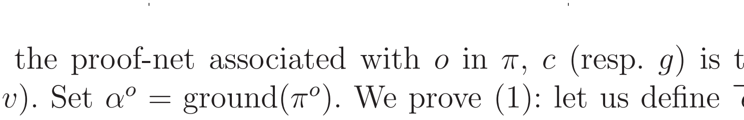
PROOF. Let $\pi \rightsquigarrow_h \pi'$ and t be the reduced cut of π. Remember that by definition of $\rightsquigarrow_h$, t has depth 0 in π. Let $\alpha = \text{ground}(\pi)$ and $\alpha' = \text{ground}(\pi')$. The proof splits in four cases, depending on the type of t: we consider only the case t is of type $(!/?)$, leaving to the reader the other cases ((ax), $(1/\bot)$, $(\otimes/\parr)$) which are easier.
If t is of type $(!/?)$, then our nets are as in the $(!/?)$ case of Fig. 3.[13] This case is delicate, since the !-link o dispatches several residues of its box π^o in π' (at any depth). Let $\flat(w)$ be the set of $\flat$-nodes associated with the ?-link w, we set $\text{depth}\flat(w) = \sum_{v\in\flat(w)}(\text{depth}(v)+1)$. The proof is by induction on $\text{depth}\flat(w)$.

Case depth$\flat$(w) = 0, i.e. w is a ?-link without premises. Let us prove (1): let us define $\overrightarrow{e} : \pi'$ from any $e : \pi$. If d' (resp. l') is an edge (resp. a !-link) of α', then d' (resp. l') is the residue of a unique edge d (resp. !-link l) of α. Moreover the $\flat$-structure associated with l' is the same as the one associated with l. So define $\overrightarrow{e}(d') = e(d)$ and $\overrightarrow{e}(l') = e(l)$. Notice that $\overrightarrow{e}$ is well-defined. Moreover, we have $|\overrightarrow{e}| = |e|$. As for the sizes, remark that $s(\alpha') = s(\alpha) - 2$, since a, b are the only two *logical* edges of α erased in α'. Moreover, since $e(o) = [\,]$, we deduce:

$$\sum_{l\in!(\alpha)}\ \sum_{e^l\in e(l)} s\left(e^l\right) = \sum_{\substack{l\in!(\alpha)\\ l\neq o}}\ \sum_{e^l\in e(l)} s\left(e^l\right) = \sum_{l'\in!(\alpha')}\ \sum_{e^{l'}\in\overrightarrow{e}(l')} s\left(e^{l'}\right)$$

We conclude: $s(\overrightarrow{e}) = s(e) - 2$.
Conversely, let us prove (2): consider $e' : \pi'$. Let d (resp. l) be any edge (resp. !-link) of π s.t. d is not a conclusion of o (resp. $l \neq o$). Then d (resp. l) has a unique residue d' (resp. l') in α', moreover the $\flat$-structure associated with l' is the same as the one associated with l. So set: $\overleftarrow{e'}(d) = e'(d')$ (resp. $\overleftarrow{e'}(l) = e'(l')$). Moreover define $\overleftarrow{e'}(o) = [\,]$ and $\overleftarrow{e'}(d) = (-, [\,])$ for every auxiliary conclusion d of o, $\overleftarrow{e'}(a) = (+, [\,])$ for the main conclusion a of o. Remark that $\overleftarrow{e'}$ is well-defined and check that $|\overleftarrow{e'}| = |e'|$, and $s\left(\overleftarrow{e'}\right) = s(e') + 2$.

Case depth$\flat$(w) = 1, i.e. w is a ?-link with only one premise which is conclusion of a $\flat$-node v in α. This means π, π' are as follows:



where π^o is the proof-net associated with o in π, c (resp. g) is the premise of w (resp. v). Set $\alpha^o = \text{ground}(\pi^o)$. We prove (1): let us define $\overrightarrow{e} : \pi'$ from

---

[13] To be precise, Fig. 3 deals with the general case where t is at any depth of π.

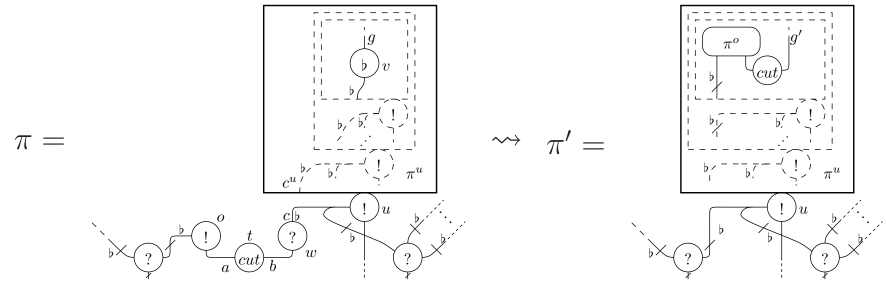
$e : \pi$. First of all remark that $e(o) = [e^o]$, since the multiset in $e(a)$ contains exactly one element (that is $e^o(a^o) = e(g)^\perp$). If d' (resp. l') is an edge (resp. a !-link) of α', then its ancestor d (resp. l) is in α or in α^o. In the first case, set: $\overrightarrow{e}(d') = e(d)$ (resp. $\overrightarrow{e}(l') = e(l)$); in the second case: $\overrightarrow{e}(d') = e^o(d)$ (resp. $\overrightarrow{e}(l') = e^o(l)$).
Clearly $|e| = |\overrightarrow{e}|$. Moreover notice that $s(\alpha') = s(\alpha) + s(\alpha^o) - 2$ (t's reduction erases the logical edges a and b), so that:

$$\begin{aligned}
s(\overrightarrow{e}) &= s(\alpha') + \sum_{l' \in !(\alpha')} \sum_{e^{l'} \in \overrightarrow{e}(l')} s\left(e^{l'}\right) \\
&= s(\alpha) + s(\alpha^o) - 2 + \sum_{l \in !(\alpha^o)} \sum_{e^l \in e^o(l)} s\left(e^l\right) + \sum_{l \in !(\alpha) \setminus \{o\}} \sum_{e^l \in e(l)} s\left(e^l\right) \\
&= s(\alpha) - 2 + s(e^o) + \sum_{l \in !(\alpha) \setminus \{o\}} \sum_{e^l \in e(l)} s\left(e^l\right) = s(e) - 2 \ .
\end{aligned}$$

We prove (2): let us define $\overleftarrow{e'} : \pi$ from $e' : \pi'$. Let d (resp. l) be an edge of α s.t. d is not a conclusion of o neither conclusion nor premise of w (resp. $l \neq o$). Then d (resp. o) has a unique residue d' (resp. l') in α': set $\overleftarrow{e'}(d) = e'(d')$ (resp. $\overleftarrow{e'}(l) = e'(l')$). Let e^o be the restriction of e' to π^o (which is a sub♭-net of π') and define $\overleftarrow{e'}(o) = [e^o]$, $\overleftarrow{e'}(a) = (+, [e^o(a^o)])$ and $\overleftarrow{e'}(b) = \overleftarrow{e'}(c) = \overleftarrow{e'}(a)^\perp = \left(-, [\overleftarrow{e'}(g)]\right)$, and finally, for every auxiliary conclusion f of o let f^o be the corresponding edge of π^o and set $\overleftarrow{e'}(f) = e^o(f^o)$. Remark that this definition of $\overleftarrow{e'}$ makes sense (i.e. $\overleftarrow{e'}$ is indeed an experiment). As in the former case, one can prove $|e'| = |\overleftarrow{e'}|$ and $s(e') = s\left(\overleftarrow{e'}\right) - 2$.

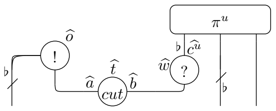
Case depth♭(w) > 1, i.e. either w has more than one premise, or it has exactly one premise and this premise is associated with a ♭-node in a !-link. We thus split in two subcases.

If w is associated with exactly one ♭-node v and v is in a !-link u, then π and π' have the following shape:



where π^o (resp. π^u) is the ♭-net associated with o (resp. u) in π, c (resp. g) is the premise of w (resp. v). Let now $e : \pi$ and let us define $\overrightarrow{e} : \pi'$. Let d'

(resp. l') be an edge (resp. a !-link) of α', then its ancestor d (resp. l) is in α. Moreover if $l' \neq u'$, then $\pi^{l'} = \pi^{l}$. So set: $\overrightarrow{e}(d') = e(d)$ and $\overrightarrow{e}(l') = e(l)$, when $l' \neq u'$. It remains to define $\overrightarrow{e}(u')$. For this, consider the following $\widehat{\pi}$:

$\widehat{\pi} =$ 

where π^o is associated with $\widehat{o}$. Remark that $\widehat{\pi} \rightsquigarrow_h \pi^{u'}$, so we can apply the induction hypothesis to $\widehat{\pi}$ (indeed depth$\flat(\widehat{w}) =$ depth$\flat(w) - 1$).[14] Let us define from $e : \pi$ an $\widehat{e} : \widehat{\pi}$. Set $\widehat{\alpha} = \text{ground}(\widehat{\pi})$. Let $e(o) = [e^o_1, \dots, e^o_h]$ and $e(u) = [e^u_1, \dots e^u_k]$, for $h, k \geq 0$. By definition, $e(b) = e(a)^\perp$, i.e. $\left(+, \sum_{i \leq h}[e^o_i(a^o)]\right)^\perp = \left(-, \sum_{j \leq k} \mu_j\right)$, where a^o is the conclusion of π^o associated with a, c^u is the conclusion of π^u associated with c, and for every $j \leq k$, $e^u_j(c^u) = (-, \mu_j)$. This means that $\sum_{i \leq h}[e^o_i(a^o)^\perp] = \sum_{j \leq k} \mu_j$, i.e. there is a function[15] $f : \{1, \dots, h\} \to \{1, \dots, k\}$, s.t. for every $j \leq k$, $\mu_j = \sum_{i \in f^{-1}(j)}[e^o_i(a^o)^\perp]$: let us fix such an f once for all. For each $j \leq k$, let $\widehat{e_j} : \widehat{\pi}$ be defined as follows:

- for every !-link $\widehat{l} \in \widehat{\alpha}$:
  - if $\widehat{l}$ is in π^u, set: $\widehat{e_j}(\widehat{l}) = e^u_j(\widehat{l})$,
  - otherwise $\widehat{l} = \widehat{o}$, then define: $\widehat{e_j}(\widehat{o}) = [e^o_i; i \in f^{-1}(j)]$,
- for every edge $\widehat{d} \in \widehat{\alpha}$:
  - if $\widehat{d}$ is in π^u, set: $\widehat{e_j}(\widehat{d}) = e^u_j(\widehat{d})$,
  - otherwise, $\widehat{d}$ is $\widehat{b}$ or a conclusion of $\widehat{o}$. Define: $\widehat{e_j}(\widehat{b}) = e^u_j(c^u)$, $\widehat{e_j}(\widehat{a}) = (+, [e^o_i(a^o) \text{ s.t. } i \in f^{-1}(j)])$, and for every other auxiliary conclusion $\widehat{d}$ of $\widehat{o}$, let $\widehat{e_j}(\widehat{d}) = \left(-, \sum_{i \in f^{-1}(j)} \nu_i\right)$, where d^o is the conclusion of π^o associated with $\widehat{d}$, and for every $i \in f^{-1}(j)$, $e^o_i(d^o) = (-, \nu_i)$.

Remark that $\widehat{e_j} : \widehat{\pi}$ is well-defined, in particular $\widehat{e_j}(\widehat{b}) = \widehat{e_j}(\widehat{a})^\perp$, since by definition of e_j and that of f, $\widehat{e_j}(\widehat{b}) = e^u_j(c^u) = (-, \mu_j) = \left(-, \sum_{i \in f^{-1}(j)}[e^o_i(a^o)^\perp]\right) = \widehat{e_j}(\widehat{a})^\perp$. Applying, for every $j \leq k$, the induction hypothesis to $\widehat{e_j} : \widehat{\pi}$, we obtain the existence of $e^{u'}_j : \pi^{u'}$, s.t. $|e^{u'}_j| = |\widehat{e_j}|$ and $s\left(e^{u'}_j\right) = s\left(\widehat{e_j}\right) - 2$.

Finally we can complete the definition of $\overrightarrow{e}$, by setting: $\overrightarrow{e}(u') = [e^{u'}_1, \dots, e^{u'}_k]$. We leave to the reader the proof that $\overrightarrow{e}$ is well defined and that $|e| = |\overrightarrow{e}|$. Let us prove instead that $s\left(\overrightarrow{e}\right) = s(e) - 2$. We know that $s(\alpha') = s(\alpha) - 2$, since a, b have been erased by t's reduction; moreover, for each $j \leq k$, $s\left(e^{u'}_j\right) = s\left(\widehat{e_j}\right) - 2$. Notice that, by the definition of $\widehat{e_j}$, we known that:

---

[14] Recall that cut elimination is defined on nets and not on $\flat$-nets, however we have adopted the convention to speak of *the cut elimination of a box π^o of a net π*, meaning the cut elimination of the net obtained by adding to π^o the ?-links of π associated with the structural conclusions of π^o.

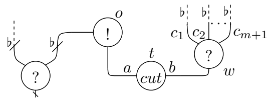
[15] Notice that this function is not necessarily unique (due to the fact that $[e^o_1(a^o)^\perp, \dots, e^o_h(a^o)^\perp]$ is a *multi*set), and this implies that $\overrightarrow{e}$ is not unique (and similarly for (2), $\overleftarrow{e}$ is not unique): recall the example of Figure 7.

$s\left(\widehat{e}_j\right) = \sum_{i\in f^{-1}(j)} s\left(e_i^o\right) + s\left(e_j^u\right) + 2$ (+2 since $\widehat{\pi}$ has the logical edges $\widehat{a}, \widehat{b}$ in addition to π^o and π^u). So, $s\left(e_j^{u'}\right) = \sum_{i\in f^{-1}(j)} s\left(e_i^o\right) + s\left(e_j^u\right)$, from which we conclude that:

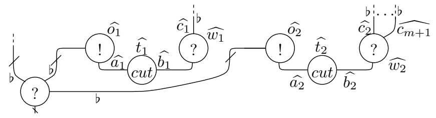
$$\begin{aligned} s\left(\overrightarrow{e}\right) &= s\left(\alpha'\right) + \sum_{\substack{l'\in !(\alpha') \\ e^{l'}\in \overrightarrow{e}(l')}} s\left(e^{l'}\right) = s\left(\alpha\right) - 2 + \sum_{\substack{l\in !(\alpha) \\ l\neq o,u \\ e^{l}\in e(l)}} s\left(e^{l}\right) + \sum_{e^{u'}\in \overrightarrow{e}(u')} s\left(e^{u'}\right) \\ &= s\left(\alpha\right) - 2 + \sum_{\substack{l\in !(\alpha) \\ l\neq o,u \\ e^{l}\in e(l)}} s\left(e^{l}\right) + \sum_{e^{o}\in e(o)} s\left(e^{o}\right) + \sum_{e^{u}\in e(u)} s\left(e^{u}\right) = s\left(e\right) - 2 \end{aligned}$$

The definition of an experiment $\overleftarrow{e'} : \pi$ from an experiment $e' : \pi'$ is completely symmetric to the definition of $\overrightarrow{e} : \pi'$ from $e : \pi$ and it is left to the reader.

If w has more than one premise, then π has the following shape:

$\pi =$ 

The proof of this case is an easy variant of the former one, we just sketch the proof here. The key ingredient is to define a structure $\widehat{\pi}$ obtained from π by substituting the above highlighted subgraph with the following one:

$\widehat{\pi} =$ 

where with both $\widehat{o_1}$, $\widehat{o_2}$ is associated the ♭-net π^o associated with o in π. Let $\widehat{\pi}'$ be the result of reducing $\widehat{t_1}$ in $\widehat{\pi}$, so that $\widehat{\pi} \rightsquigarrow_h \widehat{\pi}'$. Moreover notice that $\widehat{\pi}' \rightsquigarrow_h \pi'$, by reducing the residue of $\widehat{t_2}$ in $\widehat{\pi}'$. The next step is to show that from any any experiment $e : \pi$, one can define (similarly to the former case) an experiment $\widehat{e} : \widehat{\pi}$, s.t. $|\widehat{e}| = |e|$ and $s\left(\widehat{e}\right) = s\left(e\right) + 2$. Once we have $\widehat{e} : \widehat{\pi}$, we can apply the induction hypotheses on $\widehat{\pi}$ first, and on $\widehat{\pi}'$ thereafter (indeed depth♭$(\widehat{w_1})$, depth♭$(\widehat{w_2})$ $<$ depth♭(w)). In this way we get the experiments $\overrightarrow{\widehat{e}} : \widehat{\pi}'$ and $\overrightarrow{\overrightarrow{\widehat{e}}} : \pi'$, s.t. $s\left(\overrightarrow{\overrightarrow{\widehat{e}}}\right) = s\left(\overrightarrow{\widehat{e}}\right) - 2 = s\left(\widehat{e}\right) - 4 = s\left(e\right) - 2$. Set $\overrightarrow{e} = \overrightarrow{\overrightarrow{\widehat{e}}}$.

The definition of an experiment $\overleftarrow{e'} : \pi$ from an experiment $e' : \pi'$ is completely symmetric to the definition of $\overrightarrow{e} : \pi'$ from $e : \pi$. $\square$

We now adapt the Key-lemma to the stratified case. We introduce for this purpose the notion of *exhaustive element* of D.

Definition 18 (Exhaustive element) *Let $x \in D$. We say that x is* exhaustive *if* $(+, [\,])$ *does not appear in* x[16]*. An element* $(x_1, \ldots, x_n)$ *of* $\bigparr_{i=1}^{n} D$ *is exhaustive when* x_i *is exhaustive for every* $i \in \{1, \ldots, n\}$*. An* experiment *is* exhaustive *if its result is exhaustive. Given a set* $X \subseteq D$*, we denote by* X^{ex} *the set of the exhaustive elements of* X*.*

Clearly it might be the case that x is exhaustive while $x^{\perp}$ isn't. Notice also that the definition of *exhaustive experiment* only rely on the notion of *exhaustive point* of D: if π and π' are $\flat$-nets and if $e : \pi$ and $e' : \pi'$ are s.t. $|e| = |e'|$, then either e and e' are both exhaustive or they are both non exhaustive.

Given a net π, we are going to use exhaustive experiments $e_0 : \pi$ s.t. $s(e_0) = \min\{s(e);\ e : \pi \text{ is exhaustive}\}$. In case π is cut free, these exhaustive experiments are exactly the ones called 1-*experiments* in [17], as the following fact shows.

Fact 19 *Let* e_0 *be an exhaustive experiment of a head-cut free net* π *s.t.* $s(e_0) = \min\{s(e);\ e : \pi \textit{ is exhaustive}\}$*. For every* !*-link* o *of* $\mathrm{ground}(\pi)$*, there exists an exhaustive experiment* $e_0' : \pi^o$ *s.t.*

- $s(e_0') = \min\{s(e);\ e : \pi^o \textit{ is exhaustive}\}$
- *and* $e_0(o) = [e_0']$*.*

PROOF. By induction on $\mathrm{depth}(\pi)$. Let o be a !-link of π. If $e_0(o) = [\ ]$, then e_0 is not exhaustive. If $e_0(o) = [e_0^1, \ldots, e_0^n]$ with $n \geq 2$, then clearly $s(e_0)$ is not minimal. $\square$

Let us now refine our main tool: the Key-lemma (Lemma 17). Notice that by Definition 18 of exhaustive experiment, residues and ancestors of exhaustive experiments are exhaustive (the notions of residue and ancestor of an experiment refer to the discussion before Lemma 17).

Lemma 20 *Let* π *and* π' *be two nets s.t.* $\pi \rightsquigarrow_s \pi'$*. Then:*

(1) for every $e : \pi$ *exhaustive s.t.* $s(e) = \min\{s(e);\ e : \pi \textit{ is exhaustive}\}$*, there exists* $\overrightarrow{e} : \pi'$ *s.t.* $|e| = |\overrightarrow{e}|$ *and* $s\left(\overrightarrow{e}\right) = s\left(e\right) - 2$*;*
(2) for every $e' : \pi'$ *exhaustive s.t.* $s(e') = \min\{s(e);\ e : \pi \textit{ is exhaustive}\}$*, there exists* $\overleftarrow{e'} : \pi$ *s.t.* $|\overleftarrow{e'}| = |e'|$ *and* $s\left(\overleftarrow{e'}\right) = s\left(e'\right) + 2$*.*

PROOF. Let $\pi \rightsquigarrow_s \pi'$ and t be the reduced cut of π. We proceed by induction on $\mathrm{depth}(t)$. We prove only 1, the proof of 2 being symmetric. If $\mathrm{depth}(t) = 0$, then $\pi \rightsquigarrow_h \pi'$ and we can apply Lemma 17. Otherwise, let o be the !-link of $\mathrm{ground}(\pi)$ whose box π^o contains t: the structure $t(\pi^0)$ is a one step *stratified* reduct of π^o. Let $e : \pi$ be s.t. $s(e) = \min\{s(e');\ e' : \pi \text{ is exhaustive}\}$. Because

---

[16] We mean here that the ordered sequence of characters $(+, [])$ is not a subsequence of x (as a word).

by hypothesis the reduction step leading from π to π' is stratified, we know that π is head-cut free, which allows to apply Fact 19: we have $e(o) = [e^o]$ for some exhaustive experiment $e^o : \pi^o$ s.t. $s(e^o) = \min\{s(e'); e' : \pi^o \text{ is exhaustive}\}$. By induction hypothesis (applied to $e^o : \pi^o$ and $t(\pi^o)$) there exists an exhaustive experiment $\overrightarrow{e^o} : t(\pi^o)$ s.t. $|\overrightarrow{e^o}| = |e^o|$ and $s\left(\overrightarrow{e^o}\right) = s(e^o) - 2$. We then define $\overrightarrow{e}$ by changing the value of e on the !-link o (and leaving all the rest unchanged): we set $\overrightarrow{e}(o) = [\overrightarrow{e^o}]$. One clearly has $s\left(\overrightarrow{e}\right) = s\left(e\right) - 2$. $\square$

4 Qualitative account

In this section, we use Lemmas 17 and 20 to characterize, by semantic means, (head-)normalizable nets: this is Theorem 21, which can be seen as an extension of the well-known characterization of (head-)normalizable λ-terms by means of intersection types. We also answer the following question: if π and π' are two cut free nets connected by a cut link, is it the case that the thus obtained net is (head-)normalizable? The answer is given by only referring to $[\![\pi]\!]$ and $[\![\pi']\!]$ in Corollary 24. Quantitative versions of this last result will be proven in Section 5.

Theorem 21 *Let π be a net. We have:*

(1) π is head-normalizable iff $[\![\pi]\!]$ is non-empty;
(2) π is normalizable iff $[\![\pi]\!]^{ex}$ is non-empty.

PROOF. $(\Rightarrow)$: We prove only (2); the proof of (1) is an easy variant. Assume there is a cut free net π_0 such that $\pi \rightsquigarrow^* \pi_0$. Since π_0 is cut free, it is possible to define exhaustive experiments on π_0: assign (inductively w.r.t. depth(π)) a non-empty multiset of exhaustive experiments to each !-link at depth 0. Then $[\![\pi_0]\!]^{ex}$ is non-empty, and thus $[\![\pi]\!]^{\mathrm{ex}}$ is non-empty too by Theorem 11.
$(\Leftarrow)$: One proves a bit more than (1) (resp. (2)), by induction on $\min\{s(e); e : \pi\}$ (resp. on $\min\{s(e); e : \pi \text{ is exhaustive}\}$): if $[\![\pi]\!]$ (resp. $[\![\pi]\!]^{ex}$) is non-empty then there is π_0 head-cut free (resp. cut free) such that $\pi \rightsquigarrow_h^* \pi_0$ (resp. $\pi \rightsquigarrow_s^* \pi_0$), instead of simply $\pi \rightsquigarrow^* \pi_0$.
As for (1$\Leftarrow$), if π is head-cut free, then we set $\pi_0 = \pi$; otherwise let t be a cut at depth 0 of π: t is not a clash (if it were a clash then for its premises a, b, no experiment e could enjoy $e(a) = e(b)^{\perp}$, that is $[\![\pi]\!]$ would be empty[17]). Let π' be the result of the reduction of t: $[\![\pi']\!]$ is non-empty and, by Lemma 17, $\min\{s(e); e : \pi'\} < \min\{s(e); e : \pi\}$: by induction hypothesis there is a head-cut free net π_0 s.t. $\pi' \rightsquigarrow_h^* \pi_0$. We conclude $\pi \rightsquigarrow_h \pi' \rightsquigarrow_h^* \pi_0$.
As for (2$\Leftarrow$), if π is not cut free, suppose it contains a head-cut (the case π head-cut free is a simpler variant). By (the strong version just proved of)

---

[17] The fact that for every $x \in D$ one has $x \neq x^{\perp}$ plays here a crucial role.

(1), there is a head-cut free net π' s.t. $\pi \rightsquigarrow_h^* \pi'$. We have that $[\![\pi']\!]^{ex}$ is not empty and, by Lemma 20, $\min\{s(e); e : \pi' \text{ is exhaustive}\} < \min\{s(e); e : \pi \text{ is exhaustive}\}$. Since π' is head-cut free, we can apply Fact 19: for every !-link o of π' with associated box π'^o one has that ($[\![\pi'^o]\!]^{ex}$ is non empty and that) $\min\{s(e); e : \pi'^o \text{ is exhaustive}\} < \min\{s(e); e : \pi' \text{ is exhaustive}\}$. The induction hypothesis applied to every such box π'^o yields $\pi \rightsquigarrow_h^* \pi' \rightsquigarrow_s^* \pi_0$, for some π_0 cut free. □

Theorem 21 allows to extend to nets the so-called "safeness" property of the leftmost reduction strategy in the pure λ-calculus: if a net is normalizable, its normal form can always be reached by a stratified reduction sequence.

Corollary 22 *Let π be a net. If $\pi \rightsquigarrow^* \pi_0$ for some π_0 cut free, then $\pi \rightsquigarrow_s^* \pi_0$.*

PROOF. By confluence (see [14]) a normalizable net π has a unique normal form π_0. By Theorem 21 $[\![\pi]\!]^{\mathrm{ex}}$ is non-empty, and still by (the proof of (2$\Leftarrow$) of) that theorem $\pi \rightsquigarrow_s^* \pi_0$.

The following definition introduces a notation used in the sequel.

Definition 23 *Le π and π' be two nets. Let c be a conclusion of π and let c' be a conclusion of π'. We denote by $(\pi|\pi')_{c,c'}$ the net obtained by connecting π and π' by means of a cut-link with premises c and c'.*

By the way, let us notice that for every net π with cuts one can prove by induction on π that there exists two *cut free* nets π_1 and π_2, s.t. $(\pi|\pi')_{c,c'} \rightsquigarrow_s^* \pi$, for some c (resp. c') conclusion of π_1 (resp. π_2).
Theorem 21 allows to characterize, in terms of $[\![\pi]\!]$ and $[\![\pi']\!]$, those couples of nets (π, π') s.t. $(\pi|\pi')_{c,c'}$ is (head-)normalizable.

Corollary 24 *Let π (resp. π') be a net with conclusions $\mathbf{d}, c$ (resp. $\mathbf{d}', c'$).*

(1) There is $\mathbf{x}$, $\mathbf{x}' \in \mathbf{D}$, $x \in D$ s.t. $(\mathbf{x}, x) \in [\![\pi]\!]$ and $(\mathbf{x}', x^\perp) \in [\![\pi']\!]$, iff there is a head-cut free net π'' s.t. $(\pi|\pi')_{c,c'} \rightsquigarrow_h^* \pi''$.
(2) There is $\mathbf{x}, \mathbf{x}' \in \mathbf{D}^{ex}$, $x \in D$ s.t. $(\mathbf{x}, x) \in [\![\pi]\!]$ and $(\mathbf{x}', x^\perp) \in [\![\pi]\!]$, iff there is a cut free net π'' s.t. $(\pi|\pi')_{c,c'} \rightsquigarrow_s^* \pi''$.

5 Quantitative account

We now turn our attention to the "quantitative" aspects of cut elimination. The aim is to give a purely semantic account of execution time. Of course, if $\pi_1 \rightsquigarrow^* \pi_2$ we know that $[\![\pi_1]\!] = [\![\pi_2]\!]$, so that from $[\![\pi_1]\!]$ it is clearly impossible to determine the number of steps leading from π_1 to π_2. Nevertheless, if π and

π' are two cut free nets connected by means of a cut link, we can wonder what is the number of cut elimination steps leading from the net with cut to a cut free one. We prove in this section that we can answer the question by only referring to $[\![\pi]\!]$ and $[\![\pi']\!]$. We solve the problem for both the head-reduction and the stratified reduction (Theorems 32 and 36).

We first (Subsection 5.1) give a quantitative insight into the correspondence reduction/experiment: Proposition 28 allows to recover the number of steps of a reduction from the size of an experiment. However, this is not a way to compute by purely semantic means the number of execution steps of a net: the method we look for has to refer only to the results of experiments. This shift is performed by Theorem 32 which gives a (purely semantic) bound for the length of head and stratified reduction sequences. The last Subsection 5.2 is devoted to improve Theorem 32 and eventually yields a semantic way to compute the exact length of head and stratified reduction sequences.

Definition 25 (Size of an element) *For every $x \in D$, we define the* size $s(x)$ *of x, by induction on* $\mathrm{depth}(x)$. *Let $p \in \{+, -\}$,*

- *if $x = (p, a)$ and $a \in \mathcal{A} \cup \{*\}$, then $s(x) = 1$;*
- *if $x = (p, y, z)$, then $s(x) = 1 + s(y) + s(z)$;*
- *if $x = (p, [x_1, \ldots, x_m])$, then $s(x) = 1 + \sum_{j=1}^{m} s(x_j)$;*

Given $(x_1, \ldots, x_n) \in \bigparr_{i=1}^{n} D$ $(n \geq 0)$, we set $s(x_1, \ldots, x_n) = \sum_{i=1}^{n} s(x_i)$.

Notice that for every point $x \in D$ or $x \in \bigparr_{i=1}^{n} D$, $s(x)$ is the number of occurrences of $+$, $-$ in x (seen as a word).

5.1 An upper bound to cut elimination

In this subsection we first compute the exact length of head and stratified reduction sequences by means of experiments (Proposition 28), which immediately implies that all these sequences have the same length (Corollary 29). We then prove the first truly semantic measure of execution time by bounding (by purely semantic means) the length of head and stratified reduction sequences (Theorem 32).

Definition 26 *For every $\flat$-net π, we set $s_0(\pi) = \inf\{s(\mathbf{x})\ ;\ \mathbf{x} \in [\![\pi]\!]\}$ and $s_1(\pi) = \inf\{s(\mathbf{x})\ ;\ \mathbf{x} \in [\![\pi]\!]^{ex}\}$.*

Notice that $s_0(\pi)$ (resp. $s_1(\pi)$) is an integer which only depends on $[\![\pi]\!]$ (resp. $[\![\pi]\!]^{ex}$). Moreover, if $[\![\pi]\!]$ is empty, both $s_0(\pi)$ and $s_1(\pi)$ are equal to ∞.[18] Consider the nets of Fig. 7: we have $s_0(\pi) = s_1(\pi) = 10$, which is equal to

---

[18] This remark holds since we have defined s_0/s_1 by using the inf function and not the min function: the min is undefined on the empty set, while inf gives as value ∞.

s_0/s_1 of every π's reduct. Indeed, an immediate consequence of Theorem 11 is that when $\pi \rightsquigarrow^* \pi'$ one has $s_0(\pi) = s_0(\pi')$ and $s_1(\pi) = s_1(\pi')$.

Lemma 27 *Let π be a $\flat$-net with k structural conclusions.*

(1) If π head-cut free, then: $s_0(\pi) = s(\mathrm{ground}(\pi)) + k = \min\{s(e); e : \pi\} + k$.
(2) If π cut free, then: $s_1(\pi) = s(\pi) + k = \min\{s(e); e : \pi \textit{ is exhaustive}\} + k$.

PROOF. We prove only (1), the proof of (2) being an easy variant.
Let us define an experiment $e_0 : \pi$ by associating with every !-link of ground(π) the empty multiset and with every pair of conclusions of an ax-link the pair of elements $(+, *), (-, *)$ (it does not matter in which order)[19]. Since π is head-cut free, e_0 is well-defined. Observe that $s(|e_0|) = s(\mathrm{ground}(\pi)) + k$ (this can be proven by an easy induction on $s(\mathrm{ground}(\pi))$). Moreover, we have also $s(|e_0|) = \inf\{s(|e|); e : \pi\}$, $s(e_0) = \min\{s(e); e : \pi\}$, and $s(e_0) = s(\mathrm{ground}(\pi))$. We then deduce: $s_0(\pi) = \inf\{s(|e|);\ e : \pi\} = s(|e_0|) = s(\mathrm{ground}(\pi)) + k = s(e_0) + k = \min\{s(e)\ ;\ e : \pi\} + k$. $\square$

Proposition 28 *Let π be a net and let π' (resp. π'') be a head-cut free (resp. cut free) net.*

(1) For every reduction sequence $R : \pi \rightsquigarrow_h^* \pi'$, and every $e_0 : \pi$ s.t. $s(e_0) = \min\{s(e)\ ;\ e : \pi\}$ we have $\mathrm{length}(R) = (s(e_0) - s_0(\pi))/2$
(2) For every reduction sequence $R : \pi \rightsquigarrow_s^* \pi''$, and every $e_1 : \pi$ s.t. $s(e_1) = \min\{s(e)\ ;\ e : \pi \textit{ is exhaustive}\}$ we have $\mathrm{length}(R) = (s(e_1) - s_1(\pi))/2$

PROOF. Again we prove only (1), the proof of (2) being an easy variant[20].
Because π' is head-cut free, it is always possible to define an experiment of π' (associate anything to the conclusions of ax-links and the emptyset to every !-link of π'). From $\llbracket\pi\rrbracket = \llbracket\pi'\rrbracket$ (Th. 11) one deduces that there exists $e : \pi$. Let $e_0 : \pi$ be s.t. $s(e_0) = \min\{s(e)\,;\, e : \pi\}$. The proof is by induction on length(R). Assume length(R) $= 0$, i.e. $\pi = \pi'$. By Lemma 27 one has $s_0(\pi) = s(e_0)$.
Assume length(R) $= n > 0$, i.e. $R = \pi \rightsquigarrow_h \pi_1 \rightsquigarrow_h^* \pi'$. By Lemma 17, there is an experiment $\overrightarrow{e_0} : \pi_1$ s.t. $|\overrightarrow{e_0}| = |e_0|$, and $s(\overrightarrow{e_0}) = s(e_0) - 2$. Still by Lemma 17, if $e_1 : \pi_1$ then there exists $\overleftarrow{e_1} : \pi$ s.t. $s(e_1) = s(\overleftarrow{e_1}) - 2$. Then $s(\overrightarrow{e_0}) = \min\{s(e)\,; e : \pi_1\}$. By Theorem 11, we have $\llbracket\pi\rrbracket = \llbracket\pi_1\rrbracket$ hence $s_0(\pi) = s_0(\pi_1)$. We can then apply the induction hypothesis to π_1 ($\pi_1 \rightsquigarrow_h^* \pi'$ in $n - 1$

---

[19] In the proof of (2), π is cut-free and the experiment e_0 has to be exhaustive. In this case we define e_0 by induction on depth(π), as follows:
- with every pair of conclusions of every ax-link, e_0 associates the pair of elements $(+, *)$, $(-, *)$ (it does not matter in which order);
- with every !-link o, e_0 associates the singleton $[e_0^o]$, where e_0^o is an experiment defined as e on π^o (notice that $\mathrm{depth}(\pi^o) < \mathrm{depth}(\pi)$).

[20] The proof of (2) uses Lemma 20 instead of Lemma 17.

steps and $\min\{s\,(e)\,;e:\pi_1\} = s\,(\overrightarrow{e_0})$):

$$n-1 = \frac{s(\overrightarrow{e_0}) - s_0(\pi_1)}{2} = \frac{s(e_0) - 2 - s_0(\pi)}{2} = \frac{s(e_0) - s_0(\pi)}{2} - 1 \qquad \Box$$

The reader can check Propoposition 28 with the nets of Fig. 7: $s_0(\pi) = 10$, $s(e_0) = 26$, and indeed every head-reduction sequence from π to π_8 consists of 8 head-reduction steps. An immediate consequence of Prop. 28 is the following:

Corollary 29 *Let π be a net, and π_0^1, π_0^2 (resp. π_1^1, π_1^2) be two head-cut free (resp. cut free) nets.*

(1) For every $R^1 : \pi \rightsquigarrow_h^* \pi_0^1$, $R^2 : \pi \rightsquigarrow_h^* \pi_0^2$, we have $\mathrm{length}(R^1) = \mathrm{length}(R^2)$.
(2) For every $R^1 : \pi \rightsquigarrow_s^* \pi_1^1$, $R^2 : \pi \rightsquigarrow_s^* \pi_1^2$, we have $\mathrm{length}(R^1) = \mathrm{length}(R^2)$.

We now turn to prove Theorem 32: by using purely semantic data, we can bound the number of head/stratified reduction steps. This is a simple consequence of the above Proposition 28 and the next statements.

Fact 30 *Let π be a $\flat$-net with $k+1$ conclusions s.t.* $\mathrm{ground}(\pi)$ *is a !-link o. Set $e(o) = [e_1, \ldots, e_m]$. We have: $s(|e|) - (k+1) = \sum_{j=1}^m (s(|e_j|) - k)$.*

PROOF. For every conclusion c_i of o ($i \le k+1$), let c_i^o be the corresponding conclusion of the $\flat$-net π^o associated with o. Let moreover c_1 be the main conclusion of o. We have $s(e(c_1)) = 1 + \sum_{j=1}^m s(e_j(c_1^o))$; as for the auxiliary conclusions (i.e. $1 < i \le k+1$), we have $s(e(c_i)) = \sum_{j=1}^m s(e_j(c_i^o)) - (m-1)$. We thus deduce:

$$s(|e|) - (k+1) = \sum_{i=1}^{k+1} s(e(c_i)) - (k+1) = \sum_{j=1}^{m}(\sum_{i=1}^{k+1} s(e_j(c_i^o)) - k) = \sum_{j=1}^{m}(s(|e_j|) - k)$$

The following lemma shows that the size of every experiment on a *cut free* $\flat$-net is at most the size of its result. More precisely, if π has no structural conclusions and $e : \pi$, then $s\,(e) \le s\,(|e|)$:

Lemma 31 *Let π be a cut free $\flat$-net with k structural conclusions and let $e : \pi$. Then we have $s(e) \le s(|e|) - k$.*

PROOF. The proof is by induction on $s(\pi)$. Assume that $\mathrm{ground}(\pi)$ is a !-link o with k structural conclusions. Set $e(o) = [e_1, \ldots, e_m]$ and let π^o be the box of o. Notice that π has $k+1$ conclusions. We have

$$\begin{aligned} s(e) &= 1 + \sum_{j=1}^{m} s(e_j) \leq 1 + \sum_{j=1}^{m} (s(|e_j|) - k) && \text{(by induction hypothesis)} \\ &= 1 + s(|e|) - (k+1) \ = \ s(|e|) - k && \text{(by Fact 30)} \end{aligned}$$

The other cases are left to the reader. □

Theorem 32 *Let π, π' be cut free nets, with conclusions resp.* $\mathbf{d}$, c *and* $\mathbf{d}'$, c'.

(1) For every head-cut free net π'', for every reduction sequence $R : (\pi|\pi')_{c,c'} \rightsquigarrow^*_h \pi''$, if $\mathbf{y} = (\mathbf{x}, x) \in \llbracket \pi \rrbracket$ and if $\mathbf{y}' = (\mathbf{x}', x^\perp) \in \llbracket \pi' \rrbracket$, then we have $\mathrm{length}(R) \leq (s(\mathbf{y}) + s(\mathbf{y}'))/2$.

(2) For every cut free net π'', for every reduction sequence $R : (\pi|\pi')_{c,c'} \rightsquigarrow^*_s \pi''$, for every $\mathbf{x}$, $\mathbf{x}' \in \mathbf{D}^{ex}$ and $x \in D$, if $\mathbf{y} = (\mathbf{x}, x) \in \llbracket \pi \rrbracket$ and if $\mathbf{y}' = (\mathbf{x}', x^\perp) \in \llbracket \pi' \rrbracket$, then we have $\mathrm{length}(R) \leq (s(\mathbf{y}) + s(\mathbf{y}'))/2$.

PROOF. We can prove (1) and (2) at once. There is $e : \pi$ s.t. $|e| = \mathbf{y}$ and there is $e' : \pi'$ s.t. $|e'| = \mathbf{y}'$. So, there exists $e'' : (\pi|\pi')_{c,c'}$ s.t. $s(e'') = s(e) + s(e')$. We have by Proposition 28 and Lemma 31:

$$\mathrm{length}(R) \ \leq \ \frac{s(e'')}{2} \ = \ \frac{s(e) + s(e')}{2} \ \leq \ \frac{s(|e|) + s(|e'|)}{2} \ = \ \frac{s(\mathbf{y}) + s(\mathbf{y}')}{2} \ . \qquad \square$$

5.2 The exact length of cut elimination

This last subsection is devoted to compute the exact length of head and stratified reduction sequences by purely semantic means. With the notations of Theorem 32, say that $\mathbf{x} \in \llbracket \pi \rrbracket$ and $\mathbf{x}' \in \llbracket \pi' \rrbracket$ are *compatible* when $\mathbf{x} = (\mathbf{z}, z) \in \llbracket \pi \rrbracket$ and $\mathbf{x}' = (\mathbf{z}', z^\perp) \in \llbracket \pi' \rrbracket$. For arbitrary compatible elements $\mathbf{x} \in \llbracket \pi \rrbracket$ and $\mathbf{x}' \in \llbracket \pi' \rrbracket$, it is clearly impossible to obtain an equality in Theorem 32, because there exist compatible elements with different sizes.

The only equality we have by now is that of Proposition 28, which uses the size of the experiments. A first idea is then to look for compatible elements $\mathbf{x}$ and $\mathbf{x}'$ whose sizes are equal to the sizes of the experiments used in Proposition 28: let us call these elements *suitable*. But there are cases in which compatible elements do exist but suitable compatible elements do not. Take for example an axiom as π and two axiom links followed by a ⅋ and a $\otimes$ as π' (a "η-expansion" of an axiom). In this case, all experiments on π have the same size and the same holds for the experiments on π'; hence $\mathbf{x} = |e|$ and $\mathbf{x}' = |e'|$ are suitable iff $s(|e|) = s(e)$ and $s(|e'|) = s(e')$; but one can easily check that if $s(|e|) = s(e)$ and $s(|e'|) = s(e')$, then $|e|$ and $|e'|$ *are not* compatible.

A more subtle way out is nevertheless possible, and here is where the notions of equivalence between experiments and of substitution defined in Subsection 2.2 come into the picture. As a matter of fact, we do not need the compatible elements to be suitable; it is enough that when there exist two compatible elements $\mathbf{x}$ and $\mathbf{x}'$ of $[\![\pi]\!]$ and $[\![\pi']\!]$, one can compute (using only data contained in $[\![\pi]\!]$ and $[\![\pi']\!]$) the size of the experiments with results $\mathbf{x}, \mathbf{x}'$[21]. More precisely, using the notion of substitution, Proposition 35 (the only place where we use the infinity of $\mathcal{A}$ through Lemma 33) shows how to find in $[\![\pi]\!]$, for every $\mathbf{x} \in [\![\pi]\!]$, a "suitable element w.r.t. $\mathbf{x}$" that is an element $\mathbf{y} \in [\![\pi]\!]$ such that $s(\mathbf{y}) = \min \{s(e) \ ; \ e : \pi \text{ and } |e| = \mathbf{x}\}$. By considering the least size of such $\mathbf{y} \in [\![\pi]\!]$ and $\mathbf{y}' \in [\![\pi']\!]$ *w.r.t.* $\mathbf{x} \in [\![\pi]\!]$ *and* $\mathbf{x}' \in [\![\pi']\!]$ *compatible* one obtains the exact length of head- and stratified reduction sequences starting from $(\pi|\pi')_{c,c'}$: this is Theorem 36.

Lemma 33 *Assume $\mathcal{A}$ is infinite. Let π be a cut free $\flat$-net with k structural conclusions (and possibly other logical conclusions), and let $e : \pi$. There exist $e' \sim e$ and a substitution σ (i.e. $\sigma \in \mathcal{S}$) s.t. $s(e') = s(|e'|) - k$ and $\sigma(|e'|) = |e|$.*

PROOF. We prove, by induction on $s(\pi)$, that for every infinite subset $\mathcal{A}'$ of $\mathcal{A}$, there is an experiment $e' \sim e$ s.t.: (1) $s(e') = s(|e'|) - k$; (2) $\sigma(|e'|) = |e|$ for some $\sigma \in \mathcal{S}$; (3) every element of $\mathcal{A}$ occurring in $|e'|$ is an element of $\mathcal{A}'$. Suppose ground(π) is a !-link o (the other cases are easier and left to the reader). Let π^o be the box of o and set $e(o) = [e_1, \ldots, e_m]$. Let $\mathcal{A}_1, \ldots, \mathcal{A}_m$ be infinite, pairwise disjoint, subsets of $\mathcal{A}'$,[22] by induction hypothesis there is $e'_j \sim e_j$ for every $j \leq m$ s.t. points $1-3$ hold (for point 3, we choose for every $j \in \{1, \ldots, m\}$ as $\mathcal{A}'$ the set $\mathcal{A}_j$). In particular there is $\sigma_j \in \mathcal{S}$ s.t. $\sigma_j(|e'_j|) = |e_j|$. Define $e'(o) = [e'_1, \ldots, e'_m]$.
We now have to show that e' satisfies points $1-3$. For point 3, just remember that $\bigcup_{i \leq m} \mathcal{A}_i \subseteq \mathcal{A}'$. As for point 2, we know by induction hypothesis that, for every $j \leq m$, $\sigma_j(|e'_j|) = |e_j|$. Since $\mathcal{A}_1, \ldots, \mathcal{A}_m$ are pairwise disjoint, $\bigcup_{j \leq m} \sigma_j|_{\mathcal{A}_j}$ is a function φ from $\bigcup_{1 \leq j \leq m} \mathcal{A}_j$ to D. Let σ be the substitution induced by φ (remember Definition 13 of substitution), we have $\sigma \in \mathcal{S}$ and $\sigma(|e'|) = |e|$ (this is actually the key point of the proof). Concerning point 1, we have:

$$\begin{aligned} s(e') &= 1 + \sum_{j=1}^{m} s(e'_j) \ = \ 1 + \sum_{j=1}^{m} s(|e'_j|) - k) && \text{(by induction hypothesis)} \\ &= 1 + s\,(|e'|) - (k+1) \ = \ s(|e'|) - k && \text{(by Fact 30)} \quad \square \end{aligned}$$

Notice that in the proof of Lemma 33 we used in an essential way the fact that $\mathcal{A}_1, \ldots, \mathcal{A}_m$ are pairwise disjoint. If this were not the case, a conflict in the definition of σ could occur: if one had $x \in \mathcal{A}_{j_1} \cap \mathcal{A}_{j_2}$ and $\sigma_{j_1}(x) \neq \sigma_{j_2}(x)$, then one would be in trouble when trying to define σ from $\sigma_1, \ldots, \sigma_m$.

---

[21] This can be easily done for the previous example (axiom and η-expansion).
[22] Such $\mathcal{A}_1, \ldots, \mathcal{A}_m$ exist for m arbitrary large since $\mathcal{A}$ is infinite.

Lemma 34 *Assume $\mathcal{A}$ is infinite. Let π be a cut free net and let $e : \pi$. We have* $s(e) = \min\{s(|e'|) \,;\, e' \sim e$ *and* $\exists \sigma \in \mathcal{S}$ *s.t.* $\sigma(|e'|) = |e|\}$.

PROOF. Choose $e'_0 : \pi$ s.t. $s(|e'_0|) = \min\{s(|e'|) \,;\, e' \sim e$ and $\exists \sigma \in \mathcal{S}$ *s.t.* $\sigma(|e'|) = |e|\}$. By Lemma 31 and Fact 16, $s\,(e) = s\,(e'_0) \leq s\,(|e'_0|)$. Thus we have $s(e) \leq \min\{s(|e'|) \,;\, e' \sim e$ and $\exists \sigma \in \mathcal{S}$ *s.t.* $\sigma(|e'|) = |e|\}$.
By Lemma 33 and Fact 16, we have the opposite inequality. $\square$

Proposition 35 *Assume $\mathcal{A}$ is infinite. Let π be a cut free net and let* $\mathbf{x} \in [\![\pi]\!]$. *We have* $\min\{s(e) \,;\, e : \pi, |e| = \mathbf{x}\} = \min\{s(|e'|) \,;\, e' : \pi, \exists \sigma \in \mathcal{S}, \sigma(|e'|) = \mathbf{x}\}$.

PROOF. Set $r = \min\{s(|e'|) \,;\, e' : \pi$ and $\exists \sigma \in \mathcal{S}, \sigma(|e'|) = \mathbf{x}\}$, and $q = \min\{s(e) \,;\, |e| = \mathbf{x}\}$.
First, we prove that $q \leq r$. Let $e' : \pi$ be such that $\exists \sigma \in \mathcal{S}, \sigma(|e'|) = \mathbf{x}$. By Lemma 14, there exists $e : \pi$ such that $|e| = \mathbf{x}$ and $e \sim e'$. This means that if we take $e'_0 : \pi$ s.t. $s\,(|e'_0|) = r$, there exists $e_0 \sim e'_0$ s.t. $|e_0| = \mathbf{x}$. By Fact 16 and Lemma 31: $q \leq s\,(e_0) = s\,(e'_0) \leq s\,(|e'_0|) = r$.
The proof of $r \leq q$ is easier: let $e : \pi$ be s.t. $s\,(e) = q$. By Lemma 34, $s\,(e) = \min\{s(|e'|) \,;\, e' \sim e$ and $\exists \sigma \in \mathcal{S}$ *s.t.* $\sigma(|e'|) = |e|\} \geq \min\,\{s(|e'|) \,;\, \exists \sigma \in \mathcal{S}$ *s.t.* $\sigma(|e'|) = |e|\}$. $\square$

The point of Theorem 36 is that the length of every head (resp. stratified) reduction sequence starting from $(\pi|\pi')_{c,c'}$ (where π and π' are cut free nets) and leading to a head-cut free (resp. cut free) net can be determined from $[\![\pi]\!]$ and $[\![\pi']\!]$. W.r.t. the discussion at the beginning of this subsection, notice that here the compatibility of $\sigma(\mathbf{y}) \in [\![\pi]\!]$ and $\sigma'(\mathbf{y}') \in [\![\pi']\!]$ is expressed by stating $\sigma(\mathbf{y}_c) = \sigma'(\mathbf{y}'_{c'})^\perp$ (the notations $\mathbf{y}_c$ and $\mathbf{y}_\mathbf{d}$ were introduced in Def. 10).

Theorem 36 *Assume $\mathcal{A}$ is infinite. Let π (resp. π') be a cut free net with conclusions $\mathbf{d}, c$ (resp. $\mathbf{d}', c'$).*

(1) For every head-cut free net π'', and for every head reduction sequence R from $(\pi|\pi')_{c,c'}$ to π'', the value of $\mathrm{length}(R)$ *is:*

$$\min\left\{ \frac{s(\mathbf{y}) + s(\mathbf{y}') - s_0((\pi|\pi')_{c,c'})}{2} \;;\; \begin{array}{l} \mathbf{y} \in [\![\pi]\!], \mathbf{y}' \in [\![\pi']\!] \textit{ s.t.} \\ \exists \sigma \in \mathcal{S} \textit{ s.t. } \sigma(\mathbf{y}_c) = \sigma(\mathbf{y}'_{c'})^\perp \end{array} \right\}$$

(2) For every cut free net π'', and for every stratified reduction sequence R from $(\pi|\pi')_{c,c'}$ to π'', the value of $\mathrm{length}(R)$ *is*

$$\min\left\{ \frac{s(\mathbf{y}) + s(\mathbf{y}') - s_1((\pi|\pi')_{c,c'})}{2} \;;\; \begin{array}{l} \mathbf{y} \in [\![\pi]\!], \mathbf{y}' \in [\![\pi']\!] \textit{ s.t.} \\ \exists \sigma \in \mathcal{S} \textit{ s.t. } \sigma(\mathbf{y}_c) = \sigma(\mathbf{y}'_{c'})^\perp \textit{ and} \\ \sigma(\mathbf{y}_\mathbf{d}), \sigma(\mathbf{y}'_{\mathbf{d}'}) \textit{ are exhaustive} \end{array} \right\}$$

PROOF. We only prove statement (1). The only difference occurring in the proof of statement (2) is the presence of exhaustive points and experiments. In that case we use in a crucial way the fact that "exhaustivity" is a property of experiments depending only on their results (by Definition 18).

We have $\mathrm{length}(R) = (q - s_0((\pi|\pi')_{c,c'}))/2$ with

$$
\begin{aligned}
q &= \min\left\{ s(e) + s(e') \ ; \ e : \pi, e' : \pi' \text{ s.t. } |e| = (\mathbf{z}, z) \text{ and } |e'| = \left(\mathbf{z}', z^{\perp}\right) \right\} \\
&\quad \text{(by Propositon 28)} \\
&= \min\left\{ s(\mathbf{y}) + s(\mathbf{y}') \ ; \ \begin{array}{c} \mathbf{y} \in [\![\pi]\!], \mathbf{y}' \in [\![\pi']\!] \text{ s.t.} \\ \exists\, (\mathbf{z}, z) \in [\![\pi]\!], \exists \left(\mathbf{z}', z^{\perp}\right) \in [\![\pi']\!], \exists \sigma \in \mathcal{S} \\ \text{s.t. } \sigma(\mathbf{y}) = (\mathbf{z}, z) \text{ and } \sigma(\mathbf{y}') = \left(\mathbf{z}', z^{\perp}\right) \end{array} \right\} \\
&\quad \text{(by applying Proposition 35 twice; the points of } [\![\pi]\!] \text{ and } [\![\pi']\!] \text{ we look} \\
&\quad \text{for are among those } \mathbf{y} \in [\![\pi]\!] \text{ and } \mathbf{y}' \in [\![\pi']\!] \text{ with disjoint atoms)} \\
&= \min\left\{ s(\mathbf{y}) + s(\mathbf{y}') \ ; \ \begin{array}{l} \mathbf{y} \in [\![\pi]\!], \mathbf{y}' \in [\![\pi']\!] \text{ s.t.} \\ \exists \sigma \in \mathcal{S} \text{ s.t. } \sigma(\mathbf{y})_c = \sigma(\mathbf{y}')^{\perp}_{c'} \end{array} \right\} \\
&\quad \text{(since } [\![\pi]\!], [\![\pi']\!] \text{ are closed by substitution, see Lemma 14).}
\end{aligned}
$$

We conclude by noting that the conditions $\exists \sigma \in \mathcal{S}$ s.t. $\sigma(\mathbf{y})_c = \sigma(\mathbf{y}')^{\perp}_{c'}$ and $\exists \sigma \in \mathcal{S}$ s.t. $\sigma(\mathbf{y}_c) = \sigma(\mathbf{y}'_{c'})^{\perp}$ are equivalent. □

Remark 37 *(1) In [14] the distinction between correct proof-structures (nets) and the non correct ones (simply "structures") is crucial: the very careful handling of the acyclicity condition of Definition 3 was very helpful for a rather sharp understanding of cut elimination in the framework of structures/nets. We could do the same here (and we did it in the preliminary version of this work [8]), because Theorem 36 can be proven for structures.*
However, if we eventually decided to restrict to nets, this is not only in order to have more standard (even though less general) results, but also because we wouldn't gain that much with structures. Indeed, in Theorem 36 we would in any case need to assume that a cut free *normal form of a given structure does exist, which amounts to eliminate the "non correct" part of the computation, i.e. our structure "computationally behaves" like...a net!*

(2) In presence of simple types (propositional formulas), the notion of η-expanded net can be defined: simply consider axiom links typed by atomic formulas. An immediate consequence of the restriction to such nets is that the notion of substitution becomes useless and the statement of Theorem 36 can be simplified (just erase every reference to substitutions).

References

[1] P. Baillot. Stratified coherent spaces: a denotational semantics for light linear logic. *Theor. Comput. Sci.*, 318(1–2):29–55, 2004.

[2] M. Coppo, M. Dezani-Ciancaglini, and B. Venneri. Principal type schemes and λ-calculus semantics. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, 1980.

[3] V. Danos. *La Logique Linéaire appliquée à l'étude de divers processus de normalisation*. Ph.D. thesis, Université Paris 7, 1990.

[4] V. Danos and L. Regnier. Proof-nets and the Hilbert space. In *Advances in Linear Logic*, volume 222 of *London Math. Soc. Lecture Note Ser.* Cambridge University Press, 1995.

[5] D. de Carvalho. Execution time of lambda-terms via non-uniform semantics and intersection types. Preprint Institut de Mathématique de Luminy 2006.

[6] D. de Carvalho. Intersection types for light affine lambda calculus. *Electr. Notes Theor. Comput. Sci.*, 136:133–152, 2005.

[7] D. de Carvalho. *Sémantiques de la logique linéaire et temps de calcul*. Thèse de doctorat, Université Aix-Marseille II, 2007.

[8] D. de Carvalho, M. Pagani, and L. Tortora de Falco. A semantic measure of the execution time in linear logic. RR 6441, INRIA, 2008.

[9] J.-Y. Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

[10] J.-Y. Girard. Light linear logic. *Information and Computation*, 143(2), 1998.

[11] O. Laurent and L. Tortora de Falco. Obsessional cliques: a semantic characterization of bounded time complexity. In *Proceedings of the 21st annual IEEE symposium on Logic In Computer Science*, 2006.

[12] A. S. Murawski and C.-H. L. Ong. Discreet games, light affine logic and ptime computation. In *Proceedings of the 14th annual EACSL conference, Computer Science Logic, Fischbachau, Germany*, volume 1862 of *Lecture Notes in Comput. Sci.* Springer, 2000.

[13] M. Pagani. Proofs, denotational semantics and observational equivalences in multiplicative linear logic. *Math. Structures Comput. Sci.*, 17(2):341–359, 2007.

[14] M. Pagani and L. Tortora de Falco. Strong normalization property for second order linear logic. Submitted, 2007.

[15] L. Regnier. *Lambda-Calcul et Réseaux*. Ph.D. thesis, Université Paris 7, 1992.

[16] K. Terui. *Light logic and polynomial time computation*. Ph.D. thesis, Keio University, 2002.

[17] L. Tortora de Falco. Obsessional experiments for linear logic proof-nets. *Math. Structures Comput. Sci.*, 13(6):799–855, 2003.