Università degli Studi
Roma Tre
Dipartimento di Filosofia

Université Paris Diderot
– Paris 7
U.F.R. d'Informatique –
Laboratoire PPS

Tesi di dottorato in
Filosofia e Teoria delle
Scienze Umane

Thèse de doctorat en
Sciences Mathématiques
de Paris Centre –
specialité Informatique

Differential nets, experiments and reduction

Giulio Guerrieri

Supervisors:	Lorenzo Tortora de Falco (Roma Tre) Thomas Ehrhard (Paris 7)
Referees:	Laurent Regnier Marcelo Fiore
Jury:	Vito Michele Abrusci Thomas Ehrhard Damiano Mazza Lorenzo Tortora de Falco

20th June 2013

Contents

Introduction

Linear Logic

Starting from semantical investigations about λ-calculus, Girard introduced in 1986 ([Gir87]) Linear Logic (LL), a refinement of intuitionistic and classical logic allowing a fine analysis of the use of resources during the cut-elimination (i.e. execution via Curry-Howard isomorphism) process of proofs (i.e. programs via Curry-Howard correspondence): by means of the introduction of the new connectives ! and ?, LL gives a logical *status* to duplication and erasure operations (corresponding to structural rules of intuitionistic and classical logic). One of the novelties of LL is its representation of proof by means of some particular graphs, the nets, giving a more geometrical account of the cut-elimination/execution process. Of course, not all the elements of this particular class of graphs, the proof-structures, are nets, i.e. correspond to proofs of LL's sequent calculus, but there exists a correctness criterion ([DR89], among others) characterizing all (and only in certain frameworks of LL) the proof-structures which are nets. So, proof-structures become some interesting objects in themselves from a computational point of view, in virtue of their geometrical aspect. Actually, proof structures still display some sequentialized aspects because of the presence of boxes, which define erasable or duplicable resources during the cut-elimination process. A box, indeed, must contain a proof structure that satisfies some constrains: all its conclusions, called auxiliary port, except one, called principle port, must be conclusion of a ?-link. This provides a deeply inductive character to proof structures.

Revisiting the syntax of nets. A first contribution of my thesis is revisiting the syntax of LL nets. Inspired by the presentation of nets developed in [dCT12] according to the formalism of interaction nets introduced by Lafont in [Laf95], I defined a syntax for the multiplicative and exponential framework of LL proof structures, in which there is not an explicit constructor for boxes; so, it's possible to recover boxes by means of some functions (the arrows) in a "purely geometrical" way under suitable conditions: these arrows associate with every !-link its auxiliary doors and the cuts contained in it at depth 0. Such a geometrical approach allows to define proof structures in a non inductive way, bringing out the Girard's original inner meaning of Linear Logic ([Gir87]). Differently from [dCT12], my definition allows to define also the cut-elimination procedure for proof structures: the exponential reduction step involves the composition of arrows. In these proof structures I defined also a correctness criterion (connection and acyclicity) in a completely non inductive (that is "purely geometric") way. The syntax introduced in this thesis turns out to be compatible also with the cut-elimination procedure for differential nets of DiLL_0 (see below), but not those of DiLL, where a box might contain a sum of nets.

Differential nets and Taylor expansion. In [Ehr05] Ehrhard defines a denotational semantics for λ-calculus and LL proof nets: the finiteness spaces, in which types (formulas) are interpreted by vectorial spaces and λ-terms (LL nets) by infinitely differentiable functions defined as power series (i.e. Taylor expansions) on these spaces. Differentiation can be internalized at a syntactical level thanks to an extension of the λ-calculus with differential operators: the differential λ-calculus, introduced by Ehrhard and Regnier in [ER03]. The authors have then extended the differential operators to LL nets, obtaining the differential nets (DiLL, [ER06b]), where the differential constructors assume an interesting form: they correspond to "symmetrizing" the exponential connectives. This means that the rules related to the two modalities ! and ? are perfectly symmetric, apart from the promotion rule. The differential versions of λ-calculus and LL allow a finer analysis of the use of resources during the computation process. We call Λ^{res} (resp. DiLL_0) the fragment of differential λ-calculus (resp. DiLL) characterized by the fact that only linear applications are possible (resp. by the fact that there are no boxes). Linearity, that is the absence of a promotion rule, entails that every term of Λ^{res} (resp. every net of DiLL_0) is trivially strongly normalizable (their sizes strictly decrease under reduction). Furthermore, linearity also causes that a term in Λ^{res} (resp. a DiLL_0 net) reduces to a sum of terms in Λ^{res} (resp. a sum of DiLL_0 nets), because a linear resource (it can be used exactly once), required from several parts, determines a plurality of possible choices. Hence, for every term in $\mathsf{\Lambda}^{\mathsf{res}}$ (resp. DiLL_0 net) t, its normal form $\mathsf{NF}(t)$ exists and it is a finite linear combination of λ-terms in Λ^{res} (resp. DiLL_0 nets). Λ^{res} (resp. DiLL_0) can be seen as an analysis tool for λ-calculus (resp. LL nets), thanks to the Taylor expansion $(\,)^*$, which associates with every λ-term (resp. LL net) a (potentially infinite) sum of terms in Λ^{res} (resp. DiLL_0 nets). In [ER08] it is proved that, given a every ordinary λ-term M, one can sum up all the normal forms of the resource λ-terms in M^*. Thus one obtains the normal form $\mathsf{NF}(M^*)$ of M^*, a (in general) infinite linear combination of terms in Λ^{res} with relational coefficients. In [ER06a] it is showed that $\mathsf{NF}(M^*)$ is the Taylor expansion of the Böhm tree $\mathsf{BT}(M)$ of M

(the notion of Taylor expansion is naturally extended to Böhm trees), that is

$$\mathsf{NF}(M^*) = (\mathsf{BT}(M))^* \tag{1}$$

In other words, the Taylor expansion commutes with normalization, where normalizing an ordinary λ-term means here computing its Böhm tree. The Böhm tree of a ordinary lambda term can be seen as the normal form of the head linear reduction which is the call-by-name reduction implemented by (a version of) the Krivine's abstract machine ([Kri07]) .

Taking advantage of a separation theorem for differential nets ([MP07]), I have demonstrated that the Taylor expansion (without taking into account the coefficients) commutes with the cut-elimination process. This means that, for every LL net π, one has $\mathsf{NF}(\pi^*) = (\mathsf{NF}(\pi))^*$ (the analogous of equation (1) for LL nets). At the same time Mazza and Pagani have shown two distinct DiLL_0 nets ρ and ρ' in the Taylor expansion π^* of a LL net π whose respective normal forms $\mathsf{NF}(\rho)$ and $\mathsf{NF}(\rho')$ are not disjoint. This example shows the difference of the case of λ-calculus with respect to LL: indeed, a crucial passage in the proof of the equation (1) in [ER08, ER06a] (for the λ-calculus) consists in defining a coherence relation among the λ-terms with sources such that:

- for every ordinary λ-term M, all the elements of M^* are coherent two by two among them
- if t and t' are coherent λ-terms with sources, then $\mathsf{NF}(t)$ and $\mathsf{NF}(t')$ are disjoint.

The Mazza and Pagani's example shows that it is impossible to define such a relation on LL nets.

Differential nets and experiments of relational semantics. In my thesis I tried to understand precisely and rigorously the strict relationship between differential nets without boxes (i.e. resources λ-terms) and experiments of LL nets. An experiment (notion introduced in [Gir87] and studied in detail in [Tor00, Tor03]) is a function which permits to associate with every LL net π a point of the interpretation of π in the relational model, the interpretation of π being the set of points resulting from all the possible experiments of π. Experiments, hence, act as a bridge between syntax and semantics. Among all the points of relational semantics of a LL net π, some of them are "more important": the injective points, that is those in which every their atom occurs exactly twice. If π is without cuts, the injective points are the results of experiments that associates with every axiom a different element of the web. From the injective points it is possible to reconstruct every other point by substitution. Furthermore two injective points of the interpretation of π can uniquely differ for the atom names, showing the "same structure".

Then, we say that the two injective points are equivalent and that the one can be transformed in the other by a suitable substitution of atoms with atoms. Given a net π of LL, we denote by $[\![\pi]\!]$ the subset of the relational interpretation of π, formed by the injective points quotiented modulo the injective substitutions. In collaboration with Tortora de Falco and Pellissier, I have demonstrated that the Taylor expansion of a cut-free η-expansed LL net π coincides with $[\![\pi]\!]$. In other words, a differential net in the Taylor expansion of a cut-free η-expansed LL net π is a canonical representative of an equivalence class of injective points of the relational interpretation of π. It remains to be investigated the meaning of a differential net in the Taylor expansion of a LL net π with cuts. In this case the difficulty is that it can reduce itself in several differential nets without cuts.

In collaboration with Tortora de Falco and Pellissier I have characterized the relations of the relational model corresponding to interpretations of some acyclic and connected LL net. In fact, if two cut-free η-expansed LL nets π_1 and π_2, acyclic and connected have the same 2-point in the Taylor expansion (i.e. the the differential net obtained recursively taking for every box two copies of its contents), then $\pi_1 = \pi_2$. In other words, a cut-free, η-expanded, acyclic and connected LL net is completely characterized by the 2-point in its Taylor expansion. This result simplifies the proof of injectivity of relational semantics with respect to LL (see [dCT12]) in the acyclic and connected case. Moreover thanks to this result it is possible to define an algorithm that, given a relation of the relational model, takes its 2-point α (if it exists) and tries to recover a cut-free, η-expansed, acyclic and connected LL net. If this procedure ends successfully, then one has found the only cut-free and η-expansed LL net that has α in its relational interpretation; otherwise, no acyclic and connected LL net has α in its interpretation. This result of surjectivity is based on the fundamental hypothesis of connection, pointing out the importance of this notion.

Call-by-value lambda-calculus

In the ordinary (also called "call-by-name") λ-calculus, the prototype of any functional programming language, the values are either variables or abstractions (λ-terms of the shape $\lambda x M$). So, the λ-terms are either values or applications (λ-terms of the shape MN). The "call-by-value" λ-calculus is the version of λ-calculus allowing to reduce only the β_{v}-redexes, i.e. β-redexes of the shape $(\lambda x M)V$ where V is a value. The call-by-value λ-calculus was introduced by Plotkin in '70 ([Plo75]) in order to give a version of λ-calculus closer to the real implementation of functional programming languages. The relationship between call-by-value λ-calculus and Linear Logic was widely studied for the first time by Maraist, Wadler *et al.* in [MOTW95] in '90.

Recently in [Ehr12] Ehrhard introduced a version (called Λ_{CBV}) of the call-by-value λ-calculus such that values and terms are disjoint sets defined by mutual induction: a value is either a variable or an abstraction λxM where M is a term, a term is either an application $(M)N$ where M and N are terms, or a "promoted" value $V^!$ where V is a value. This distinction can be explained from the Linear Logic point of view: in [Ehr12] Ehrhard presented a general notion of denotational model for Λ_{CBV} corresponding to the translation $(\,)^{\mathsf{b}}$ defined as "boring" by Girard ([Gir87]) of the intuitionistic logic into LL, whereby $(A \Rightarrow B)^{\mathsf{b}} = !\mathsf{A}^{\mathsf{b}} \multimap !B^{\mathsf{b}}$ (thus in the untyped case, the intuitionistic isomorphism $o \simeq (o \Rightarrow o)$ becomes $o \simeq (!o \multimap !o)$). In my thesis I studied the relationship between Λ_{CBV} and LL from a syntactical point of view (already implicit in [Ehr12]). I defined the translation of terms and values of Λ_{CBV} into LL nets: the idea is that a "promoted" value corresponds to a box in the LL nets, therefore a β_{v}-redex $(\lambda xM)^!V^!$ corresponds to a cut between the box representing $(\lambda xM)^!$ and a dereliction (the application is linear on the left); this allows the box representing $V^!$ to get in the net representing M and duplicate at will. In general, one step of β_{v}-reduction in Λ_{CBV} corresponds to several steps of cut-elimination in LL-nets.

Reduction and call-by-value Krivine's machine. Trees. In [Ehr12] Ehrhard proved that the interpretation of a term M in Λ_{CBV} is empty if and only if M is strongly normalizable for the $\hat{\beta}_{\mathsf{v}}$-reduction, where the $\hat{\beta}_{\mathsf{v}}$-reduction (or weak β_{v}-reduction) is a restriction of the β_{v}-reduction obtained by forbidding reductions under abstractions. This result is the analogous of the well-known theorem for the ordinary (i.e. call-by-name) λ-calculus whereby a term is head normalizable if and only if its interpretation in the Engler's denotational model is not empty. In my thesis I developed a survey about $\hat{\beta}_{\mathsf{v}}$-reduction, in order to see to what extent the $\hat{\beta}_{\mathsf{v}}$-reduction can be considered in Λ_{CBV} as a analogue of the head reduction in ordinary λ-calculus. A first difference is obvious: in the ordinary λ-calculus the head redex of any term, if any, is unique, whereas a term in Λ_{CBV} might have several $\hat{beta}_{\mathsf{v}}$-redexes (in LL-nets they correspond to cuts at depth 0), but these $\hat{\beta}_{\mathsf{v}}$-redexes are not overlapping, hence the $\hat{\beta}_{\mathsf{v}}$-reduction is strongly confluent. Therefore, one can define a parallel $\hat{\beta}_{\mathsf{v}}$-reduction reducing in one step all the $\hat{\beta}_{\mathsf{v}}$-redexes: if a term M is $\hat{\beta}_{\mathsf{v}}$-normalizable, then the parallel $\hat{\beta}_{\mathsf{v}}$-reduction reduces M to its $\hat{\beta}_{\mathsf{v}}$-normal form. So, the fact of having several $\hat{\beta}_{\mathsf{v}}$-redxes is not a substantial difference with respect to the head reduction of ordinary λ-calculus.

The structure of a term M can be represented by a binary tree, called the applicative tree of M: it breaks up the applications in M until to "promoted" values which are subterms of M (they are the leaves of the applicative tree of M). So, the $\hat{\beta}_{\mathsf{v}}$-redexes of any term are characterizable as the nodes whose left (resp. right) child is a "promoted" abstraction (resp. "promoted" value). The notions of applicative tree and parallel $\hat{\beta}_{\mathsf{v}}$-reduction suggest the definition of a tree-like structure which is similar to a Böhm tree for the "call-by-value"

λ-calculus. Nowadays for the "call-by-value" λ-calculus there does not yet exist a notion of Böhm tree (see for example the recent [NGP12]).

In my thesis I also defined an abstract machine for Λ_{CBV} similar to the Krivine's abstract machine for the ordinary (i.e. call-by-name) λ-calculus define in [Kri07, DR04]. The abstract machines play an important role in implementing programming languages because on the one hand they are "sufficiently abstract" to relate easily to the notion of reduction of λ-calculus, on the other hand they are closer to executions of a real machine, by imposing among other things a precise reduction strategy. I introduced two call-by-value Krivine's machines K^{l} and K^{r}: I showed that the K^{l} (resp. K^{r}) machine with an input term M will search for the leftmost (resp. rightmost) $\hat{\beta}_{\mathsf{v}}$-redex in the applicative tree of M and then reduce it. By the good proprieties of the $\hat{\beta}_{\mathsf{v}}$-reduction, if a closed term M is $\hat{\beta}_{\mathsf{v}}$-normalizable then its $\hat{\beta}_{\mathsf{v}}$-normal form computed by K^{l} and K^{r}; actually this result holds more generally for any "random" call-by-value Krivine's machine at each execution step chooses whether to apply the left or right reduction strategy.

Translations, σ-equivalence and σ-reduction. There exists two continuation passing style (CPS) translations of Λ_{CBV} into the ordinary (i.e. call-by-name) λ-calculus: $(\,)^{\mathsf{l}}$ (already defined in [Plo75, Sel01]) and $(\,)^{\mathsf{r}}$ (introduced in my thesis), whose only difference is in the translation of application, more precisely in the choice of putting the function (in case of $(\,)^{\mathsf{l}}$) or the argument (in case of $(\,)^{\mathsf{r}}$). I showed that, modulo these CPS translations, the β_{v}-reduction corresponds to the β-reduction of ordinary λ-calculus. The following result is more interesting: the call-by-value Krivine's machine K^{l} (resp. K^{r}) is simulated by the call-by-name Krivine's machine modulo the CPS translation $(\,)^{\mathsf{l}}$ (risp. $(\,)^{\mathsf{r}}$).

In the ordinary λ-calculus, σ-equivalence ([Reg92, Reg94]) equates terms differing only in their sequential structure but behaving the same. The σ-equivalence can be characterized by encoding λ-terms into LL nets by means of the Girard's "call-by-name" translation $(A \Rightarrow B) \rightsquigarrow (!A \multimap B)$: two λ-terms are σ-equivalents if and only if their translations into LL nets are the same. I proved an analogous result for Λ_{CBV} by means of the "boring" translation of the intuitionistic arrow $(A \Rightarrow B) \rightsquigarrow (!A \multimap !B)$ into LL. The σ_{v}-equivalence relation thus obtained on terms and values in $\Lambda_{\mathsf{CBV}}$is not included in β_{v}-equivalence, differently from what happens in ordinary λ-calculus, where σ-equivalence is included in β-equivalence.

Even more surprisingly, I showed that it is possible to give an orientation to two of the three rules generating σ_{v}-equivalence, in such a way to get a "completion" of β_{v}-reduction: the add of the σ-reduction rules allows to simulate the Accattoli and Paolini's call-by-value λ-calculus with explicit substitutions (λ_{vsub}, introduced in [AP12]). One of the novelties of λ_{vsub} is that it allows to characterize the solvable terms by means of internal (i.e.

call-by-value) reduction rules. Thanks to the simulation, it is reasonable to expect that the solvability is characterizable in Λ_{CBV} by means of internal (i.e. call-by-value) reduction rules without using explicit substitutions.

Part I

Linear Logic

Chapter 1

A non inductive syntax

This section is devoted to present in full details the syntactical object for which we prove our main result: proof-structure (definition 52). We adopt the interaction nets point of view (see for example [Laf95, ER06b, Pag09, Tra11, dCT12]) and pass through intermediate objects: cell-bases (definition 1), pre-pre-proof-structures (definition 16), pre-proof-structures (definition 43). Our approach, definitions and notations are those of [dCT12] (in particular, our syntactical objects are untyped as in [LT06, PT10, dCPT11]) up to some differences that will be explained in the following. Essentially the principal novelties with respect to the syntax of [dCT12] are:

- our framework can represent DiLL-proof-structures, which are the differential generalization (where boxes and duals of ?-links are allowed, see for example [ER06b, MP07, Pag09, Tra11]) of the MELL-proof-structures (the multiplicative-exponential framework of Linear Logic, see for example [Gir87, DR95, Tor03, dCPT11, dCT12]);
- our objects are not necessarily cut-free; moreover it is possible to define the cut-elimination in two frameworks of our syntax, the DiLL_0-proof-structures (the DiLL-proof-structures without boxes) and the MELL-proof-structures; this fact answers positively to the difficulties raised in [dCT12] about the definition of cut-elimination on untyped proof-structures;
- our definition of proof-structure is completely non-inductive, so a proof-structure is precisely a labeled hyper-graph; the boxes are computed by starting from its principal door and by using only some "geometrical informations" in this hyper-graph; our geometrical point of view is strengthen by our choice of untyped syntactical objects.

Notation. We set $\mathcal{T} = \{1, \bot, \otimes, \parr, !, ?\}$ whose elements are the *connectives of the multiplicative and exponential framework of Linear Logic*. We say that 1, $\bot$, $\otimes$, $\parr$ (resp. !, ?) are the *multiplicative* (resp. *exponential*) connectives, and 1, $\bot$ are the *units*.

1.1 Cells and ports

In the following definition of cell-base, we introduce cells and ports. This definition differs from that one in [dCT12] only because in our cell-base there is not the function #: this means that the word "linear" used in [dCT12] makes no sense in our syntax.

Definition 1 (Module-base, (pseudo-)cell-base)**.** *A* module-base *is a* 5*-tuple* $\mathbb{C} = (\mathsf{t}, \mathcal{P}, \mathsf{C}, \mathsf{P}^{\mathsf{pri}}, \mathsf{P}^{\mathsf{left}})$ *such that:*

- t *is a function such that* $\mathsf{dom}(\mathsf{t})$ *is a finite set and* $\mathsf{codom}(\mathsf{t}) = \mathcal{T} \cup \{ax\}$*; we set* $\mathcal{C}(\mathbb{C}) = \mathsf{dom}(\mathsf{t})$ *whose elements are the* cells of $\mathbb{C}$*; for every* $l \in \mathcal{C}(\mathbb{C})$*,* $\mathsf{t}(l)$ *is the* label of l*; for every* $t, t' \in \mathcal{T}$*, we set* $\mathcal{C}^t(\mathbb{C}) = \{l \in \mathcal{C}(\mathbb{C}) \mid \mathsf{t}(l) = t\}$ *(whose elements are the* t-cells of $\mathbb{C}$*) and* $\mathcal{C}^{t,t'}(\mathbb{C}) = \mathcal{C}^t(\mathbb{C}) \cup \mathcal{C}^{t'}(\mathbb{C})$*;*
- $\mathcal{P}$ *is a finite set whose elements are the* ports of $\mathbb{C}$*; we set* $\mathcal{P}(\mathbb{C}) = \mathcal{P}$*;*
- $\mathsf{C} : \mathcal{P}(\mathbb{C}) \to \mathcal{C}(\mathbb{C})$ *is a surjection such that for every* $l \in \mathcal{C}(\mathbb{C})$*,*
  - *if* $\mathsf{t}(l) \in \{1, \bot, ax\}$ *then* $\mathsf{card}(\{p \in \mathcal{P}(\mathbb{C}) \mid \mathsf{C}(p) = l\}) = 1$*,*
  - *if* $\mathsf{t}(l) \in \{\otimes, \parr\}$ *then* $1 \leq \mathsf{card}(\{p \in \mathcal{P}(\mathbb{C}) \mid \mathsf{C}(p) = l\}) \leq 3$*;*

  for every $l \in \mathcal{C}(\mathbb{C})$*, we set* $\mathcal{P}_l(\mathbb{C}) = \{p \in \mathcal{P}(\mathbb{C}) \mid \mathsf{C}(p) = l\}$ *whose elements are the* ports of l*;*
- $\mathsf{P}^{\mathsf{pri}} : \mathcal{C}(\mathbb{C}) \to \mathcal{P}(\mathbb{C})$ *is a function such that* $\mathsf{C} \circ \mathsf{P}^{\mathsf{pri}} = id_{\mathcal{C}(\mathbb{C})}$*; for every* $l \in \mathcal{C}(\mathbb{C})$*,* $\mathsf{P}^{\mathsf{pri}}(l)$ *is the* principal port *(or* conclusion*)* of l*, moreover we set* $\mathcal{P}^{\mathsf{aux}}_l(\mathbb{C}) = \mathcal{P}_l(\mathbb{C}) \setminus \{\mathsf{P}^{\mathsf{pri}}(l)\}$ *whose elements are the* auxiliary ports *(or* premises*)* of l*, and* $\mathsf{a}_{\mathbb{C}}(l) = \mathsf{card}(\mathcal{P}^{\mathsf{aux}}_l(\mathbb{C}))$ *which is the* arity of l*; we set* $\mathcal{P}^{\mathsf{pri}}(\mathbb{C}) = \mathsf{im}(\mathsf{P}^{\mathsf{pri}})$ *whose elements are the* principal ports of $\mathbb{C}$*, and* $\mathcal{P}^{\mathsf{aux}}(\mathbb{C}) = \bigcup_{l \in \mathcal{C}(\mathbb{C})} \mathcal{P}^{\mathsf{aux}}_l(\mathbb{C})$ *whose elements are the* auxiliary ports of $\mathbb{C}$*; we set* $\mathcal{C}^{\otimes,\parr}_2(\mathbb{C}) = \{l \in \mathcal{C}^{\otimes,\parr}(\mathbb{C}) \mid \mathsf{a}_{\mathbb{C}}(l) = 2\}$*.*
- $\mathsf{P}^{\mathsf{left}} : \mathcal{C}^{\otimes,\parr}_2(\mathbb{C}) \to \mathcal{P}^{\mathsf{aux}}(\mathbb{C})$ *is a function such that, for every* $l \in \mathcal{C}^{\otimes,\parr}_2(\mathbb{C})$*, one has* $\mathsf{P}^{\mathsf{left}}(l) \in \mathcal{P}^{\mathsf{aux}}_l(\mathbb{C})$*.*

A pseudo-cell-base *is a module-base such that* $\mathcal{C}^{\otimes,\parr}(\mathbb{C}) = \mathcal{C}^{\otimes,\parr}_2(\mathbb{C})$*.*
A cell-base *is a pseudo-cell-base such that* $\mathcal{C}^{ax}(\mathbb{C}) = \emptyset$*.*
We denote by **ModuleBases** *(resp.* **PseudoCells***;* **Cells***) the set of module-bases (resp. pseudo-cell-bases; cell-bases).*

Intuitively, a module-base corresponds to a set of "links with their premises and conclusions" in the standard theory of linear logic proof-nets (see for example [Gir87, DR95, Tor03, Pag09, dCPT11]). More precisely, cells correspond to links, the principal port of a cell corresponds to the conclusion of a link and an auxiliary port of a cell corresponds to a premise of a link.

Note that our presentation reformulates the linear logic "*nouvelle syntaxe*" of [Reg92, DR95] (where the ?-links have any arity) in the style of (differential) interaction nets (see [Laf95, ER06b]).

Notation. Let $\mathbb{C} = (\mathsf{t}, \mathcal{P}, \mathsf{C}, \mathsf{P}^{\mathsf{pri}}, \mathsf{P}^{\mathsf{left}})$ be a module-base. We set $\mathsf{t}_{\mathbb{C}} = \mathsf{t}$, $\mathsf{C}_{\mathbb{C}} = \mathsf{C}$, $\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}} = \mathsf{P}^{\mathsf{pri}}$, $\mathsf{P}^{\mathsf{left}}_{\mathbb{C}} = \mathsf{P}^{\mathsf{left}}$. We recall the notations $\mathcal{P}(\mathbb{C}) = \mathcal{P}$ and $\mathcal{C}(\mathbb{C}) = \mathsf{dom}(\mathsf{t}_{\mathbb{C}})$.

For $\mathbb{C} \in$ **ModuleBases**, the function $\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}$ allows to distinguish the principal port from the auxiliary ones of any cell of $\mathbb{C}$. As expected, for the binary $\otimes$- and $⅋$-cells of $\mathbb{C}$, the function $\mathsf{P}^{\mathsf{left}}_{\mathbb{C}}$ allows to distinguish the left auxiliary port from the right one, whereas for the other kinds of cells a similar function is not defined because their auxiliary ports (if any) are not ordered.

Typically a cell l in a module-base $\mathbb{C}$ is graphically depicted as a triangle with its label $\mathsf{t}_{\mathbb{C}}(l)$ inside, the principal port being on a vertex and the auxiliary ones on the opposed side (in such a way that when the principal port is downwards the left auxiliary ports of a binary $\otimes$- or $⅋$-cell is placed on the left).

Remark 2. Let $\mathbb{C} \in$ **ModuleBases**.

The functions $\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}$ and $\mathsf{P}^{\mathsf{left}}_{\mathbb{C}}$ induce the functions:

- $\mathsf{P}^{\mathsf{aux}}_{\mathbb{C}} : \mathcal{C}(\mathbb{C}) \to \mathscr{P}(\mathcal{P}^{\mathsf{aux}}(\mathbb{C}))$ defined by $\mathsf{P}^{\mathsf{aux}}_{\mathbb{C}}(l) = \mathcal{P}^{\mathsf{aux}}_{l}$ for every $l \in \mathcal{C}(\mathbb{C})$; thus $\mathsf{im}(\mathsf{P}^{\mathsf{aux}}_{\mathbb{C}}) = \mathcal{P}^{\mathsf{aux}}(\mathbb{C})$;
- $\mathsf{P}^{\mathsf{right}}_{\mathbb{C}} : \mathcal{C}^{\otimes,⅋}_{2}(\mathbb{C}) \to \mathcal{P}^{\mathsf{aux}}(\mathbb{C})$ defined by $\{\mathsf{P}^{\mathsf{right}}_{\mathbb{C}}(l)\} = \mathcal{P}^{\mathsf{aux}}_{l} \smallsetminus \{\mathsf{P}^{\mathsf{left}}_{\mathbb{C}}(l)\}$ for every $l \in \mathcal{C}^{\otimes,⅋}_{2}(\mathbb{C})$; note that $\mathsf{P}^{\mathsf{right}}_{\mathbb{C}}$ is well-defined since the binary $\otimes$- and $⅋$-cells have exactly two auxiliary ports.

Notice that $\mathcal{P}(\mathbb{C}) = \mathcal{P}^{\mathsf{pri}}(\mathbb{C}) \uplus \mathcal{P}^{\mathsf{aux}}(\mathbb{C})$ and $\mathcal{P}_{l}(\mathbb{C}) = \mathcal{P}^{\mathsf{pri}}_{l}(\mathbb{C}) \uplus \mathcal{P}^{\mathsf{aux}}_{l}(\mathbb{C})$ for every $l \in \mathcal{C}(\mathbb{C})$.

Furthermore, if $\mathbb{C} \in$ **PseudoCells**, then $\mathsf{dom}(\mathsf{P}^{\mathsf{left}}_{\mathbb{C}}) = \mathcal{C}^{\otimes,⅋}(\mathbb{C}) = \mathsf{dom}(\mathsf{P}^{\mathsf{right}}_{\mathbb{C}})$.

Among the cell-bases, there is the *empty cell-base* $\mathbb{C}$ defined by $\mathcal{P}(\mathbb{C}) = \emptyset$ and $\mathsf{t}_{\mathbb{C}}$, $\mathsf{C}_{\mathbb{C}}$, $\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}$ and $\mathsf{P}^{\mathsf{left}}_{\mathbb{C}}$ are empty functions.

The following notion will be used in definitions 4 and 6

Definition 3 (Completeness)**.** *Let* $\mathbb{C} \in$ **ModuleBases** *and let* $Q \subseteq \mathcal{P}(\mathbb{C})$*: Q is* $\mathbb{C}$-complete *when, for every* $l \in \mathcal{C}(\mathbb{C})$*, if* $\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l) \in Q$*, then* $\mathcal{P}^{\mathsf{aux}}_{l}(\mathbb{C}) \subseteq Q$*.*

The following definitions 4, 6 and 8 formalize some intuitive notions of:

- erasure of some cells and ports in a module-base;
- submodule-base of a module-base;
- disjoint union of module-bases.

Definition 4 (Erasure of cells and ports)**.** *Let* $\mathbb{C} \in$ **ModuleBases** *and* $n \in \mathbb{N}$.

Let $l_1, \ldots, l_n \in \mathcal{C}(\mathbb{C})$*, let* $Q \subseteq \mathcal{P}(\mathbb{C})$ *be* $\mathbb{C}$*-complete and let* $L_Q = \{l \in \mathcal{C}(\mathbb{C}) \mid \mathcal{P}_l(\mathbb{C}) \subseteq Q\}$*. The* erasure of $l_1, \ldots, l_n$ and Q in $\mathbb{C}$ *is* $\mathbb{C}' = (\mathsf{t}', \mathcal{P}', \mathsf{C}', \mathsf{P}'^{\mathsf{pri}}, \mathsf{P}'^{\mathsf{left}})$ *where:*

- $\mathsf{t}' = \mathsf{t}_{\mathbb{C}}\restriction_{\mathcal{C}(\mathbb{C}) \setminus (L_Q \cup \{l_1, \ldots, l_n\})}$*;*
- $\mathcal{P}' = \mathcal{P}(\mathbb{C}) \setminus (Q \cup \bigcup_{i=1}^{n} \mathcal{P}_{l_i}(\mathbb{C}))$*;*
- $\mathsf{C}' = \mathsf{C}_{\mathbb{C}}\restriction_{\mathcal{P}'}$*;*
- $\mathsf{P}'^{\mathsf{pri}} = \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}\restriction_{\mathcal{C}(\mathbb{C}) \setminus (L_Q \cup \{l_1, \ldots, l_n\})}$*;*
- $\mathsf{P}'^{\mathsf{left}} = \mathsf{P}^{\mathsf{left}}_{\mathbb{C}}\restriction_{\mathcal{C}_2^{\otimes, ⅋}(\mathbb{C}) \setminus (L_Q \cup \{l_1, \ldots, l_n\})}$*.*

We say then that "$\mathbb{C}'$ is obtained from $\mathbb{C}$ by erasing $l_1, \ldots, l_n$ and Q*".*

Remark 5. For every $\mathbb{C} \in$ **ModuleBases** (resp. $\mathbb{C} \in$ **PseudoCells**; $\mathbb{C} \in$ **Cells**) and $l_1, \ldots, l_n \in \mathcal{C}(\mathbb{C})$, if $\mathbb{C}'$ is obtained from $\mathbb{C}$ by erasing $l_1, \ldots, l_n$, then $\mathbb{C}' \in$ **ModulesBases** (resp. $\mathbb{C}' \in$ **PseudoCells**; $\mathbb{C}' \in$ **Cells**); moreover, if $Q \subseteq \mathcal{P}(\mathbb{C})$ is $\mathbb{C}$-complete and $\mathbb{C}'$ is obtained from $\mathbb{C}$ by erasing $l_1, \ldots, l_n$ and Q, then $\mathbb{C}' \in$ **ModulesBases**.

Definition 6 (Submodule-base)**.** *Let* $\mathbb{C} \in$ **ModuleBases**. *Let* $Q \subseteq \mathcal{P}(\mathbb{C})$ *be* $\mathbb{C}$*-complete and such that, for every* $l \in \mathcal{C}(\mathbb{C})$ *and* $p \in Q$*, if* $p \in \mathcal{P}^{\mathsf{aux}}_l(\mathbb{C})$ *then* $\mathcal{P}_l(\mathbb{C}) \subseteq Q$*;*[1] *let* $L_Q = \{l \in \mathcal{C}(\mathbb{C}) \mid \mathcal{P}_l(\mathbb{C}) \subseteq Q\}$*. The* submodule-base of $\mathbb{C}$ generated by Q *is* $\mathsf{module}_{\mathbb{C}}(Q) = (\mathsf{t}', \mathcal{P}', \mathsf{C}', \mathsf{P}'^{\mathsf{pri}}, \mathsf{P}'^{\mathsf{left}})$ *where:*

- $\mathsf{t}' = \mathsf{t}_{\mathbb{C}}\restriction_{L_Q}$*;*
- $\mathcal{P}' = Q$*;*
- $\mathsf{C}' = \mathsf{C}_{\mathbb{C}}\restriction_{Q}$*;*
- $\mathsf{P}'^{\mathsf{pri}} = \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}\restriction_{L_Q}$*;*
- $\mathsf{P}'^{\mathsf{left}} = \mathsf{P}^{\mathsf{left}}_{\mathbb{C}}\restriction_{\mathcal{C}_2^{\otimes, ⅋}(\mathbb{C}) \cap L_Q}$*.*

Remark 7. Let $\mathbb{C} \in$ **ModuleBases** (resp. $\mathbb{C} \in$ **PseudoCells**; $\mathbb{C} \in$ **Cells**). Let $Q \subseteq \mathcal{P}(\mathbb{C})$ be $\mathbb{C}$-complete and such that, for every $l \in \mathcal{C}(\mathbb{C})$ and $p \in Q$, if $p \in \mathcal{P}^{\mathsf{aux}}_l(\mathbb{C})$ then $\mathcal{P}_l(\mathbb{C}) \subseteq Q$. Then $\mathsf{module}_{\mathbb{C}}(Q) \in$ **Modules** (resp. $\mathsf{module}_{\mathbb{C}}(Q) \in$ **PseudoCells**; $\mathsf{module}_{\mathbb{C}}(Q) \in$ **Cells**).

---

[1] According to definition 3, this entails that, for every $l \in \mathcal{C}(\mathbb{C})$, either $\mathcal{P}_l(\Phi) \cap Q = \emptyset$ or $\mathcal{P}_l(\Phi) \subseteq Q$.

Definition 8 (Disjoint union of module-bases)**.** *Let* $\mathbb{C}$ *and* $\mathbb{C}' \in$ **ModuleBases***:* $\mathbb{C}$ *and* $\mathbb{C}'$ *are* disjoint *if* $\mathcal{C}(\mathbb{C}) \cap \mathcal{C}(\mathbb{C}') = \emptyset$ *and* $\mathcal{P}(\mathbb{C}) \cap \mathcal{P}(\mathbb{C}') = \emptyset$*.*

Let $n \in \mathbb{N}$ *and let* $\mathbb{C}_1, \dots, \mathbb{C}_n \in$ **ModuleBases** *be pairwise disjoint: the* disjoint union of $\mathbb{C}_1, \dots, \mathbb{C}_n$ *is*

$$\biguplus_{i=1}^{n} \mathbb{C}_i = \big(\bigcup_{i=1}^{n} \mathsf{t}_{\mathbb{C}_i}, \bigcup_{i=1}^{n} \mathcal{P}(\mathbb{C}_i), \bigcup_{i=1}^{n} \mathsf{C}_{\mathbb{C}_i}, \bigcup_{i=1}^{n} \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}_i}, \bigcup_{i=1}^{n} \mathsf{P}^{\mathsf{left}}_{\mathbb{C}_i}\big)\,.$$

If $n = 2$*, the disjoint union of* $\mathbb{C}_1$ *and* $\mathbb{C}_2$ *is denoted by* $\mathbb{C}_1 \uplus \mathbb{C}_2$*.*

Remark 9. For every $n \in \mathbb{N}$, if $\mathbb{C}_1, \dots, \mathbb{C}_n \in$ **ModuleBases** (resp. $\mathbb{C}_1, \dots, \mathbb{C}_n \in$ **PseudoCells**; $\mathbb{C}_1, \dots, \mathbb{C}_n \in$ **Cells**) are pairwise disjoint and $\mathbb{C} = \biguplus_{i=1}^{n} \mathbb{C}_i$, then $\mathbb{C} \in$ **ModuleBases** (resp. $\mathbb{C} \in$ **PseudoCells**; $\mathbb{C} \in$ **Cells**).

The two following operations in a module-base of joining cells and separating a port from a cell formalize respectively the ideas of:

- "gluing" several cells of the same type in a unique new cell (of the same type) in such a way that the auxiliary ports of the new cell are all the auxiliary ports of the old cells (this notion will be used in definition 30);
- "separating" an auxiliary port p from its cell and "putting" p in a new cell of the same type, having p as its unique auxiliary port (this notion will be used in definition 32).

In a certain sense, the operations of separating a port from a cell l and joining l and the new cell created by the separation are "inverses" each other.

Definition 10 (Join of cells)**.** *Let* $\mathbb{C} \in$ **ModuleBases** *and let* $n \in \mathbb{N}^*$*.*

Let $l_1, \dots, l_n \in \mathcal{C}(\mathbb{C}) \smallsetminus \mathcal{C}^{\otimes,\parr}(\mathbb{C})$ *with* $\mathsf{t}_{\mathbb{C}}(l_i) = \mathsf{t}_{\mathbb{C}}(l_j)$ *for every* $1 \leq i \neq j \leq n$*, let* $l_0 \notin \mathcal{C}(\mathbb{C})$ *and let* $p_0 \notin \mathcal{P}(\mathbb{C})$*. The* join of $l_1, \dots, l_n$ in $\mathbb{C}$ *is* $\mathsf{join}_{\mathbb{C}}(l_1, \dots, l_n) = (\mathsf{t}', \mathcal{P}', \mathsf{C}', \mathsf{P}'^{\mathsf{pri}}, \mathsf{P}'^{\mathsf{left}})$ *where:*

- $\mathsf{t}' : (\mathcal{C}(\mathbb{C}) \smallsetminus \{l_1, \dots, l_n\}) \cup \{l_0\} \to \mathcal{T}$ *is the function defined by:*

$$\mathsf{t}'(l) = \begin{cases} \mathsf{t}_{\mathbb{C}}(l) & \text{if } l \in \mathcal{C}(\mathbb{C}) \smallsetminus \{l_1, \dots, l_n\} \\ \mathsf{t}_{\mathbb{C}}(l_1) & \text{if } l = l_0 \end{cases}$$

- $\mathcal{P}' = (\mathcal{P}(\mathbb{C}) \smallsetminus \{\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l_1), \dots, \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l_n)\}) \cup \{p_0\}$*;*
- $\mathsf{C}' : (\mathcal{P}(\mathbb{C}) \smallsetminus \{\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l_1), \dots, \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l_n)\}) \cup \{p_0\} \to \mathsf{dom}(\mathsf{t}')$ *is the function defined by:*

$$\mathsf{C}'(p) = \begin{cases} \mathsf{C}_{\mathbb{C}}(p) & \text{if } p \in \mathcal{P}(\mathbb{C}) \smallsetminus \big(\{\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l_1), \dots, \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l_n)\} \cup \bigcup_{i=1}^{n} \mathcal{P}^{\mathsf{aux}}_{l_i}(\mathbb{C})\big) \\ l_0 & \text{if } p \in \{p_0\} \cup \bigcup_{i=1}^{n} \mathcal{P}^{\mathsf{aux}}_{l_i}(\mathbb{C}) \end{cases}$$

- $\mathsf{P}'^{\mathsf{pri}} : \mathsf{dom}(\mathsf{t}') \to \mathcal{P}'$ *is the function defined by:*

$$\mathsf{P}'^{\mathsf{pri}}(l) = \begin{cases} \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l) & \textit{if } l \in \mathcal{C}(\mathbb{C}) \smallsetminus \{l_1, \ldots, l_n\} \\ p_0 & \textit{if } l = l_0 \end{cases}$$

- $\mathsf{P}'^{\mathsf{left}} = \mathsf{P}^{\mathsf{left}}_{\mathbb{C}}$.

We denote l_0 *by* $\mathsf{celljoin}_{\mathbb{C}}(l_1, \ldots, l_n)$.

Remark 11. For every $\mathbb{C} \in$ **ModuleBases** (resp. $\mathbb{C} \in$ **PseudoCells**; $\mathbb{C} \in$ **Cells**), $n \in \mathbb{N}^*$ and $l_1, \ldots, l_n \in \mathcal{C}(\mathbb{C})$ with $\mathsf{t}_{\mathbb{C}}(l_i) = \mathsf{t}_{\mathbb{C}}(l_j) \notin \{\otimes, \invamp\}$ for any $1 \leq i \neq j \leq n$, one has $\mathsf{join}_{\mathbb{C}}(l_1, \ldots, l_n) \in$ **ModulesBases** (resp. $\mathsf{join}_{\mathbb{C}}(l_1, \ldots, l_n) \in$ **PseudoCells**; $\mathsf{join}_{\mathbb{C}}(l_1, \ldots, l_n) \in$ **Cells**).

Definition 12 (Separation of a port)**.** *Let* $\mathbb{C} \in$ **ModulesBases***, let* $l_1 \in \mathcal{C}(\mathbb{C}) \smallsetminus \mathcal{C}^{\otimes,\invamp}(\mathbb{C})$ *and* $l_0 \notin \mathcal{C}(\mathbb{C})$*, let* $p_1 \in \mathcal{P}^{\mathsf{aux}}_{l_1}(\mathbb{C})$ *and* $p_0 \notin \mathcal{P}(\mathbb{C})$*. The* separation of p_1 in $\mathbb{C}$ *is* $\mathsf{sep}_R(p_1) = (\mathsf{t}', \mathcal{P}', \mathsf{C}', \mathsf{P}'^{\mathsf{pri}}, \mathsf{P}'^{\mathsf{left}})$ *where:*

- $\mathsf{t}' : \mathcal{C}(\mathbb{C}) \cup \{l_0\} \to \mathcal{T}$ *is the function defined by:*

$$\mathsf{t}'(l) = \begin{cases} \mathsf{t}_{\mathbb{C}}(l) & \textit{if } l \in \mathcal{C}(\mathbb{C}) \\ \mathsf{t}_{\mathbb{C}}(l_1) & \textit{if } l = l_0 \end{cases}$$

- $\mathcal{P}' = \mathcal{P}(\mathbb{C}) \cup \{p_0\}$*;*

- $\mathsf{C}' : \mathcal{P}(\mathbb{C}) \cup \{p_0\} \to \mathsf{dom}(\mathsf{t}')$ *is the function defined by:*

$$\mathsf{C}'(p) = \begin{cases} \mathsf{C}_{\mathbb{C}}(p) & \textit{if } p \in \mathcal{P}(\mathbb{C}) \smallsetminus \{p_1\} \\ l_0 & \textit{if } p \in \{p_0, p_1\} \end{cases}$$

- $\mathsf{P}'^{\mathsf{pri}} : \mathsf{dom}(\mathsf{t}') \to \mathcal{P}'$ *is the function defined by:*

$$\mathsf{P}'^{\mathsf{pri}}(l) = \begin{cases} \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l) & \textit{if } l \in \mathcal{C}(\mathbb{C}) \\ p_0 & \textit{if } l = l_0 \end{cases}$$

- $\mathsf{P}'^{\mathsf{left}} = \mathsf{P}^{\mathsf{left}}_{\mathbb{C}}$.

We denote l_0 *(resp.* l_1*) by* $\mathsf{newsep}_{\mathbb{C}}(p_1)$ *(resp.* $\mathsf{oldsep}_{\mathbb{C}}(p_1)$*).*

Remark 13. For every $\mathbb{C} \in$ **ModuleBases** (resp. $\mathbb{C} \in$ **PseudoCells**; $\mathbb{C} \in$ **Cells**), $l_1 \in \mathcal{C}(\mathbb{C}) \smallsetminus \mathcal{C}^{\otimes,\invamp}(\mathbb{C})$ and $l_0 \notin \mathcal{C}(\mathbb{C})$, let $p_1 \in \mathcal{P}^{\mathsf{aux}}_{l}(\mathbb{C})$ and $p_0 \notin \mathcal{P}(\mathbb{C})$, one has $\mathsf{join}_{\mathbb{C}}(l_1, \ldots, l_n) \in$ **ModulesBases** (resp. $\mathsf{join}_{\mathbb{C}}(l_1, \ldots, l_n) \in$ **PseudoCells**; $\mathsf{join}_{\mathbb{C}}(l_1, \ldots, l_n) \in$ **Cells**).

We introduce the notion of "identity" (or better said isomorphism) between two module-bases. The idea is that two module-bases are isomorphic iff they are identical up to the names of their cells and ports (in particular, they have the same graphical representation).

Definition 14 (Isomorphism on module-bases)**.** *Let* $\mathbb{C}, \mathbb{C}' \in \mathbf{ModuleBases}$.

An isomorphism from $\mathbb{C}$ to $\mathbb{C}'$ *is a pair* $\varphi = (\varphi_{\mathcal{C}}, \varphi_{\mathcal{P}})$ *of bijections* $\varphi_{\mathcal{C}} : \mathcal{C}(\mathbb{C}) \to \mathcal{C}(\mathbb{C}')$ *and* $\varphi_{\mathcal{P}} : \mathcal{P}(\mathbb{C}) \to \mathcal{P}(\mathbb{C}')$ *such that the following diagrams commute:*

$$\begin{tikzcd}
\mathcal{C}(\mathbb{C}) \arrow[r, "\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}"] \arrow[d, "\varphi_{\mathcal{C}}"'] & \mathcal{P}(\mathbb{C}) \arrow[r, "\mathsf{C}_{\mathbb{C}}"] \arrow[d, "\varphi_{\mathcal{P}}"'] & \mathcal{C}(\mathbb{C}) \arrow[r, "\mathsf{t}_{\mathbb{C}}"] \arrow[d, "\varphi_{\mathcal{C}}"'] & \mathcal{T} \\
\mathcal{C}(\mathbb{C}') \arrow[r, "\mathsf{P}^{\mathsf{pri}}_{\mathbb{C}'}"] & \mathcal{P}(\mathbb{C}') \arrow[r, "\mathsf{C}_{\mathbb{C}'}"] & \mathcal{C}(\mathbb{C}') \arrow[ur, "\mathsf{t}_{\mathbb{C}'}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{C}_2^{\otimes,\parr}(\mathbb{C}) \arrow[r, "\mathsf{P}^{\mathsf{left}}_{\mathbb{C}}"] \arrow[d, "\varphi_{\mathcal{C}}"'] & \mathcal{P}(\mathbb{C}) \arrow[d, "\varphi_{\mathcal{P}}"] \\
\mathcal{C}_2^{\otimes,\parr}(\mathbb{C}') \arrow[r, "\mathsf{P}^{\mathsf{left}}_{\mathbb{C}'}"] & \mathcal{P}(\mathbb{C}')
\end{tikzcd}$$

We write then $\varphi : \mathbb{C} \simeq \mathbb{C}'$*.*

If there exists an isomorphism from $\mathbb{C}$ *to* $\mathbb{C}$*, then we say that* $\mathbb{C}$ and $\mathbb{C}'$ are isomorphic *and we write* $\mathbb{C} \simeq \mathbb{C}'$*.*

Remark 15. Let $\mathbb{C}, \mathbb{C}' \in \mathbf{ModuleBases}$. If φ is an isomorphism from $\mathbb{C}$ to $\mathbb{C}'$ then:

1. $\mathsf{im}(\varphi_{\mathcal{P}} \restriction_{\mathcal{P}^{\mathsf{aux}}(\mathbb{C})}) = \mathcal{P}^{\mathsf{aux}}(\mathbb{C}')$ and $\mathsf{im}(\varphi_{\mathcal{P}} \restriction_{\mathcal{P}^{\mathsf{aux}}_l(\mathbb{C})}) = \mathcal{P}^{\mathsf{aux}}_{\varphi_{\mathcal{C}}(l)}(\mathbb{C}')$ (in particular, $\mathsf{a}_{\mathbb{C}}(l) = \mathsf{a}_{\mathbb{C}'}(\varphi_{\mathcal{C}}(l))$) for every $l \in \mathcal{C}(\mathbb{C})$;
2. if $\mathbb{C} \in \mathbf{PseudoCells}$ (resp. $\mathbb{C} \in \mathbf{Cells}$) then $\mathbb{C}' \in \mathbf{PseudoCells}$ (resp. $\mathbb{C}' \in \mathbf{Cells}$).

1.2 Pre-pre-proof-structures

A pre-pre-proof-structure (see also the analogous definition in [dCT12]) is morally a (hyper-)graph consisting of a cell-base, isolated ports (not belonging to any cell of the cell-base), wires connecting the ports of its cells and the isolated ones, and arrows to add some informations.

Definition 16 (Module, (pseudo-)pre-pre-proof-structure)**.** *A* module *is a 6-tuple* $\Phi = (\mathbb{C}, \mathcal{I}, \mathcal{D}, \mathcal{W}, \mathsf{auxd}, \mathsf{bc})$ *where:*

- $\mathbb{C} \in \mathbf{ModuleBases}$ *is the* module-base of Φ*; we set* $\mathcal{C}(\Phi) = \mathcal{C}(\mathbb{C})$ *whose elements are the* cells of Φ*;*
- $\mathcal{I}$ *and* $\mathcal{D}$ *are finite sets (whose elements are respectively the* isolated ports of Φ *and the* deadlocks of Φ*), satisfying* $\mathcal{I} \cap \mathcal{P}(\mathbb{C}) = \emptyset$*,* $\mathcal{D} \cap \mathcal{P}(\mathbb{C}) = \emptyset$ *and* $\mathcal{I} \cap \mathcal{D} = \emptyset$*; we set* $\mathcal{P}(\Phi) = \mathcal{P}(\mathbb{C}) \cup \mathcal{I} \cup \mathcal{D}$ *whose elements are the* ports of Φ*;* Φ is deadlock-free *if* $\mathcal{D}(\Phi) = \emptyset$*;*

- $\mathcal{W} \subseteq \mathscr{P}_2(\mathcal{P}(\Phi) \smallsetminus \mathcal{D})$ *such that:*

  1. *for every* $w, w' \in \mathcal{W}$*, if* $w \cap w' \neq \emptyset$ *then* $w = w'$*,*
  2. $\mathcal{P}^{\mathsf{aux}}(\mathbb{C}) \cup \mathcal{I} \subseteq \bigcup \mathcal{W}$*,*

  the elements of $\mathcal{W}$ *are the* wires of Φ*; we set* $\mathcal{C}uts(\Phi) = \{\{p,q\} \in \mathcal{W} \mid p, q \in \mathcal{P}^{\mathsf{pri}}(\mathbb{C})\}$ *whose elements are the* cuts of Φ*; any* $p \in \bigcup \mathcal{C}uts(\Phi)$ *is a* cut port of Φ*;* Φ *is* cut-free *if* $\mathcal{C}uts(\Phi) = \emptyset$*;*

- auxd *is a partial function from* $\mathcal{C}^!(\mathbb{C})$ *to* $\mathscr{P}(\mathcal{P}^{\mathsf{aux}}(\mathbb{C}))$ *such that for every* $l \in \mathcal{C}(\mathbb{C})$*, if* auxd *is defined in* l *then:*

  - $\mathsf{a}_{\mathbb{C}}(l) = 1$,
  - *if* $p \in \mathsf{auxd}(l)$*, then* $p \in \mathcal{P}^{\mathsf{aux}}_{l'}(\mathbb{C})$ *for some* $l' \in \mathcal{C}^?(\mathbb{C})$*; we say that* p *is an* auxiliary door of l*;*

  we set $\mathcal{C}^{\mathsf{prom}}(\Phi) = \mathsf{dom}(\mathsf{auxd})$ *(resp.* $\mathcal{A}uxdoors(\Phi) = \bigcup \mathsf{im}(\mathsf{auxd})$*) whose elements are the* promotions cells *(resp.* auxiliary doors*)* of Φ*; if* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ *and* p *is the premise of* l*, we set* $\mathsf{doors}_{\Phi}(l) = \{p\} \cup \mathsf{auxd}(l)$ *and we say that* p *is the* principal door of l in Φ *and any* $q \in \mathsf{auxd}(l)$ *is an* auxiliary door of l in Φ*;*

- bc *is a function from* $\mathcal{C}^{\mathsf{prom}}(\Phi)$ *to* $\mathscr{P}(\bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D})$ *such that:*

  - *if* $\mathsf{bc}(l) \cap \mathsf{bc}(l') \neq \emptyset$ *for some* $l, l' \in \mathcal{C}^{\mathsf{prom}}(\Phi)$*, then* $l = l'$*,*[2]
  - *if* $\{p,q\} \in \mathcal{C}uts(\Phi)$ *and* $p \in \mathsf{bc}(l) \cap \bigcup \mathcal{C}uts(\Phi)$ *for some* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi)$*, then* $q \in \mathsf{bc}(l)$*;*[3]

  we set $\mathsf{cutports}_{\Phi}(l) = \mathsf{bc}(l) \cap \bigcup \mathcal{C}uts(\Phi)$ *and* $\mathsf{deadlocks}_{\Phi}(l) = \mathsf{bc}(l) \cap \mathcal{D}$ *for every* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi)$*.*

A pseudo-structure *is a module* $\Phi = (\mathbb{C}, \mathcal{I}, \mathcal{D}, \mathcal{W}, \mathsf{auxd}, \mathsf{bc})$ *such that* $\mathbb{C}$ *is a pseudo-cell-base,* $\{\{p,p'\} \in \mathcal{W} \mid \exists\, l, l' \in \mathcal{C}^{ax}(\mathbb{C}) : p = \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l) \text{ and } p' = \mathsf{P}^{\mathsf{pri}}_{\mathbb{C}}(l')\} = \emptyset$[4] *and*

3. *for every* $w \in \mathcal{W}$*, if* $w \cap \mathcal{I} \neq \emptyset$ *then* $w \cap \mathcal{P}^{\mathsf{pri}}(\mathbb{C}) = \emptyset$*;*

we say then that $\mathbb{C}$ *is the* pseudo-cell-base of Φ.

A pre-pre-proof-structure *(or* ppps *for short) is a pseudo-structure* Φ *such that the pseudo-cell-base* $\mathbb{C}$ *of* Φ *is a cell-base; we say then that* $\mathbb{C}$ *is the* cell-base of Φ.

---

[2] This conditions means that for every cut or deadlock, there exists at most one !-cell pointing to it.

[3] This conditions means that, for every cut w of Φ, the function bc either points to both ports of w or does not point to any port of w. Therefore, we are entitled to talk about a cut associated with a promotion cell by the function bc.

[4] This means that in a pseudo-structure there is no cut connecting the principal ports of two ax-cells.

We denote by **Modules** *(resp.* **PseudoPPPS***;* **PPPS***) the set of modules (resp. pseudo-structures; pre-pre-proof-structures).*

In a module, an isolated port is depicted as a dot, a wire $\{p, q\}$ is graphically depicted as a line connecting the ports p and q, a deadlock is graphically depicted as a circle. If l is a promotion cell then its label is depicted as $\mathsf{!p}$, furthermore the fact that an auxiliary port q of a ?-cell is an auxiliary door of l is represented graphically by a dotted arrow from l to q; likewise, if q is a deadlock or cut port in $\mathsf{bc}(l)$, this is represented graphically by a dotted arrow from l to q or to the cut w such that $q \in w$.

A promotion cell of a ppps Φ has to be seen as a “candidate for a box”, i.e. a cell which is the starting point to attempt to compute the box (a particular sub-graph of Φ) associated with it (in general, it is not always possible, see definition 46).

Our definition of pre-pre-proof-structure differs from that one in [dCT12] by the following points:

- in our pre-pre-proof-structures, cuts (wires connecting the principal ports of two different cells) are allowed;
- in order to be closed under cut-elimination, in our definition of ppps we add the set $\mathcal{D}$ of deadlocks; a deadlock has to be seen as a sort of degenerate cut (morally, it is an axiom whose conclusions are connected by a cut, but our syntax cannot express that explicitly);
- in order to handle differential nets with our syntax, the !-cells’ arity does not need to be 1; furthermore, not all unary !-cells are “candidates for a box”;
- with respect to the definition in [dCT12], in our ppps we add the “arrow” functions auxd and bc which associate with every promotion cell l respectively the set of its auxiliary doors and the set of cut ports and deadlocks of depth[5] 0 in the “box-candidate” associated with l; a promotion cell l might have no auxiliary doors (resp. no cuts nor deadlocks) associated with it, this is the case when $\mathsf{auxd}(l) = \emptyset$ (resp. $\mathsf{bc}(l) = \emptyset$).

Definition 17 (Free port, axiom, arrow)**.** *Let* $\Phi = (\mathbb{C}, \mathcal{I}, \mathcal{D}, \mathcal{W}, \mathsf{auxd}, \mathsf{bc}) \in$ **Modules**. *We set:*

- $\mathbb{C}(\Phi) = \mathbb{C}$, $\mathcal{I}(\Phi) = \mathcal{I}$, $\mathcal{W}(\Phi) = \mathcal{W}$, $\mathcal{D}(\Phi) = \mathcal{D}$, $\mathsf{auxd}_\Phi = \mathsf{auxd}$, $\mathsf{bc}_\Phi = \mathsf{bc}$; $\mathcal{C}^t(\Phi) = \mathcal{C}^t(\mathbb{C}(\Phi))$ *and* $\mathcal{C}^{t,t'}(\Phi) = \mathcal{C}^{t,t'}(\mathbb{C}(\Phi))$ *for every* $t, t' \in \mathcal{T}$; $\mathcal{C}_2^{\otimes,\parr}(\Phi) = \mathcal{C}_2^{\otimes,\parr}(\mathbb{C}(\Phi))$; $\mathcal{P}^{\mathsf{pri}}(\Phi) = \mathcal{P}^{\mathsf{pri}}(\mathbb{C}(\Phi))$, $\mathcal{P}^{\mathsf{aux}}(\Phi) = \mathcal{P}^{\mathsf{aux}}(\mathbb{C}(\Phi))$; $\mathcal{P}_l^{\mathsf{aux}}(\Phi) = \mathcal{P}_l^{\mathsf{aux}}(\mathbb{C}(\Phi))$ *and* $\mathcal{P}_l(\Phi) = \mathcal{P}_l(\mathbb{C}(\Phi))$ *for every* $l \in \mathcal{C}(\Phi)$; $\mathsf{P}_\Phi^{\mathsf{pri}} = \mathsf{P}_{\mathbb{C}(\Phi)}^{\mathsf{pri}}$, $\mathsf{P}_\Phi^{\mathsf{left}} = \mathsf{P}_{\mathbb{C}(\Phi)}^{\mathsf{left}}$, $\mathsf{P}_\Phi^{\mathsf{aux}} = \mathsf{P}_{\mathbb{C}(\Phi)}^{\mathsf{aux}}$; $\mathsf{t}_\Phi = \mathsf{t}_{\mathbb{C}(\Phi)}$, $\mathsf{C}_\Phi = \mathsf{C}_{\mathbb{C}(\Phi)}$, $\mathsf{a}_\Phi = \mathsf{a}_{\mathbb{C}(\Phi)}$;

---

[5] See definition 59 and proposition 62 for the notion of depth.

- $\mathcal{P}^{\mathsf{free}}(\Phi) = \mathcal{I}(\Phi) \cup (\mathcal{P}^{\mathsf{pri}}(\Phi) \setminus \bigcup \mathcal{W}(\Phi))$ *whose elements are the* free ports *(or* conclusions*)* of Φ*;* $\mathcal{C}^{\mathsf{term}}(\Phi) = \{l \in \mathcal{C}(\Phi) \mid \mathsf{P}^{\mathsf{pri}}_{\Phi}(l) \in \mathcal{P}^{\mathsf{free}}(\Phi)\}$ *whose elements are the* terminal cells of Φ*;*

- $\mathcal{A}x(\Phi) = \{\{p, q\} \in \mathcal{W}(\Phi) \mid p, q \notin \mathcal{P}^{\mathsf{pri}}(\Phi)\}$ *whose elements are the* axioms of Φ*; any* $p \in \bigcup \mathcal{A}x(\Phi)$ *is an* axiom port of Φ*;* $\mathcal{A}x^{\mathsf{term}}(\Phi) = \{w \in \mathcal{A}x(\Phi) \mid \exists p \in w : p \in \mathcal{I}(\Phi)\}$ *(resp.* $\mathcal{A}x^{\mathsf{isol}}(\Phi) = \{w \in \mathcal{A}x(\Phi) \mid \forall p \in w : p \in \mathcal{I}(\Phi)\}$*) whose elements are the* terminal *(resp.* isolated*)* axioms of Φ*;*

- $\mathcal{A}rrows(\Phi) = \{\{p, q\} \in \mathscr{P}_2(\mathcal{P}(\Phi)) \mid \exists\, l \in \mathcal{C}^{\mathsf{prom}}(\Phi) : p \in \mathcal{P}^{\mathsf{aux}}_{l}(\Phi),\ q \in \mathsf{auxd}_{\Phi}(l) \cup \mathsf{bc}_{\Phi}(l)\}$*, whose elements are the* arrows of Φ*;*

- $\mathcal{C}uts_0(\Phi) = \mathcal{C}uts(\Phi) \setminus \mathscr{P}_2(\bigcup \mathsf{im}(\mathsf{bc}_{\Phi}))$ *(whose elements are the* cuts at depth 0 of Φ*) and* $\mathcal{D}_0(\Phi) = \mathcal{D}(\Phi) \setminus \bigcup \mathsf{im}(\mathsf{bc}_{\Phi})$ *(whose elements are the* deadlocks at depth 0 of Φ*) .*

For a module Φ, p is an isolated port of Φ when p is a port of some axiom and a conclusion of Φ. The meaning of the conditions on the set of wires in definition 16 is the following:

- condition 1 implies that three ports cannot be connected by two wires,

- condition 2 entails that auxiliary ports can never be conclusions of a ppps,

- condition 3 (only for pseudo-structures) implies that when the principal port of some cell is connected to another port this is necessarily a port of some cell, hence "hanging" wires (i.e. connecting a principal port and an isolated one) are not allowed in pseudo-structures.

Intuitively, a module Φ can be seen as:

- a finite undirected graph whose labeled nodes are the cells, deadlocks and free ports of Φ, and whose edges are the wires of Φ and the arrows connecting each promotion cell of Φ with its auxiliary doors, its cut ports and its deadlocks;

- a finite undirected hyper-graph whose nodes are the ports and deadlocks of Φ, whose labeled hyper-edges are the cells (connecting all its ports) of Φ, and whose edges are the wires of Φ and arrows connecting each promotion cell of Φ with its auxiliary doors, its cut ports and its deadlocks.

Remark 18. Let $\Phi \in \mathbf{Modules}$. $\mathcal{P}(\Phi) = \mathcal{P}^{\mathsf{pri}}(\Phi) \uplus \mathcal{P}^{\mathsf{aux}}(\Phi) \uplus \mathcal{I}(\Phi) \uplus \mathcal{D}(\Phi)$ and $\mathcal{P}(\Phi) \setminus \bigcup \mathcal{W}(\Phi) = \{\mathsf{P}^{\mathsf{pri}}_{\Phi}(l) \mid l \in \mathcal{C}^{\mathsf{term}}(\Phi)\} \uplus \mathcal{D}(\Phi)$, thus any port of Φ is a conclusion of Φ iff it is either the principal port of a terminal cell of Φ or an axiom port of Φ. In particular, $\mathcal{I}(\Phi) \subseteq \bigcup \mathcal{A}x(\Phi)$.

Among the ppps, the *empty ppps* Φ is defined by:

- $\mathbb{C}(\Phi)$ is the empty cell-base;
- $\mathcal{I}(\Phi) = \mathcal{D}(\Phi) = \mathcal{W}(\Phi) = \emptyset$;
- auxd_Φ and bc_Φ are the empty functions.

The following notion defines how to transform two different *ax*-cells of a pseudo-structure in an axiom: it will be used to associate with every point of $D^{<\omega}$ a DiLL_0-proof-structure (see definitions 43, 74 and 94).

Definition 19 (Connecting pairs of *ax*-cells)**.** *Let* $\Phi \in \mathbf{PseudoPPPS}$.

Let $l_1, l_2 \in \mathcal{C}^{ax}(\Phi)$ *with* $l_1 \neq l_2$. *We say that* Φ' is obtained from Φ by connecting l_1 and l_2 *if* $\Phi' = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where* $\mathcal{D}' = \mathcal{D}(\Phi)$, $\mathsf{auxd}' = \mathsf{auxd}_\Phi$, $\mathsf{bc}' = \mathsf{bc}_\Phi$, $\mathbb{C}'$ *is obtained by* $\mathbb{C}(\Phi)$ *by erasing* l_1 *and* l_2, *and furthermore:*

- *if* l_1 *and* l_2 *are not terminal cells of* Φ *and* p_1 *and* p_2 *are the auxiliary port of* Φ *such that* $\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_i), p_i\} \in \mathcal{W}(\Phi)$ *for* $i \in \{1,2\}$, *then* $\mathcal{W}' = (\mathcal{W}(\Phi) \smallsetminus \{\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_1), p_1\}, \{\mathsf{P}^{\mathsf{pri}}_\Phi(l_2), p_2\}\}) \cup \{\{p_1, p_2\}\}$ *and* $\mathcal{I}' = \mathcal{I}(\Phi)$*;*
- *if* l_1 *is a terminal cell of* Φ *and* l_2 *is not and* p_2 *is the auxiliary port of* Φ *such that* $\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_2), p_2\} \in \mathcal{W}(\Phi)$, *then* $\mathcal{I}' = \mathcal{I}(\Phi) \cup \{\mathsf{P}^{\mathsf{pri}}_\Phi(l_1)\}$ *and* $\mathcal{W}' = (\mathcal{W}(\Phi) \smallsetminus \{\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_2), p_2\}\}) \cup \{\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_1), p_2\}\}$*;*
- *if* l_2 *is a terminal cell of* Φ *and* l_1 *is not and* p_1 *is the auxiliary port of* Φ *such that* $\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_1), p_1\} \in \mathcal{W}(\Phi)$, *then* $\mathcal{I}' = \mathcal{I}(\Phi) \cup \{\mathsf{P}^{\mathsf{pri}}_\Phi(l_2)\}$ *and* $\mathcal{W}' = (\mathcal{W}(\Phi) \smallsetminus \{\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_1), p_1\}\}) \cup \{\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_2), p_1\}\}$*;*
- *if* l_1 *and* l_2 *are terminal cells of* Φ, *then* $\mathcal{I}' = \mathcal{I}(\Phi) \cup \{\mathsf{P}^{\mathsf{pri}}_\Phi(l_1), \mathsf{P}^{\mathsf{pri}}_\Phi(l_2)\}$ *and* $\mathcal{W}' = \mathcal{W}(\Phi) \cup \{\{\mathsf{P}^{\mathsf{pri}}_\Phi(l_2), \mathsf{P}^{\mathsf{pri}}_\Phi(l_1)\}\}$.

Let $n \in \mathbb{N}$ *and let* $l_1, l'_1, \dots, l_n, l'_n$ *be pairwise distinct ax-cells of* Φ. *We say that* Φ' is obtained from Φ by connecting $(l_1, l'_1), \dots, (l_n, l'_n)$ *when:*

- *if* $n = 0$ *then* $\Phi' = \Phi$*;*
- *if* $n > 0$ *then* Φ' *is obtained from* Φ'' *by connecting* l_n *and* l'_n, *where* Φ'' *is obtained from* Φ *by connecting* $(l_1, l'_1), \dots, (l_{n-1}, l'_{n-1})$.

Remark 20. For every $\Phi \in \mathbf{PseudoPPPS}$ and pairwise distinct *ax*-cells $l_1, l'_1, \dots, l_n, l'_n$ (for some $n \in \mathbb{N}$), if Φ' is obtained from Φ by connecting $(l_1, l'_1), \dots, (l_n, l'_n)$ then $\Phi' \in \mathbf{PseudoPPPS}$; moreover, if $\{l_1, l'_1, \dots, l_n, l'_n\} = \mathcal{C}^{ax}(\Phi)$, then $\Phi \in \mathbf{PPPS}$.

The following notion will be used in the sequel, for example in definition 105.

Definition 21 (Erasure of terminal cells, erasure of a cut at depth 0, erasure of hanging wires)**.** *Let* $\Phi \in \textbf{Modules}$ *and let* $n \in \mathbb{N}$.

Let $l_1, \ldots, l_n \in \mathcal{C}^{\mathsf{term}}(\Phi)$ *be such that, for every* $v \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ *and* $1 \leq i \leq n$*, one has* $\mathcal{P}^{\mathsf{aux}}_{l_i}(\Phi) \cap \mathsf{auxd}_\Phi(v) = \emptyset$*. The* erasure of $l_1, \ldots, l_n$ in Φ *is* $\Phi' = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where:*

- $\mathbb{C}'$ *is obtained from* $\mathbb{C}(\Phi)$ *by erasing* $l_1, \ldots, l_n$*;*
- $\mathcal{I}' = \mathcal{I}(\Phi) \cup \{p \in \bigcup_{i=1}^n \mathcal{P}^{\mathsf{aux}}_{l_i}(\Phi) \mid \exists\, q \in \mathcal{P}(\Phi) \setminus \mathcal{P}^{\mathsf{pri}}(\Phi) : \{p, q\} \in \mathcal{W}(\Phi)\}$*;*
- $\mathcal{D}' = \mathcal{D}(\Phi)$*;*
- $\mathcal{W}' = \{\{p, q\} \in \mathcal{W}(\Phi) \mid p \notin \mathcal{P}^{\mathsf{pri}}(\Phi)$ *or* $q \notin \bigcup_{i=1}^n \mathcal{P}^{\mathsf{aux}}_{l_i}(\Phi)\}$*;*
- $\mathsf{auxd}' = \mathsf{auxd}_\Phi \restriction_{\mathcal{C}^{\mathsf{prom}}(\Phi) \setminus \{l_1, \ldots, l_n\}}$*;*
- $\mathsf{bc}' = \mathsf{bc}_\Phi \restriction_{\mathcal{C}^{\mathsf{prom}}(\Phi) \setminus \{l_1, \ldots, l_n\}}$.

We say then that "Φ' is obtained from Φ by erasing $l_1, \ldots, l_n$*".*

Let $w_1, \ldots, w_n \in \mathcal{C}uts_0(\Phi)$*. The* erasure of $w_1, \ldots, w_n$ in Φ *is* $\Phi' = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where:*

- $\mathbb{C}' = \mathbb{C}(\Phi)$*,* $\mathcal{I}' = \mathcal{I}(\Phi)$ *and* $\mathcal{D}' = \mathcal{D}(\Phi)$*;*
- $\mathcal{W}' = \mathcal{W}(\Phi) \setminus \{w_1, \ldots, w_n\}$*;*
- $\mathsf{auxd}' = \mathsf{auxd}_\Phi$ *and* $\mathsf{bc}' = \mathsf{bc}_\Phi$.

We say then that "Φ' is obtained from Φ by erasing $w_1, \ldots, w_n$*".*

Let $H = \{p \in \mathcal{P}^{\mathsf{free}}(\Phi) \mid \exists\, q \in \mathcal{P}^{\mathsf{pri}}(\Phi) : \{p, q\} \in \mathcal{W}(\Phi)\}$*. The* erasure of the hanging wires in Φ *is* $\mathsf{nohang}(\Phi) = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where:*

- $\mathbb{C}' = \mathbb{C}(\Phi)$*;*
- $\mathcal{I}' = \mathcal{I}(\Phi) \setminus H$*;*
- $\mathcal{D}' = \mathcal{D}(\Phi)$*;*
- $\mathcal{W}' = \mathcal{W}(\Phi) \setminus \{w \in \mathcal{W}(\Phi) \mid \exists\, p \in w \cap H\}$*;*
- $\mathsf{auxd}' = \mathsf{auxd}_\Phi$ *and* $\mathsf{bc}' = \mathsf{bc}_\Phi$.

Let T *be a set such that* $T \cap (\mathcal{P}^{\mathsf{free}}(\Phi) \cap \mathcal{P}^{\mathsf{pri}}(\Phi)) = \emptyset$ *and* $\mathsf{p} : \mathcal{P}^{\mathsf{free}}(\Phi) \cap \mathcal{P}^{\mathsf{pri}}(\Phi) \to T$ *be a bijection. The* add of the hanging wires in Φ *is* $\mathsf{hang}(\Phi) = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where:*

- $\mathbb{C}' = \mathbb{C}(\Phi)$*;*
- $\mathcal{I}' = \mathcal{I}(\Phi) \cup T$*;*
- $\mathcal{D}' = \mathcal{D}(\Phi)$*;*

- $\mathcal{W}' = \mathcal{W}(\Phi) \cup \{\{q, \mathsf{p}(q)\} \mid q \in \mathcal{P}^{\mathsf{free}}(\Phi) \cap \mathcal{P}^{\mathsf{pri}}(\Phi)\}$;
- $\mathsf{auxd}' = \mathsf{auxd}_\Phi$ *and* $\mathsf{bc}' = \mathsf{bc}_\Phi$.

Remark 22. Let $\Phi \in$ **Modules** (resp. $\Phi \in$ **PseudoPPPS**; $\Phi \in$ **PPPS**) and let $n \in \mathbb{N}$.

Let $l_1, \ldots, l_n \in \mathcal{C}^{\mathsf{term}}(\Phi)$ be such that, for every $v \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ and $1 \leq i \leq n$, one has $\mathcal{P}^{\mathsf{aux}}_{l_i}(\Phi) \cap \mathsf{auxd}_\Phi(v) = \emptyset$. If Φ' is the erasure of $l_1, \ldots, l_n$ in Φ then $\Phi' \in$ **Modules** (resp. $\Phi' \in$ **PseudoPPPS**; $\Phi' \in$ **PPPS**).

Let $w_1, \ldots, w_n \in \mathcal{C}uts_0(\Phi)$. If Φ' is the erasure of $w_1, \ldots, w_n$ in Φ then $\Phi' \in$ **Modules** (resp. $\Phi' \in$ **PseudoPPPS**; $\Phi' \in$ **PPPS**).

One has $\mathsf{nohang}(\Phi), \mathsf{hang}(\Phi) \in$ **Modules**. Furthermore, if $\Phi \in$ **PseduoPPPS** then $\mathsf{nohang}(\Phi) = \Phi$.

Roughly speaking, the erasure of a terminal cell l in a module Φ is the module obtained from Φ by erasing l, its principal port, any hanging wire created by this erasure and the auxiliary ports of l which are not axiom ports in Φ. This operation might create new isolated ports: the auxiliary ports of l which are axiom ports of Φ. The request that no auxiliary port of l is pointed by an arrow of any promotion cell of Φ is mandatory to make sure that the erasure of l in Φ is a module.

The following notions will be used in definitions 24 and 26.

Definition 23 (Completeness and erasability of a set of ports)**.** *Let* $\Phi \in$ **Modules***.*

Let $Q \subseteq \mathcal{P}(\Phi)$*:* Q *is* Φ-complete *(resp.* Φ-erasable*) if* Q *is* $\mathbb{C}(\Phi)$*-complete and such that, for every* $\{p, q\} \in \mathcal{W}(\Phi)$*, if* $p \in Q \setminus \mathcal{P}^{\mathsf{pri}}(\Phi)$ *(resp. if* $p \in Q$*) then* $q \in Q$*.*

The following definitions 24, 26 and 28 formalize some intuitive notions of:

- erasure of some ports, cells and wires in a module;
- submodule of a module;
- disjoint union of modules (it will be used in definition 94).

They are generalizations for modules of the corresponding operations seen in definitions 4, 6 and 8 for module-bases.

Definition 24 (Erasure of ports, cells and wires)**.** *Let* $\Phi \in$ **Modules** *and let* $Q \subseteq \mathcal{P}(\Phi)$ *be* Φ*-erasable.*

The erasure of Q in Φ *is* $\Phi' = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where:*

- $\mathbb{C}'$ *is the erasure of* Q *in* Φ*;*
- $\mathcal{I}' = \mathcal{I}(\Phi) \setminus Q$*;*

- $\mathcal{D}' = \mathcal{D}(\Phi) \smallsetminus Q$;
- $\mathcal{W}' = \{w \in \mathcal{W}(\Phi) \mid \forall p \in w : p \notin Q\}$;[6]
- $\mathsf{auxd}' : (\mathcal{C}^{\mathsf{prom}}(\Phi) \smallsetminus L_Q) \to \mathscr{P}(\mathcal{P}^{\mathsf{aux}}(\Phi))$ *is a function such that, for every* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi) \smallsetminus L_Q$*, one has* $\mathsf{auxd}'(l) = \mathsf{auxd}_\Phi(l) \smallsetminus Q$;
- $\mathsf{bc}' : (\mathcal{C}^{\mathsf{prom}}(\Phi) \smallsetminus L_Q) \to \mathscr{P}(\bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D})$ *is a function such that, for every* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi) \smallsetminus L_Q$*, one has* $\mathsf{bc}'(l) = \mathsf{bc}_\Phi(l) \smallsetminus Q$.

We say then that "Φ' is obtained from Φ by erasing Q*".*

Remark 25. Let $\Phi \in \mathbf{Modules}$. If $Q \subseteq \mathcal{P}(\Phi)$ is Φ-erasable and Φ' is the erasure of Q in Φ, then $\Phi' \in \mathbf{Modules}$.

Definition 26 (Submodule)**.** *Let* $\Phi \in \mathbf{Modules}$.

Let $Q \subseteq \mathcal{P}(\Phi)$ *be* Φ*-complete, let* $L_Q = \{l \in \mathcal{C}(\Phi) \mid \mathcal{P}_l(\Phi) \subseteq Q\}$ *and let* $Q' = \{p \in Q \mid \exists\, l \in L_Q : p \in \mathcal{P}_l(\Phi)\}$*. The* submodule of Φ generated by Q *is* $\mathsf{module}_\Phi(Q) = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where:*

- $\mathbb{C}' = \mathsf{module}_{\mathbb{C}(\Phi)}(Q')$;
- $\mathcal{I}' = Q \smallsetminus (Q' \cup \mathcal{D}(\Phi))$;
- $\mathcal{D}' = \mathcal{D}(\Phi) \cap Q$;
- $\mathcal{W}' = \{w \in \mathcal{W}(\Phi) \mid \forall p \in w : p \in Q\}$;
- $\mathsf{auxd}' : (\mathcal{C}^{\mathsf{prom}}(\Phi) \cap L_Q) \to \mathscr{P}(\mathcal{P}^{\mathsf{aux}}(\Phi))$ *is a function such that, for every* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi) \cap L_Q$*, one has* $\mathsf{auxd}'(l) = \mathsf{auxd}_\Phi(l) \cap Q$;
- $\mathsf{bc}' : (\mathcal{C}^{\mathsf{prom}}(\Phi) \cap L_Q) \to \mathscr{P}(\bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D})$ *is a function such that, for every* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi) \cap L_Q$*, one has* $\mathsf{bc}'(l) = \mathsf{bc}_\Phi(l) \cap Q$.

With reference to notation used in definition 26, Q' is $\mathbb{C}(\Phi)$-complete and such that, for every $l \in \mathcal{C}(\Phi)$ and $p \in Q$, if $p \in \mathcal{P}^{\mathsf{aux}}_l(\Phi)$ then $\mathcal{P}_l(\Phi) \subseteq Q$, therefore $\mathsf{module}_{\mathbb{C}(\Phi)}(Q')$ is well-defined.

Remark 27. If $\Phi \in \mathbf{Modules}$ and $Q \subseteq \mathcal{P}(\Phi)$ is Φ-complete, then $\mathsf{module}_\Phi(Q) \in \mathbf{Modules}$. Furthermore, if $\Phi \in \mathbf{PseduoPPPS}$ (resp. $\Phi \in \mathbf{PPPS}$) then $\mathsf{nohang}(\mathsf{module}_\Phi(Q)) \in \mathbf{PseduoPPPS}$ (resp. $\mathsf{nohang}(\mathsf{module}_\Phi(Q)) \in \mathbf{PPPS}$).

Definition 28 (Disjoint union of modules)**.** *Let* $\Phi, \Phi' \in \mathbf{Modules}$*:* Φ *and* Φ' *are* disjoint *if* $\mathbb{C}(\Phi)$ *and* $\mathbb{C}(\Phi')$ *are disjoint,* $\mathcal{I}(\Phi) \cap \mathcal{I}(\Phi') = \emptyset$ *and* $\mathcal{D}(\Phi) \cap \mathcal{D}(\Phi') = \emptyset$.[7]

---

[6] According to definition 23, this is equivalent to $\mathcal{W}' = \{w \in \mathcal{W}(\Phi) \mid \exists\, p \in w : p \notin Q\}$.

[7] This implies that $\mathcal{W}(\Phi) \cap \mathcal{W}(\Phi') = \emptyset$ and $\mathcal{C}^{\mathsf{prom}}(\Phi) \cap \mathcal{C}^{\mathsf{prom}}(\Phi') = \emptyset$.

Let $n \in \mathbb{N}$ *and let* $\Phi_1, \ldots, \Phi_n \in$ **Modules** *be pairwise disjoint. The* disjoint union of $\Phi_1, \ldots, \Phi_n$ *is*

$$\biguplus_{i=1}^{n} \Phi_i = \Big(\biguplus_{i=1}^{n} \mathbb{C}(\Phi_i), \bigcup_{i=1}^{n} \mathcal{I}(\Phi_i), \bigcup_{i=1}^{n} \mathcal{D}(\Phi_i), \bigcup_{i=1}^{n} \mathcal{W}(\Phi_i), \bigcup_{i=1}^{n} \mathsf{auxd}_{\Phi_i}, \bigcup_{i=1}^{n} \mathsf{bc}_{\Phi_i} \Big) .$$

If $n = 2$*, the disjoint union of* Φ_1 *and* Φ_2 *is denoted by* $\Phi_1 \uplus \Phi_2$*.*

Remark 29. For every $n \in \mathbb{N}$, if $\Phi_1, \ldots, \Phi_n \in$ **Modules** (resp. $\Phi_1, \ldots, \Phi_n \in$ **PseudoPPPS**; $\Phi_1, \ldots, \Phi_n \in$ **PPPS**) are pairwise disjoint and $\Phi = \biguplus_{i=1}^{n} \Phi_i$, then $\Phi \in$ **Modules** (resp. $\Phi \in$ **PseudoPPPS**; $\Phi \in$ **PPPS**).

The two following operations on a module of joining terminal cells and separating a port from a terminal cell formalize respectively the ideas of:

- "gluing" several terminal cells of the same type in a unique new terminal cell (of the same type) in such a way that the auxiliary ports of the new cell are all the auxiliary ports of the old cells;
- "separating" an auxiliary port p from its terminal cell and "putting" p in a new terminal cell of the same type, having p as its unique auxiliary port.

They are generalizations for modules of the corresponding operations seen in definitions 10 and 12 for module-bases.

Definition 30 (Join of terminal cells)**.** *Let* $\Phi \in$ **Modules***, let* $n \in \mathbb{N}^*$*, let* $l_1, \ldots, l_n \in \mathcal{C}^{\mathsf{term}}(R) \smallsetminus \mathcal{C}^{\otimes,\parr}(\Phi)$ *be such that* $\mathsf{t}_\Phi(l_i) = \mathsf{t}_\Phi(l_j)$ *for every* $1 \leq i \neq j \leq n$*. The* join of $l_1, \ldots, l_n$ in Φ *is* $\mathsf{join}_\Phi(l_1, \ldots, l_n) = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where:*

- $\mathbb{C}' = \mathsf{join}_{\mathbb{C}}(l_1, \ldots, l_n)$*;*
- $\mathcal{I}' = \mathcal{I}(\Phi)$ *and* $\mathcal{D}' = \mathcal{D}(\Phi)$*;*
- $\mathcal{W}' = \mathcal{W}(\Phi)$*;*
- $\mathsf{auxd}' = \mathsf{auxd}_\Phi$ *and* $\mathsf{bc}' = \mathsf{bc}_\Phi$*.*

Remark 31. Let $\Phi \in$ **Modules** (resp. $\Phi \in$ **PPPS**). For every $n \in \mathbb{N}^*$ and $l_1, \ldots, l_n \in \mathcal{C}^{\mathsf{term}}(R) \smallsetminus \mathcal{C}^{\otimes,\parr}(\Phi)$ such that $\mathsf{t}_\Phi(l_i) = \mathsf{t}_\Phi(l_j)$ for any $1 \leq i \neq j \leq n$, one has $\mathsf{join}_\Phi(p) \in$ **Modules** (resp. $\mathsf{join}_\Phi(p) \in$ **PPPS**).

Definition 32 (Separation of a port from a terminal cell)**.** *Let* $\Phi \in$ **Modules***, let* $l \in \mathcal{C}^{\mathsf{term}}(R) \smallsetminus \mathcal{C}^{\otimes,\parr}(\Phi)$ *and* $p \in \mathcal{P}^{\mathsf{aux}}_l(\Phi)$*. The* separation of p in Φ *is* $\mathsf{sep}_\Phi(p) = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *where:*

- $\mathbb{C}' = \mathsf{sep}_{\mathbb{C}(\Phi)}(p)$*;*

- $\mathcal{I}' = \mathcal{I}(\Phi)$ *and* $\mathcal{D}' = \mathcal{D}(\Phi)$*;*
- $\mathcal{W}' = \mathcal{W}(\Phi)$*;*
- $\mathsf{auxd}' = \mathsf{auxd}_\Phi$ *and* $\mathsf{bc}' = \mathsf{bc}_\Phi$.

We set $\mathsf{newsep}_\Phi(p) = \mathsf{newsep}_{\mathbb{C}(\Phi)}(p)$ *and* $\mathsf{oldsep}_\Phi(p) = \mathsf{oldsep}_{\mathbb{C}(\Phi)}(p)$*.*

Remark 33. Let $\Phi \in \mathbf{Modules}$ (resp. $\Phi \in \mathbf{PPPS}$). For every $l \in \mathcal{C}^{\mathsf{term}}(R) \setminus \mathcal{C}^{\otimes,\parr}(\Phi)$ and $p \in \mathcal{P}^{\mathsf{aux}}_l(\Phi)$, one has $\mathsf{sep}_\Phi(p) \in \mathbf{Modules}$ (resp. $\mathsf{sep}_\Phi(p) \in \mathbf{PPPS}$).

We introduce the notion of "identity" (or better said isomorphism) between two modules. The idea is that two modules are isomorphic iff they are identical up to the names of their cells and ports (in particular, they have the same graphical representation).

Definition 34 (Isomorphism on modules)**.** *Let* $\Phi, \Phi' \in \mathbf{PPPS}$*.*

An isomorphism from Φ to Φ' *is a pair* $\varphi = (\varphi_{\mathcal{C}}, \varphi_{\mathcal{P}})$ *such that:*

- $\varphi_{\mathcal{P}} : \mathcal{P}(\Phi) \to \mathcal{P}(\Phi')$ *is a bijection where* $\mathsf{im}(\varphi_{\mathcal{P}} \upharpoonright_{\mathcal{I}(\Phi)}) = \mathcal{I}(\Phi')$ *and* $\mathsf{im}(\varphi_{\mathcal{P}} \upharpoonright_{\mathcal{D}(\Phi)}) = \mathcal{D}(\Phi')$*;*
- $(\varphi_{\mathcal{C}}, \varphi_{\mathcal{P}} \upharpoonright_{\mathcal{P}(\mathbb{C}(\Phi))}) : \mathbb{C}(\Phi) \simeq \mathbb{C}(\Phi')$*;*
- *for every* $\{p, q\} \in \mathscr{P}_2(\mathcal{P}(\Phi))$*, we have* $\{p, q\} \in \mathcal{W}(\Phi)$ *iff* $\{\varphi_{\mathcal{P}}(p), \varphi_{\mathcal{P}}(q)\} \in \mathcal{W}(\Phi')$*;*
- $\mathsf{im}(\varphi_{\mathcal{C}} \upharpoonright_{\mathcal{C}^{\mathsf{prom}}(\Phi)}) = \mathcal{C}^{\mathsf{prom}}(\Phi')$*;*
- *the following diagrams commute:*

$$\begin{tikzcd}
\mathcal{C}^{\mathsf{prom}}(\Phi) \arrow[r, "\mathsf{auxd}_\Phi"] \arrow[d, "\varphi_{\mathcal{C}}"'] & \mathscr{P}(\mathcal{A}uxdoors(\Phi)) \arrow[d, "\mathscr{P}(\varphi_{\mathcal{P}})"] \\
\mathcal{C}^{\mathsf{prom}}(\Phi') \arrow[r, "\mathsf{auxd}_{\Phi'}"] & \mathscr{P}(\mathcal{A}uxdoors(\Phi'))
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{C}^{\mathsf{prom}}(\Phi) \arrow[r, "\mathsf{bc}_\Phi"] \arrow[d, "\varphi_{\mathcal{C}}"'] & \mathsf{im}(\mathsf{bc}_\Phi) \arrow[d, "\mathscr{P}(\varphi_{\mathcal{P}})"] \\
\mathcal{C}^{\mathsf{prom}}(\Phi') \arrow[r, "\mathsf{bc}_{\Phi'}"] & \mathsf{im}(\mathsf{bc}_{\Phi'})
\end{tikzcd}$$

We write then $\varphi : \Phi \simeq \Phi'$*.*

If there exists an isomorphism from Φ *to* Φ'*, then we say that* Φ and Φ' are isomorphic *and we write* $\Phi \simeq \Phi'$*.*

Remark 35. Let $\Phi, \Phi' \in \mathbf{Modules}$. If φ is an isomorphism from Φ to Φ' then:

1. $\mathsf{im}(\varphi_{\mathcal{P}} \upharpoonright_{\mathcal{P}^{\mathsf{free}}(\Phi)} = \mathcal{P}^{\mathsf{free}}(\Phi'))$;
2. if $\Phi \in \mathbf{PseudoPPPS}$ (resp. $\Phi \in \mathbf{PPPS}$) then $\Phi' \in \mathbf{PseduoPPPS}$ (resp. $\Phi' \in \mathbf{PPPS}$).

1.3 Paths

In this section we introduce some usual notions of graph theory, merely adapted to our syntax. The only originality in our definition is that we consider the arrows (i.e. the pairs of ports connected by functions auxd_Φ or bc_Φ) as edges in our undirected (hyper-)graphs.

Definition 36 (Path, connection, acyclicity)**.** *Let* $\Phi \in \mathbf{Modules}$.

A path in Φ *is a sequence* $(p_i)_{i \in I}$ *of ports of* Φ *where* I *is an initial segment of* $\mathbb{N}$ *and such that, for every* $i, i+1 \in I$*:*

- $p_i \neq p_{i+1}$*;*
- *one of the following conditions holds*
  - *either* $\{p_i, p_{i+1}\} \in \mathcal{W}(\Phi) \cup \mathcal{A}rrows(\Phi)$*,*
  - *or* p_i *and* p_{i+1} *are ports of a same cell of* Φ*,*
- *if* $i+2 \in I$ *and* $p_i = p_{i+2}$ *then* p_i *and* p_{i+1} *are ports of a same cell of* Φ *and* $\{p_i, p_{i+1}\} \in \mathcal{W}(\Phi)$*.*[8]

Let $\varphi = (p_i)_{i \in I}$ *be a path in* Φ*. For every* $i \in I$*,* φ crosses p_i*, moreover if* $i \neq 0$ *and* $i+1 \in I$ *then* φ crosses internally p_i*. For every* $c \in \mathcal{W}(\Phi) \cup \mathcal{A}rrows(\Phi)$ *(resp.* $c \in \mathcal{C}(\Phi)$*),* φ crosses c *if there exist* $i, i+1 \in I$ *such that* $\{p_i, p_{i+1}\} = c$ *(resp.* $p_i, p_{i+1} \in \mathcal{P}_c(\Phi)$*). If* $I = \emptyset$ *then* φ *is the* empty path*. If* $I \neq \emptyset$ *then* p_0 *is the* start port of φ *(or* φ starts from p_0*). If* $I = \{0, \dots, n\}$ *for some* $n \in \mathbb{N}$*, then the path* φ *is said* finite *and* from p_0 to p_n *(or* connecting p_0 and p_n*), furthermore* n *is the* length of φ *(denoted by* $\mathsf{length}(\varphi)$*) and* p_n *is the* end port of φ *(or* φ ends in p_n*). If* $I = \mathbb{N}$*, then the path* φ *is said* infinite*. The* terminal ports of φ *are the start and end (if any) ports of* φ*.*

A cycle *in* Φ *is a finite path* $(p_i)_{0 \leq i \leq n}$ *in* Φ *such that* $p_0 = p_n$ *and* $n \neq 0$*.*

Φ *is* acyclic *if there is no deadlock nor cycle in* Φ*.*

Let p, q *be ports of* ρ*:* p *and* q *are* connected *if there exists a path in* Φ *from* p *to* q*.*

Φ *is* connected *if all ports* p, q *of* Φ *are connected.*

Remark 37. Let $\Phi \in \mathbf{Modules}$. If $p \in \mathcal{I}(\Phi) \cup \mathcal{D}(\Phi)$ or $p \in \mathcal{P}_l(\Phi)$ for some 0-ary cell l (and so $p = \mathsf{P}^{\mathsf{pri}}_\Phi(l)$) and if φ is a path in Φ crossing p, then p is a terminal port of φ.

Notice that a finite path on a module is not empty and it might have length 0 (i.e. it consists of only one port), whereas an empty path has no length. A path on a module can crosses arrows.

Notation. If a path in $\Phi \in \mathbf{Modules}$ is finite, it is often denoted by a finite sequence $(p_i)_{i \in I}$ of ports of Φ where I is not an initial segment of $\mathbb{N}$ but only a finite set.

---

[8] This condition is imposed to facilitate the definition of cycle.

The following notions will be used to compute the box associated with a promotion cell (definitions 46 and 48) in a ppps.

Definition 38 (Ascending path, path above a promotion cell, box-crossing path)**.** *Let* $\Phi \in \mathbf{PPPS}$.

A path $(p_i)_{i\in I}$ *in* Φ *(where* I *is an initial segment of* $\mathbb{N}$*) is* ascending *if, for every* $i, i+1 \in I$*, one of the following conditions holds:*

- *if* $p_i = \mathsf{P}^{\mathsf{pri}}_{\Phi}(l)$ *for some* $l \in \mathcal{C}(\Phi)$ *then* $p_{i+1} \in \mathcal{P}^{\mathsf{aux}}_{l}(\Phi)$*;*
- *if* $p_i \in \mathcal{P}^{\mathsf{aux}}(\Phi)$ *then:*
  - *either* $p_{i+1} \in \mathcal{P}^{\mathsf{pri}}(\Phi)$ *with* $\{p_i, p_{i+1}\} \in \mathcal{W}(\Phi)$*,*
  - *or* $p_{i+1} \in \mathsf{bc}_{\Phi}(l) \cup \mathsf{auxd}_{\Phi}(l)$ *for some* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ *such that* $p_i \in \mathcal{P}^{\mathsf{aux}}_{l}(\Phi)$*.*

We define a binary relation $\preccurlyeq_{\Phi}$ *on* $\mathcal{P}(\Phi)$ *by:* $p \preccurlyeq_{\Phi} q$ *if there exists an ascending path in* Φ *from* p *to* q*. For every* $n \in \mathbb{N}$*, we write* $p \preccurlyeq^{n}_{\Phi} q$ *if there exists an ascending path of length* n *from* p *to* q*.*

Let $l \in \mathcal{C}^{\mathsf{prom}}(\Phi)$*, let* p_l *be the (unique) auxiliary port of* l*. A* path above[9] l in Φ *is an ascending path in* Φ *starting from* p_l*. We set* $\mathsf{cdabove}_{\Phi}(l) = \{q \in \bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D}(\Phi) \mid \exists$ *path above* l *ending in* $q\}$*.*

For every $l, l' \in \mathcal{C}(\Phi)$*, we say that* l' is $\preccurlyeq$-above[9] l *if there exists an ascending path from* $\mathsf{P}^{\mathsf{pri}}_{\Phi}(l)$ *to* $\mathsf{P}^{\mathsf{pri}}_{\Phi}(l')$*.*

A path $(p_i)_{i\in I}$ *in* Φ *(where* I *is an initial segment of* $\mathbb{N}$*) is* box-crossing *if it is ascending and, for every* $i+1 \in I$*, if* $p_{i+1} \in \mathcal{A}uxdoors(\Phi)$ *then* $p_i = \mathsf{P}^{\mathsf{pri}}_{\Phi}(l)$ *for the* $l \in \mathcal{C}^{?}(\Phi)$ *such that* $p_{i+1} \in \mathcal{P}^{\mathsf{aux}}_{l}(\Phi)$*.*

We define a binary relation $\preceq_{\Phi}$ *on* $\mathcal{P}(\Phi)$ *by:* $p \preceq_{\Phi} q$ *if there exists a box-crossing path from* p *to* q*. For every* $n \in \mathbb{N}$*, we write* $p \preceq^{n}_{\Phi} q$ *if there exists a box-crossing path of length* n *from* p *to* q*.*

Roughly speaking, a box-crossing path in a ppps is an ascending path that cannot cross an arrow from a promotion cell to one of the auxiliary doors associated with it (but it can cross an arrow from a promotion cell to a cut port or a deadlock associated with it).

Remark 39. Let $\Phi \in \mathbf{PPPS}$.

1. Clearly, $\preceq^{1}_{\Phi} \subseteq \preccurlyeq^{1}_{\Phi}$ and $\preceq_{\Phi} \subseteq \preccurlyeq_{\Phi}$, furthermore $\preccurlyeq_{\Phi} = \bigcup_{n\in\mathbb{N}} \preccurlyeq^{n}_{\Phi}$ and $\preceq_{\Phi} = \bigcup_{n\in\mathbb{N}} \preceq^{n}_{\Phi}$.
2. An ascending path in Φ starting from or ending in $p \in \mathcal{I}(\Phi) \cup \mathcal{D}_0(\Phi)$ is necessarily of length 0. More generally, if $\varphi = (p_i)_{i\in I}$ is an ascending path in Φ and $p_j \in \bigcup \mathcal{A}x(\Phi) \cup \mathcal{D}(\Phi)$ for some $j \in I$ then $I = \{0, \ldots, j\}$ i.e. p_j is the end port of φ.

---

[9] The use of the term “above” will be justified in the case of proof-structures by proposition 55.1

3. $\preceq_\Phi, \preccurlyeq_\Phi \subseteq \mathcal{P}(\Phi)^2$ are pre-order relations, but in general they are not order relations because they are not antisymmetric. Some examples of a ppps Φ such that $\preceq_\Phi$ or $\preccurlyeq_\Phi$ is not antisymmetric are given in remarks 41.2-3.

4. An ascending path $(p_i)_{i\in I}$ in Φ is necessarily such that if $i+1 \in I$ and $p_{i+1} \in \mathsf{bc}_\Phi(l)$ for some $l \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ then p_i is the unique auxiliary port of l.

1.4 Pre-proof-structures

Given a ppps, one might expect that there is an intuitive notion of "above/below" for its ports as done in the following definition 40, and that an axiom port is "above" a unique conclusion or cut port. But for general ppps this is wrong, because a ppps might have a "vicious cycle", i.e. two ports which are "above" each other.

Definition 40. *For every $\Phi \in$ **PPPS**, we define a the binary relation $<^1_\Phi$ on $\mathcal{P}(\Phi)$ as follows: $p <^1_\Phi p'$ if one of the following conditions holds:*

- *there exists a cell l of Φ such that p is the principal port of l and p' is an auxiliary port of l,*
- *p' is the principal port of some cell l' of Φ, p is an auxiliary port of some cell l of Φ and $\{p, p'\}$ is a wire of Φ.*

The binary relation $\leq_\Phi$ (resp. $<_\Phi$) on $\mathcal{P}(\Phi)$ is the reflexive-transitive (resp. transitive) closure of $<^1_\Phi$. For every $n \in \mathbb{N}$ and $p, p' \in \mathcal{P}(\Phi)$, we write that $p \leq^n_\Phi p'$ if there exists a finite sequence $(p_i)_{0\leq i\leq n}$ of ports of Φ such that $p_0 = p$, $p_n = p'$ and $p_i <^1_\Phi p_{i+1}$ for every $0 \leq i \leq n-1$.

Our definition of $\leq_\Phi$ for a ppps Φ is identical to that one in [dCT12], where we consider cut ports as minimal elements (i.e. as conclusions of Φ).

Remark 41. Let $\Phi \in$ **PPPS**.

1. If $p, q \in \mathcal{P}(\Phi)$ are such that $p \leq_\Phi q$ then there exists a (box-crossing) path in Φ from p to q crossing no cuts nor axioms nor arrows. In particular, if $q \in \mathcal{I}(\Phi) \cup \mathcal{D}(\Phi)$ then there is no $p \in \mathcal{P}(\Phi)$ such that $p <^1_\Phi q$.

2. $\leq_\Phi \subseteq \mathcal{P}(\Phi)^2$ is a pre-order relation by definition, but in general, $\leq_\Phi$ is not an order relation because it is not antisymmetric. For instance, take $\Phi \in$ **PPPS** consisting of a cell l such that $\mathsf{a}_\Phi(l) > 0$ and a wire $\{p, q\}$ where p is the principal port of l and q is an auxiliary port of l: $p \leq_\Phi q$ (by the first condition) and $q \leq_\Phi p$ (by the second condition), but $p \neq q$. This is an example of "vicious cycle". A more general

example of $\Phi \in \mathbf{PPPS}$ with a "vicious cycle" is a finite sequence of cells $l_0, \ldots, l_n$ and a finite sequence of wires $w_0, \ldots, w_n$ with $n \in \mathbb{N}$ such that $\mathsf{a}_\Phi(l_i) > 0$ (where p_i and q_i are respectively the principal and an auxiliary port of l_i) for every $0 \leq i \leq n$, $w_i = \{p_i, q_{i+1}\}$ for every $0 \leq i \leq n-1$ and $w_n = \{p_n, q_0\}$.

The non-antisymmetry of $\leq_\Phi$ means that if $p, q \in \mathcal{P}(\Phi)$ are such that $p \leq_\Phi^n q$ with $n > 1$, not necessarily $p \neq q$.

3. It is immediate to verify that $<_\Phi^1 \subseteq \preceq_\Phi^1 \subseteq \preccurlyeq_\Phi^1$ and so $\leq_\Phi \subseteq \preceq_\Phi \subseteq \preccurlyeq_\Phi$. Therefore:

    - if $\preccurlyeq_\Phi$ is antisymmetric then $\preceq_\Phi$ is so;
    - if $\preceq_\Phi$ is antisymmetric then $\leq_\Phi$ is so.

    The converses fail to hold: take for instance a ppps Φ consisting of a 1-cell whose principal port is connected by a wire to the auxiliary port of a promotion cell l whose principal port is connected to the auxiliary port p of an unary terminal ?-cell and such that $\mathsf{auxd}_\Phi(l) = \{p\}$, then $\leq_\Phi$ and $\preceq_\Phi$ are antisymmetric but $\preccurlyeq_\Phi$ is not.

    Another example is a ppps Φ consisting of two 1-cells whose principal ports are connected by two wires to the auxiliary ports of respectively an unary ?-cell l' and a promotion cell l such that the principal ports of l and l' are connected by a cut and $\mathsf{bc}_\Phi(l) = \{\mathsf{P}_\Phi^{\mathsf{pri}}(l), \mathsf{P}_\Phi^{\mathsf{pri}}(l')\}$: thus $\leq_\Phi$ is antisymmetric but $\preceq_\Phi$ and $\preccurlyeq_\Phi$ are not.

The following lemmas 42 and 45 about the relation $\leq_\Phi$ are reformulations of lemmas 10 and 14 in [dCT12] to the case of ppps with cuts.

Lemma 42. *Let* $\Phi \in \mathbf{PPPS}$*, let* $p, q_1, q_2 \in \mathcal{P}(\Phi)$*, let* $c, c' \in \mathcal{P}^{\mathsf{free}}(\Phi) \cup \bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D}(\Phi)$ *and let* $a \in \bigcup \mathcal{A}x(\Phi) \cup \bigcup_{l \in \mathcal{C}(\Phi)} \{\mathsf{P}_\Phi^{\mathsf{pri}}(l) \mid \mathsf{a}_\Phi(l) = 0\} \cup \mathcal{D}(\Phi)$*.*

1. *If* $q_1 <_\Phi^1 p$ *and* $q_2 <_\Phi^1 p$ *then* $q_1 = q_2$*.*
2. *If* $q_1 \leq_\Phi p$ *and* $q_2 \leq_\Phi p$ *then* $q_1 \leq q_2$ *or* $q_2 \leq q_1$*.*
3. *If* $p \leq_\Phi c$ *(resp.* $a \leq_\Phi p$*) then* $p = c$ *(resp.* $a = p$*).*
4. *If* $c \leq_\Phi p$ *and* $c' \leq_\Phi p$ *then* $c = c'$*.*
5. *If there is no* $q \in \mathcal{P}(\Phi)$ *such that* $q <_\Phi^1 p$ *(resp.* $p <_\Phi^1 q$*), then* $p \in \mathcal{P}^{\mathsf{free}}(\Phi) \cup \bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D}(\Phi)$ *(resp.* $p \in \bigcup \mathcal{A}x(\Phi) \cup \bigcup_{l \in \mathcal{C}(\Phi)} \{\mathsf{P}_\Phi^{\mathsf{pri}}(l) \mid \mathsf{a}_\Phi(l) = 0\} \cup \mathcal{D}(\Phi)$*).*

Proof.

1. Since every cell has exactly one principal port and because three ports cannot be connected by two wires.

2. Proof by induction on $n \in \mathbb{N}$ where n is such that $q_1 \leq_\Phi^n p$. If $n = 0$ then $q_1 = p$ and so $q_2 \leq q_1$. If $n > 0$ then there exists $p_1 \in \mathcal{P}(\Phi)$ such that $q_1 \leq_\Phi^{n-1} p_1 <_\Phi^1 p$: if $q_2 = p$ then $q_1 \leq q_2$; otherwise there exists $p_2 \in \mathcal{P}(\Phi)$ such that $q_2 \leq_\Phi^n p_2 <_\Phi^1 p$, so $p_1 = p_2$ by lemma 42.1, therefore $q_1 \leq_\Phi q_2$ or $q_2 \leq_\Phi q_1$ by induction hypothesis applied to p_1.

3. If $c \in \mathcal{P}^{\mathsf{free}}(\Phi) \cup \bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D}(\Phi)$ (resp. $a \in \bigcup \mathcal{A}x(\Phi) \cup \bigcup_{l \in \mathcal{C}(\Phi)} \{\mathsf{P}_\Phi^{\mathsf{pri}}(l) \mid \mathsf{a}_\Phi(l) = 0\} \cup \mathcal{D}(\Phi)$), then $p \not<_\Phi^1 c$ (resp. $a \not<_\Phi^1 p$) for every $p \in \mathcal{P}(\Phi)$.

4. By lemma 42.2, $c \leq c'$ or $c' \leq c$; in any case, $c = c'$ by lemma 42.3.

5. If $p \notin \mathcal{P}^{\mathsf{free}}(\Phi) \cup \bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D}(\Phi)$ (resp. $p \notin \bigcup \mathcal{A}x(\Phi) \cup \bigcup_{l \in \mathcal{C}(\Phi)} \{\mathsf{P}_\Phi^{\mathsf{pri}}(l) \mid \mathsf{a}_\Phi(l) = 0\} \cup \mathcal{D}(\Phi)$), then there are only two cases:
   - either $p \in \mathcal{P}^{\mathsf{pri}}(\Phi)$ (resp. $p \in \mathcal{P}^{\mathsf{aux}}(\Phi)$) and there exists $q \in \mathcal{P}^{\mathsf{aux}}(\Phi)$ (resp. $q \in \mathcal{P}^{\mathsf{pri}}(\Phi)$) such that $\{p, q\} \in \mathcal{W}(\Phi)$, so $q <_\Phi^1 p$ (resp. $p <_\Phi^1 q$);
   - or $p \in \mathcal{P}_l^{\mathsf{aux}}(\Phi)$ (resp. $p = \mathsf{P}_\Phi^{\mathsf{pri}}(l)$) for some cell l of Φ such that $\mathcal{P}_l^{\mathsf{aux}}(\Phi) \neq \emptyset$, so $\mathsf{P}_\Phi^{\mathsf{pri}}(l) <_\Phi^1 p$ (resp. there exists $q \in \mathcal{P}_l^{\mathsf{aux}}(\Phi)$ such that $p <_\Phi^1 q$).

□

Lemma 42.3 means that conclusions, cuts ports and deadlocks (resp. axiom ports, principal ports of 0-ary cells and deadlocks) of a ppps Φ are the minimal (resp. maximal) elements of the pre-order relation $\leq_\Phi$. Lemma 42.4 implies that in a ppps Φ, an axiom port cannot be "above" (in the sense of definition 40) two different conclusions or cut ports of Φ.

Definition 43 (Pre-proof-structure)**.** *A* pre-proof-structure *(or* pps *for short) is a $\Phi \in$ **PPPS** such that $\leq_\Phi$ is antisymmetric.*

We denote by **PPS** *the set of pre-proof-structures.*

We denote by $\mathbf{PPS}_{\mathsf{DiLL}_0}$ the set of $\Phi \in \mathbf{PPS}$ such that $\mathcal{C}^{\mathsf{prom}}(\Phi) = \emptyset$, whose elements are the DiLL_0-proof-structures[10] *(or* DiLL_0-ps *for short).*

We remind that in a pps Φ, antisymmetry of the relation $\leq_\Phi$ entails that $\leq_\Phi$ is an order: this prevents from creating in Φ the "vicious cycles" seen in remark 41.2. Therefore, we can see $\leq_\Phi$ as a definition of a relation "above/below" for the ports of a pps Φ. We will see that $\preceq_\Phi$ and $\prec_\Phi$ extend this relation in the case of a proof-structure Φ (see proposition 55.1).

Remark 44. Let $\Phi \in$ **PPS** and $\Phi' \in$ **PPPS**. If $\Phi \simeq \Phi'$ then $\Phi' \in$ **PPS**.

Lemma 45. *Let $\Phi \in$ **PPS** and let $p \in \mathcal{P}(\Phi)$:*

1. *there exists exactly one $c \in \mathcal{P}^{\mathsf{free}}(\Phi) \cup \bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D}(\Phi)$ such that $c \leq_\Phi p$.*

---

[10] We have deliberately forgotten a "pre-". The reason will explained in remark 53.

2. *there exists at least one* $a \in \bigcup \mathcal{A}x(\Phi) \cup \bigcup_{l \in \mathcal{C}(\Phi)} \{\mathsf{P}^{\mathsf{pri}}_{\Phi}(l) \mid \mathsf{a}_{\Phi}(l) = 0\} \cup \mathcal{D}(\Phi)$ *such that* $p \leq_{\Phi} a$.

PROOF.

1. For the unicity, apply lemma 42.4. For the existence, we build an initial segment I of $\mathbb{N}$ and a non-empty "downward" path $(p_i)_{i \in I}$ in Φ as follows:
   - $0 \in I$ and $p_0 = p$;
   - if $i \in I$ then:
     - if $p_i \in \mathcal{P}^{\mathsf{free}}(\Phi) \cup \bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D}(\Phi)$ then $I = \{0, \dots, i\}$,
     - otherwise, by lemma 42.5, there exists $q \in \mathcal{P}(\Phi)$ such that $q <^1_{\Phi} p$, and then $i+1 \in I$ and $p_{i+1} = q$.

   By construction, $p_{i+1} <^1_{\Phi} p_i$ for every $i, i+1 \in I$. By antisymmetry of $\leq_{\Phi}$, I is finite (otherwise there would be a "vicious cycle" as $\mathcal{P}(\Phi)$ is a finite set, i.e. there would exist $i, j \in I$ such that $i < j$ and $p_i = p_j$, so $p_i <^1_{\Phi} p_{j-1}$ and $p_{j-1} \leq_{\Phi} p_i$, that is impossible by antisymmetry of $\leq_{\Phi}$), hence there exists $n \in \mathbb{N}$ such that $I = \{0, \dots, n\}$ and $p_n \in \mathcal{P}^{\mathsf{free}}(\Phi) \cup \bigcup \mathcal{C}uts(\Phi) \cup \mathcal{D}(\Phi)$, with $p_n \leq_{\Phi} p$.

2. We build an initial segment I of $\mathbb{N}$ and a non-empty "upward" path $(p_i)_{i \in I}$ in Φ as follows:
   - $0 \in I$ and $p_0 = p$;
   - if $i \in I$ then:
     - if $p_i \in \bigcup \mathcal{A}x(\Phi) \cup \bigcup_{l \in \mathcal{C}(\Phi)} \{\mathsf{P}^{\mathsf{pri}}_{\Phi}(l) \mid \mathsf{a}_{\Phi}(l) = 0\} \cup \mathcal{D}(\Phi)$ then $I = \{0, \dots, i\}$,
     - otherwise, by lemma 42.5, there exists $q \in \mathcal{P}(\Phi)$ such that $p <^1_{\Phi} q$, and then $i+1 \in I$ and $p_{i+1} = q$.

   By construction, $p_i <^1_{\Phi} p_{i+1}$ for every $i, i+1 \in I$. By antisymmetry of $\leq_{\Phi}$, I is finite (otherwise there would be a "vicious cycle" as $\mathcal{P}(\Phi)$ is a finite set, i.e. there would exist $i, j \in I$ such that $i < j$ and $p_i = p_j$, so $p_{j-1} <^1_{\Phi} p_i$ and $p_i \leq_{\Phi} p_{j-1}$, that is impossible by antisymmetry of $\leq_{\Phi}$), hence there exists $n \in \mathbb{N}$ such that $I = \{0, \dots, n\}$ and $p_n \in \bigcup \mathcal{A}x(\Phi) \cup \bigcup_{l \in \mathcal{C}(\Phi)} \{\mathsf{P}^{\mathsf{pri}}_{\Phi}(l) \mid \mathsf{a}_{\Phi}(l) = 0\} \cup \mathcal{D}(\Phi)$, with $p \leq_{\Phi} p_n$. □

Lemma 45.1 entails that in a pps Φ, any axiom port is above a unique conclusion or cut port of Φ. Lemmas 45.1-2 means that if Φ is a pps then the order relation $\leq_{\Phi}$ defines a natural "top-down" orientation on ports from axiom ports and principal ports of 0-ary cells to conclusions and cut ports of Φ, where by lemma 42.3 deadlocks of Φ have no ports above or below them (in the sense of definition 40).

1.5 Boxes and (non-inductive) proof-structures

Similarly to [dCT12], the main difference of our syntax from the usual syntaxes of linear logic (or differential linear logic with boxes) proof-nets (see for example [Gir87, Lau03, Pag09, dCPT11, Tra11]) is the absence of an explicit (inductive) constructor for boxes: this leads to define a box as a sort of sub-graph satisfying some conditions. This more "geometrical" approach was followed for example in [DR95, Tor03, MP07, dCT12]. In our syntax we have to reconstruct the boxes of a pps Φ by using some "geometrical" informations coming from Φ, in particular the arrow functions auxd_Φ and bc_Φ play a crucial role. Each promotion cell in a pre-proof-structure corresponds to the so-called "principal ports of a box" in the usual syntaxes of linear logic proof-nets. More delicate is the issue of marking out the other boundaries of a box, which are called "auxiliary ports of a box" in the usual syntax (corresponding to auxiliary doors in our syntax), and the content of a box: in order to do that, some conditions are to be fulfilled.

Definition 46 (Box)**.** *Let* $\Phi \in \mathbf{PPS}$.

Let $l \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ *and let* p_l *be the unique auxiliary port of* l.[11] *We say that* "the box of l is defined in Φ" *or* "l has a box in Φ" *when, for every* $q, q' \in \mathcal{P}(\Phi)$:

1. *if* $q \in \mathsf{auxd}_\Phi(l)$ *then* $q \not\leq_\Phi p_l$ *and* $p_l \not\leq_\Phi q$;
2. *if* $q, q' \in \mathsf{auxd}_\Phi(l)$ *with* $q \neq q'$ *then* $q \not\leq_\Phi q'$ *and* $q' \not\leq_\Phi q$;
3. *for every* $l' \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ $\preccurlyeq$*-above* l, *if* $q' \in \mathsf{cutports}_\Phi(l')$ *and* $q \in \mathsf{doors}_\Phi(l)$, *then* $q' \not<_\Phi q$;
4. *if* q *is the end port of a path above* l *in* Φ *and if* $q' = q$ *or* $\{q, q'\} \in \mathcal{A}x(\Phi)$, *then there exists* $r \in \mathsf{doors}_\Phi(l) \cup \mathsf{cdabove}_\Phi(l)$ *such that* $r \leq_\Phi q'$.[12]

We denote by $\mathcal{C}^{\mathsf{box}}(\Phi)$ *the set of* $l \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ *having a box in* Φ, *whose elements are the* box cells of Φ .

Remark 47. $\mathcal{C}^{\mathsf{box}}(\Phi) = \emptyset$ for every $\Phi \in \mathbf{PPS}_{\mathsf{DiLL}_0}$, as $\mathcal{C}^{\mathsf{prom}}(\Phi) = \emptyset$. Hence $\leq_\Phi = \preceq_\Phi = \preccurlyeq_\Phi$ for every $\Phi \in \mathbf{PPS}_{\mathsf{DiLL}_0}$

When a promotion cell l satisfies conditions 1, 2, 3 and 4 in definition 46, we are able to compute the box associated with l, by taking into account only the "geometrical" informations available in a pps. The following definition of how to compute a box is quite delicate and it is inspired by the analogous

---

[11] Remind that $\mathsf{a}_\Phi(l) = 1$ and $\mathsf{t}_\Phi(l) = \,!$ since l is a promotion cell.

[12] Because of the definition of auxd_Φ and $\mathsf{cdabove}_\Phi$, this condition means that if l has a box in Φ then either $r = p_l$ or auxd_Φ has to point from l to r or bc_Φ has to point from l' to r, where $l' \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ is "above" l (in the sense of definition 38). In particular, there exists a path above l ending in q'.

definition in [dCT12], with two further complications: our definition is completely non-inductive and our boxes might contain cuts. Intuitively, for every $l \in \mathcal{C}^{\mathsf{box}}(\Phi)$, $\mathsf{auxd}_\Phi(l)$ gives the auxiliary doors of the box $\mathsf{box}_\Phi(l)$ associated with l (i.e. the boundaries of this box as well as l), $\mathsf{bc}_\Phi(l)$ gives the cut ports and deadlocks belonging to $\mathsf{box}_\Phi(l)$ and not belonging to more inner boxes, the content of $\mathsf{box}_\Phi(l)$ is "all that is above" the doors of $\mathsf{box}_\Phi(l)$ and the cut ports pointed by $\mathsf{bc}_\Phi(l)$.

Definition 48 (Computation of a box)**.** *Let $\Phi \in \mathbf{PPS}$, let $v \in \mathcal{C}^{\mathsf{box}}(\Phi)$ and let r_v be the unique auxiliary port of v. We set:*

$$\begin{aligned} \mathsf{inbox}_\Phi(v) &= \{p \in \mathcal{P}(\Phi) \mid \exists \textit{path above } v \textit{ in } \Phi \textit{ ending in } p\} \\ B'_v &= \mathsf{inbox}_\Phi(v) \setminus \{r_v\} \\ B_v &= \begin{cases} \mathsf{inbox}_\Phi(v) & \textit{if } r_v \in \bigcup \mathcal{A}x(\Phi) \\ B'_v & \textit{otherwise.} \end{cases} \end{aligned}$$

$\mathsf{inbox}_\Phi(v)$ *is the* content of the box of v in Φ*.*

Let $\mathcal{L}_0$ and $\mathcal{P}_0$ be two sets such that there exist two bijections $p_1 : \mathcal{L}_0 \to \mathsf{auxd}_\Phi(v)$ and $p_0 : \mathcal{L}_0 \to \mathcal{P}_0$, moreover $\mathcal{L}_0 \cap (\mathscr{P}(\mathsf{C}_\Phi)(B'_v) \setminus \mathscr{P}(\mathsf{C}_\Phi)(\mathsf{auxd}_\Phi(v))) = \emptyset$ and $\mathcal{P}_0 \cap B_v = \emptyset$. We set $\mathbb{C}_v = (\mathsf{t}_v, \mathcal{P}_v, \mathsf{C}_v, \mathsf{P}^{\mathsf{pri}}_v, \mathsf{P}^{\mathsf{left}}_v)$ where:

- *t_v is a function from $\mathcal{C}(\mathbb{C}_v) = \mathcal{L}_0 \cup (\mathscr{P}(\mathsf{C}_\Phi)(B'_v) \setminus \mathscr{P}(\mathsf{C}_\Phi)(\mathsf{auxd}_\Phi(v)))$ to $\mathcal{T}$ such that $\mathsf{t}_v(l) = ?$ for every $l \in \mathcal{L}_0$ and $\mathsf{t}_v \restriction_{\mathscr{P}(\mathsf{C}_\Phi)(B'_v) \setminus \mathscr{P}(\mathsf{C}_\Phi)(\mathsf{auxd}_\Phi(v))} = \mathsf{t}_\Phi \restriction_{\mathscr{P}(\mathsf{C}_\Phi)(B'_v) \setminus \mathscr{P}(\mathsf{C}_\Phi)(\mathsf{auxd}_\Phi(v))}$;*[13]
- *$\mathcal{P}_v = \mathcal{P}_0 \cup B'_v$;*[14]
- *$\mathsf{C}_v : \mathcal{P}_v \to \mathcal{C}(\mathbb{C}_v)$ is such that, for every $p \in \mathcal{P}_v$,*

$$\mathsf{C}_v(p) = \begin{cases} \mathsf{C}_\Phi(p) & \textit{if } p \in B'_v \setminus \mathsf{auxd}_\Phi(v) \\ l & \textit{if } p = p_0(l) \textit{ or } p = p_1(l) \textit{ for some } l \in \mathcal{L}_0 \end{cases}$$

- *$\mathsf{P}^{\mathsf{pri}}_v : \mathcal{C}(\mathbb{C}_v) \to \mathcal{P}_v$ is such that, for every $l \in \mathcal{C}(\mathbb{C}_v)$,*

$$\mathsf{P}^{\mathsf{pri}}_v(l) = \begin{cases} p_0(l) & \textit{if } l \in \mathcal{L}_0 \\ \mathsf{P}^{\mathsf{pri}}_\Phi(l) & \textit{otherwise} \end{cases}$$

- $\mathsf{P}^{\mathsf{left}}_v = \mathsf{P}^{\mathsf{left}}_\Phi \restriction_{\mathcal{C}^{\otimes, \parr}(\Phi) \cap \mathcal{C}(\mathbb{C}_v)}$.

The box of v in Φ *is* $\mathsf{box}_\Phi(v) = (\mathbb{C}_v, \mathcal{I}_v, \mathcal{D}_v, \mathcal{W}_v, \mathsf{auxd}_v, \mathsf{bc}_v)$ *where:*

- $$\mathcal{I}_v = \begin{cases} \{r_v\} & \textit{if } r_v \in B_v \\ \emptyset & \textit{otherwise;} \end{cases}$$

---

[13] Notice that $v = \mathsf{C}_\Phi(r_v) \notin \mathcal{C}(\mathbb{C}_v)$.

[14] Note that $r_v \notin \mathcal{P}_v$.

- $\mathcal{W}_v = \{w \in \mathcal{W}(\Phi) \mid w \subseteq B_v\}$;
- $\mathcal{D}_v = \mathcal{D}(\Phi) \cap \mathsf{inbox}_\Phi(v)$;
- $\mathsf{auxd}_v = \mathsf{auxd}_\Phi\restriction_{\mathcal{C}^{\mathsf{prom}}(\Phi)\cap\mathcal{C}(\mathbb{C}_v)}$;
- $\mathsf{bc}_v = \mathsf{bc}_\Phi\restriction_{\mathcal{C}^{\mathsf{prom}}(\Phi)\cap\mathcal{C}(\mathbb{C}_v)}$.

The idea in definition 48 is that, given $\Phi \in \mathbf{PPS}$, we build the box $\mathsf{box}_\Phi(v)$ of $v \in \mathcal{C}^{\mathsf{box}}(\Phi)$ by starting from v by means of the paths above v (to get $\mathsf{inbox}_\Phi(v)$) and then by reconstructing $\mathsf{box}_\Phi(v)$ as the (in some sense) "smallest sub-pps" of Φ containing $\mathsf{inbox}_\Phi(v)$: nevertheless even if we give a precise definition of sub-pps, it is not correct to say that $\mathsf{box}_\Phi(v)$ is the smallest sub-pps of Φ containing $\mathsf{inbox}_\Phi(v)$ because $\mathsf{inbox}_\Phi(v) \subseteq \mathcal{P}(\Phi)$ but in general $\mathcal{P}_v \not\subseteq \mathcal{P}(\Phi)$. Some syntactical complications in the definition of $\mathsf{box}_\Phi(v)$ are due to get that propositions 49 and 54 hold, for example the issue whether the auxiliary port of v belongs or not to $\mathsf{box}_\Phi(v)$, and the add of the sets $\mathcal{L}_0$ and $\mathcal{P}_0$: every $l \in \mathcal{L}_0$ is a unary ?-cell, where $p_0(l)$ (resp. $p_1(l)$) is its principal (resp. unique auxiliary) port. Note that auxd_v is like auxd_Φ but it forgets all the arrows associated with the promotion cells of Φ that are not in $\mathcal{C}(\mathbb{C}_v)$, including v. Similarly for bc_v.

Proposition 49. *Let $\Phi \in$ **PPS** and let $v \in \mathcal{C}^{\mathsf{box}}(\Phi)$. With reference to notation of definition 48.*

1. $\mathbb{C}_v \in$ **Cells** *and $l \in \mathcal{C}(\Phi)$ for every $l \in \mathcal{C}(\mathbb{C}_v) \smallsetminus \mathcal{L}_0$.*
2. $\mathsf{box}_\Phi(v) \in$ **PPS** *with $\mathbb{C}(\mathsf{box}_\Phi(v)) = \mathbb{C}_v$.*

Proof. Evident. □

Remark 50. Let $\Phi \in$ **PPS**.

1. Given $v \in \mathcal{C}^{\mathsf{box}}(\Phi)$ whose unique auxiliary port is r_v, one has $r_v \in \mathsf{inbox}_\Phi(v)$ and $\{r_v, p_v\} \in \mathcal{W}(\Phi)$ for some $p_v \in \mathsf{inbox}_\Phi(v) \cap \mathcal{P}(\mathsf{box}_\Phi(v))$. If $\{r_v, p_v\} \in \mathcal{A}x(\Phi)$ (resp. $\{r_v, p_v\} \notin \mathcal{A}x(\Phi)$) then $\mathsf{inbox}_\Phi(v) \subseteq \mathcal{P}(\mathsf{box}_\Phi(v))$ (resp. $\mathsf{inbox}_\Phi(v) \smallsetminus \{r_v\} \subseteq \mathcal{P}(\mathsf{box}_\Phi(v))$), $\{r_v, p_v\} \in \mathcal{W}(\mathsf{box}_\Phi(v))$ (resp. $\{r_v, p_v\} \notin \mathcal{W}(\mathsf{box}_\Phi(v))$) and r_v (resp. p_v) is the unique $q \in \mathcal{P}^{\mathsf{free}}(\mathsf{box}_\Phi(v))$ such that $\mathsf{P}^{\mathsf{pri}}_\Phi(v) <_\Phi q$.
2. Let $v, v' \in \mathcal{C}^{\mathsf{box}}(\Phi)$: there exists a path above v in Φ ending in the unique auxiliary port $p_{v'}$ of v' iff $\mathsf{inbox}_R(v') \subseteq \mathsf{inbox}_R(v)$. Indeed, for the left-to-right direction, for every $p \in \mathcal{P}(\Phi)$, if $p \in \mathsf{inbox}_R(v')$ then there exists a path φ' above v' in Φ ending in p; by hypothesis and definition of ascending path, there exists a path φ above v in Φ ending in $\mathsf{P}^{\mathsf{pri}}_\Phi(v')$, hence $\varphi \cdot \varphi'$ is a path above v in Φ ending in p, hence $p \in \mathsf{inbox}_R(v)$. Conversely, $p_{v'} \in \mathsf{inbox}_R(v') \subseteq \mathsf{inbox}_R(v)$, so there exists a path above v in Φ ending in $p_{v'}$.

The following lemma shows an expected property of boxes in a pps: all that is below (in the sense of definition 40) the doors of a box cannot be inside this box, i.e. the doors of the box associated with a promotion cell are the boundaries of this box. The proof of this lemma uses all the conditions mentioned in definition 46, so it reveals indirectly their importance.

Lemma 51. *Let* $\Phi \in \mathbf{PPS}$ *and* $v \in \mathcal{C}^{\mathsf{box}}(\Phi)$. *For every* $p, q \in \mathcal{P}(\Phi)$, *if* $p <_\Phi q$ *and* $q \in \mathsf{doors}_\Phi(v)$ *then* $p \notin \mathsf{inbox}_\Phi(v)$.

Proof. Let $q \in \mathsf{doors}_\Phi(v)$. If $p \in \mathsf{inbox}_\Phi(v)$ then there exists a path above v ending in p. By condition 4 in definition 46, there exists $r \in \mathsf{doors}_\Phi(v) \cup \mathsf{cdabove}_\Phi(v)$ such that $r \leq_\Phi p$. If $r \in \mathsf{doors}_\Phi(v)$ then $p \not<_\Phi q$ by conditions 1-2 of definition 46. Otherwise $r \in \mathsf{cdabove}_\Phi(v)$ and then there exists $v' \in \mathcal{C}^{\mathsf{prom}}(\Phi)$ $\preccurlyeq$-above v such that $r \in \mathsf{bc}_\Phi(v')$; there are only two cases: either $r \in \mathsf{deadlocks}_\Phi(v')$ and so $r = p$ and $p \not<_\Phi q$ by lemma 42.3, or $r \in \mathsf{cutports}_\Phi(v')$ and so $p \not<_\Phi q$ otherwise $r \leq_\Phi p <_\Phi q$ that is impossible by condition 3 in definition 46. ☐

We can introduce now the syntactical objects for which we prove our main result: proof-structures.

Definition 52 (Proof-structure)**.** *A* proof-structure *(or* ps *for short) is a* $R \in \mathbf{PPS}$ *such that:*

- $\mathcal{C}^{\mathsf{box}}(R) = \mathcal{C}^{\mathsf{prom}}(R)$;[15]
- *(*nesting condition*) for every* $l, l' \in \mathcal{C}^{\mathsf{box}}(R)$ *either* $\mathsf{inbox}_R(l) \subseteq \mathsf{inbox}_R(l')$ *or* $\mathsf{inbox}_R(l') \subseteq \mathsf{inbox}_R(l)$ *or* $\mathsf{inbox}_R(l) \cap \mathsf{inbox}_R(l') = \emptyset$;

We denote by $\mathbf{PS}$ *the set of proof-structures.*

We denote by $\mathbf{PS}_{\mathsf{MELL}}$ *the set of* $R \in \mathbf{PS}$ *such that* $\mathcal{C}^!(R) = \mathcal{C}^{\mathsf{box}}(R)$, *whose elements are the* MELL-proof-structures *(or* MELL-ps *for short).*

We point out that our definition of proof-structure is completely non-inductive, as in [DR95, Tor03, MP07], differently from the usual definitions of proof-structure in the literature on linear logic and its differential version (see for example [Gir87, Lau03, Pag09, dCPT11, Tra11, dCT12]). This leads to consider the linear logic proof-structures as "really geometrical" objects, in accordance with the Girard's original spirit. This definition allows also to define the cut-elimination directly on these "geometrical" objects. Actually the definition of cut-free proof-structure given in [dCT12] can be reformulated in a non-inductive way (this remark was our starting point).

Remark 53. $\mathbf{PPS}_{\mathsf{DiLL}_0} \subseteq \mathbf{PS}$, since $\mathcal{C}^{\mathsf{box}}(\Phi) \subseteq \mathcal{C}^{\mathsf{prom}}(\Phi) = \emptyset$ for every $\Phi \in \mathbf{PPS}_{\mathsf{DiLL}_0}$.

---

[15] This means that every promotion cell of R has a box in R (in the sense of definition 46).

Proposition 54. *Let* $R \in \mathbf{PS}$ *and let* $l \in \mathcal{C}^{\mathsf{box}}(\Phi)$.

1. $\mathsf{box}_R(l) \in \mathbf{PS}$.
2. *If* $\varphi : R \simeq R'$ *then* $\varphi_{\mathcal{P}}(l) \in \mathcal{C}^{\mathsf{box}}(R')$ *and* $\mathsf{box}_R(l) \simeq \mathsf{box}_{R'}(\varphi_{\mathcal{C}}(l))$.

Proof. Evident. □

The following proposition says that in a ps R, $\preceq_R$ and $\preccurlyeq_R$ are order relations. In a certain sense, in the case of a ps R, $\preceq_R$ and $\preccurlyeq_R$ are "good generalizations" of $\leq_R$ (they extend $\leq_\Phi$ as a relation "above/below" for the ports of R: see remark 41.3 but also the following lemma 57), with the further property that any promotion cell is the least element (with respect to $\preccurlyeq_R$) of the box associated with it (proposition 55.2).

Proposition 55. *Let* $R \in \mathbf{PS}$.

1. $\preceq_R$ *and* $\preccurlyeq_R$ *are order relations on* $\mathcal{P}(R)$.
2. *For every* $v \in \mathcal{C}^{\mathsf{box}}(R)$, *the unique auxiliary port of* v *is the least element in* $\mathsf{inbox}_R(v)$ *with respect to* $\preccurlyeq_R$.
3. *For every* $v, v' \in \mathcal{C}^{\mathsf{box}}(R)$, *if* $v \neq v'$ *then* $\mathsf{inbox}_R(v) \neq \mathsf{inbox}_R(v')$.

Proof.

1. By remarks 39.3 and 41.3, it suffices to show that $\preccurlyeq_R$ is antisymmetric. Let us suppose by absurd that $\preccurlyeq_R$ is not antisymmetric, so there exist $p, q \in \mathcal{P}(R)$ such that $p \preccurlyeq_R q$, $q \preccurlyeq_R p$ and $p \neq q$. As $R \in \mathbf{PPS}$, $\leq_R$ is antisymmetric, thus it is impossible that $p \leq_R q$ and $q \leq_R p$. Hence, there exist $p_0, p_1, p_2 \in \mathcal{P}(R)$ and $v \in \mathcal{C}^{\mathsf{prom}}(R) = \mathcal{C}^{\mathsf{box}}(R)$ (as $R \in \mathbf{PS}$) such that p_0 (resp. p_1) is the principal (resp. unique auxiliary) port of v, $p_2 \in \mathsf{auxd}_R(v) \cup \mathsf{cutports}_R(v)$ and $p_2 \preccurlyeq_R p_0$ ($p_2 \notin \mathsf{deadlocks}_R(v)$ by remark 39.2 since $p_2 \preccurlyeq_R^n p_1$ for some $n \in \mathbb{N}^*$). Therefore $p_0 \in \mathsf{inbox}_R(v)$ $p_0 <_R p_1$ and $p_1 \in \mathsf{doors}_R(v)$, that is impossible by lemma 51.
2. By definition of $\mathsf{inbox}_R(v)$ and $\preccurlyeq_R$, if p is the unique auxiliary port of v then $p \preccurlyeq_R q$ for every $q \in \mathsf{inbox}_R(v)$. We conclude thanks to proposition 55.1.
3. Let p_v (resp. $p_{v'}$) the unique auxiliary port of v (resp. v'). If there is not path above v ending in $p_{v'}$ then $p_{v'} \notin \mathsf{inbox}_R(v)$ but $p_{v'} \in \mathsf{inbox}_R(v')$, hence $\mathsf{inbox}_R(v) \neq \mathsf{inbox}_R(v')$. Otherwise there exists a path above v ending in $p_{v'}$ (thus $p_v \preccurlyeq_R p_{v'}$), moreover $p_v \neq p_{v'}$ since $v \neq v'$; by antisymmetry of $\preccurlyeq_R$ (proposition 55.1), there is no path above v' ending in p_v, so $p_v \notin \mathsf{inbox}_R(v')$ but $p_{v'} \in \mathsf{inbox}_R(v')$, therefore $\mathsf{inbox}_R(v) \neq \mathsf{inbox}_R(v')$.

□

Remark 56. In the proof of proposition 55 the hypothesis that $R \in \mathbf{PS}$ is used only to ensure that every promotion cell of R has a box defined in R, in particular we never used the hypothesis that R fulfills the nesting condition. Notice that in the examples showed in remark 41.3 the promotion cells have no box defined. An example of pps Φ such that $\mathcal{C}^{\mathsf{box}}(\Phi) = \mathcal{C}^{\mathsf{prom}}(\Phi)$ (and so $\preccurlyeq_\Phi$ is an order relation) but the nesting condition is not fulfilled (and so $\Phi \notin \mathbf{PS}$) is the following: take one $\bot$-cell whose principal port is connected by a wire to the auxiliary port p of an unary ?-cell and two 1-cells whose principal ports are both connected by a wire respectively to the unique auxiliary ports p_v and $p_{v'}$ of two !-cells v and v' which have both an arrow pointing to p; in this case $p \in \mathsf{inbox}_\Phi(v) \cap \mathsf{inbox}_\Phi(v')$ but $\mathsf{inbox}_\Phi(v) \nsubseteq \mathsf{inbox}_\Phi(v')$ (because the $p_v \in \mathsf{inbox}_R(v) \smallsetminus \mathsf{inbox}_R(v')$) and $\mathsf{inbox}_R(v) \nsubseteq \mathsf{inbox}_R(v')$ (because $p_{v'} \in \mathsf{inbox}_R(v') \smallsetminus \mathsf{inbox}_R(v)$).

Given a ps R, we can generalize lemmas 42 and 45 for the order relation $\preceq_R$, which is a restriction of $\preccurlyeq_R$ (see remark 41.3).

Lemma 57. *Let $R \in \mathbf{PS}$ and $p, p', q \in \mathcal{P}(R)$.*

1. *If $p \preceq_R^1 q$ and $p' \preceq_R^1 q$ then $p = p'$.*
2. *If $p \preceq_R q$ and $p' \preceq_R q'$ then either $p \preceq_R p'$ or $p' \preceq_R p$.*
3. *For every $c \in \mathcal{P}^{\mathsf{free}}(R) \cup \bigcup \mathcal{C}uts_0(R) \cup \mathcal{D}_0(R)$, if there exists an ascending path from q to c then $q = c$.*
4. *There exists at most one box-crossing path in R from p to q.*
5. *There exists a unique $c \in \mathcal{P}^{\mathsf{free}}(R) \cup \bigcup \mathcal{C}uts_0(R) \cup \mathcal{D}_0(R)$ such that $c \preceq_R q$.*

Proof.

1. $q \notin \mathcal{I}(R) \cup \mathcal{D}_0(R)$ by remark 39.2, hence there are only three cases:
   - $q \in \mathcal{P}^{\mathsf{pri}}(R)$, so $p \in \mathcal{P}^{\mathsf{aux}}(R)$ and either $\{p, q\} \in \mathcal{W}(R)$ or p is the unique auxiliary port of some $l \in \mathcal{C}^{\mathsf{box}}(R)$ such that $q \in \mathsf{cutports}_R(l)$; analogously for p'; by definition of ppps (in particular, condition 1 about the set of wires in definition 16), necessarily $p = p'$;
   - $q \in \mathcal{P}^{\mathsf{aux}}(R)$, so $p = \mathsf{P}_R^{\mathsf{pri}}(l) = p'$ for the $l \in \mathcal{C}(R)$ such that $q \in \mathcal{P}_l^{\mathsf{aux}}(R)$ (because of the definition of box-crossing path);
   - $q \in \mathcal{D}(R)$ and there exists $v \in \mathcal{C}^{\mathsf{box}}(R)$ such that $q \in \mathsf{bc}_R(v)$, hence p and p' are the unique auxiliary port of v, therefore $p = p'$.
2. By induction on the length $n \in \mathbb{N}$ of the box-crossing path $(p_i)_{i \in I}$ from p to q. If $n = 0$ then $p = p_0 = q$, thus there exists a box-crossing path

from p' to p by hypothesis. If $n > 0$ then there are only two cases: if $p' = q$ then there exists a box-crossing path from p to p' by hypothesis; otherwise there exist $q' \in \mathcal{P}(\Phi)$, a box-crossing path from p' to q' and a box-crossing path of length 1 from q' to q, so $p_{n-1} = q'$ by lemma 57.1, therefore there exists a box-crossing path from p to p' or from p' to p by induction hypothesis applied to p_{n-1}.

3. By definition 38, there is no ascending path of length 1 from q to c, hence the only possibility is that the path from q to c has length 0, therefore $q = c$.

4. Let us suppose that $(p_i)_{0 \leq i \leq m}$ and $(q_j)_{0 \leq j \leq n}$ (for some $m, n \in \mathbb{N}$) are two ascending paths such that $p_0 = p = q_0$ and $p_m = q = q_n$: we prove by induction on m that $m = n$ and $p_i = q_i$ for every $0 \leq i \leq m$.

   If $m = 0$ then $p = q_0 = p_0 = q$, furthermore $n = 0$ (otherwise $q = q_0 \preccurlyeq_R^1 q_1$ and $q_1 \preccurlyeq_R q_n = q$ with $q \neq q_1$, that is impossible since $\preccurlyeq_R$ is antisymmetric by proposition 55.1).

   If $m > 0$ then $p_{m-1} = q_{n-1}$ by lemma 57.1. By induction hypothesis, $m - 1 = n - 1$ and $p_i = q_i$ for every $0 \leq i \leq m - 1$, thus we can conclude.

5. For the unicity, if $c, c' \in \mathcal{P}^{\mathsf{free}}(R) \cup \bigcup \mathcal{C}uts_0(R) \cup \mathcal{D}_0(R)$ are such that there exist two box-crossing path in R from c to q and from c' to q then there exists a box-crossing path in R from c to c' or from c' to c by lemma 57.2, hence $c = c'$ by lemma 57.3.

   For the existence, we build an initial segment J of $\mathbb{N}$ and a sequence of finite paths $(\varphi_j)_{j \in J}$ in Φ as follows:

   - $0 \in J$ and φ_0 is the path of length 0 consisting only of q;
   - if $j \in J$ and $\varphi_j = (p_i)_{0 \leq i \leq n}$ then:
     - if $p_0 \in \mathcal{P}^{\mathsf{free}}(\Phi) \cup \bigcup \mathcal{C}uts_0(\Phi) \cup \mathcal{D}_0(R)$ then $J = \{0, \ldots, j\}$,
     - otherwise, $j + 1 \in J$ and $\varphi_{j+1} = (q_i)_{0 \leq i \leq n+1}$ where $q_{i+1} = p_i$ for every $0 \leq i \leq n$ and
       - if $p_0 \in \mathcal{P}_l^{\mathsf{aux}}(\Phi)$ for some $l \in \mathcal{C}(\Phi)$ then $q_0 = \mathsf{P}_{\Phi}^{\mathsf{pri}}(l)$;
       - if $p_0 \in \mathcal{P}^{\mathsf{pri}}(\Phi)$ then either $q_0 \in \mathcal{P}^{\mathsf{aux}}(\Phi)$ with $\{p_0, q_0\} \in \mathcal{W}(\Phi)$, or q_0 is the unique auxiliary port of some $v \in \mathcal{C}^{\mathsf{box}}(\Phi)$ such that $p_0 \in \mathsf{cutports}_{\Phi}(v)$;
       - if $p_0 \in \mathcal{D}(R)$ with $p_0 \in \mathsf{bc}_R(v)$ for some $v \in \mathcal{C}^{\mathsf{box}}(R)$ then q_0 is the unique auxiliary port of v.

   By construction, for every $j \in J$, φ_j is a box-crossing path in R of length j ending in q and moreover, if $j + 1 \in J$ then φ_j is a sub-path of φ_{j+1}. J is a finite set (otherwise there would be an infinite box-crossing path

φ_ω such that, for every $j \in J = \mathbb{N}$, φ_j would be a sub-path of φ_ω, that is impossible since $\mathcal{P}(\Phi)$ is a finite set and $\preccurlyeq_R$ is antisymmetric), hence there exists $m \in \mathbb{N}$ such that $J = \{0, \ldots, m\}$ and φ_m is a box-crossing path from $c \in \mathcal{P}^{\mathsf{free}}(R) \cup \bigcup \mathcal{C}uts_0(R) \cup \mathcal{D}_0(R)$ to q. $\square$

Notice that lemmas 57.1,2,4,5 do not hold in the case of generic ascending paths (i.e. for the order relation $\preccurlyeq_R$).

Definition 58. *Let* $R \in \mathbf{PS}$ *and* $p \in \mathcal{P}(R)$.

We denote by $\mathsf{c}_R(p)$ *the unique* $c \in \mathcal{P}^{\mathsf{free}}(R) \cup \bigcup \mathcal{C}uts_0(R) \cup \mathcal{D}_0(R)$ *such that* $c \preceq_R p$.

We set $\mathsf{boxesof}_R(p) = \{v \in \mathcal{C}^{\mathsf{box}}(R) \mid p \in \mathsf{inbox}_R(v)\}$. *If* $\mathsf{boxesof}_R(p) \neq \emptyset$, *we denote by* $\mathsf{C}^{\mathsf{box}}_R(p)$ *the* $v \in \mathsf{boxesof}_R(p)$ *such that* $\mathsf{inbox}_R(v)$ *is minimal with respect to* $\subseteq$.

The ground of R *is* $\mathcal{G}round(R) = \{q \in \mathcal{P}(R) \mid \mathsf{boxesof}_R(q) = \emptyset\}$.

By lemma 57.5, the function c_R is well-defined for any $R \in \mathbf{PS}$. By the nesting condition, $\mathsf{C}^{\mathsf{box}}_R(p)$ is well-defined for every $R \in \mathbf{PS}$ and $p \in \mathcal{P}(R)$ such that $p \in \mathsf{inbox}_R(v)$ for some $v \in \mathcal{C}^{\mathsf{box}}(R)$.

Given $R \in \mathbf{PS}$ and $p \in \mathcal{P}(R)$, $\mathsf{boxesof}_R(p)$ is morally the set of boxes in R containing p.

In spite of our non-inductive definition of proof-structure, we can recover some typical informations of the inductive one, such as the depth of a port and the depth of a proof-structure. The following definition is nothing but the adaptation to our syntax (allowing ps with cuts and deadlocks) of the corresponding definition in [dCT12].

Definition 59 (Depth). *Let* $R \in \mathbf{PS}$.

Let $p \in \mathcal{P}(R)$ *and let* φ_p *be the box-crossing path from* $\mathsf{c}_R(p)$ *to* p. *The* depth of p in R, *denoted by* $\mathsf{depth}_R(p)$, *is a nonnegative integer defined by:*

$$\begin{aligned}\mathsf{depth}_R(p) \quad = \quad & \mathsf{card}(\{l \in \mathcal{C}^{\mathsf{box}}(R) \mid \varphi_p \textit{ crosses } l\}) \ + \\ & \sum_{q \in \mathcal{A}uxdoors(R)} \mathsf{card}(\{l' \in \mathcal{C}^{\mathsf{box}}(R) \mid \varphi_p \textit{ crosses } q \in \mathsf{auxd}_R(l')\})\,.\end{aligned}$$

If $\mathsf{depth}_R(p) = n$ *then we say that "p is at depth n in R".*

For every $l \in \mathcal{C}(R)$, *the* depth of l in R, *denoted by* $\mathsf{depth}_R(l)$, *is the depth of* $\mathsf{P}^{\mathsf{pri}}_R(l)$ *in* R. *If* $\mathsf{depth}_R(l) = n$ *then we say that "l is at depth n in R".*

For every $v \in \mathcal{C}^{\mathsf{box}}(R)$, *the* depth of $\mathsf{box}_R(v)$ in R, *denoted by* $\mathsf{depth}_R(\mathsf{box}_R(v))$, *is the depth of v in R. If* $\mathsf{depth}_R(\mathsf{box}_R(v)) = n$ *then we say that "$\mathsf{box}_R(v)$ is at depth n in R".*

The depth of R *is* $\mathsf{depth}(R) = \sup\{\mathsf{depth}_R(p) \mid p \in \mathcal{P}(R)\}$.

The depth of a port p in a ps R is well-defined thanks to lemma 57.4, which says that there exists a unique box-crossing path φ_p from $\mathsf{c}_R(p)$ to p. Roughly speaking, $\mathsf{depth}_R(p)$ is calculated by counting the number of promotion cells and arrows pointing to auxiliary doors of R crossed by φ_p.

Remark 60. Let $R \in \mathbf{PS}$.

1. If $R \in \mathbf{PPS}_{\mathsf{DiLL}_0}$ then $\mathsf{depth}(R) = 0$, as $\mathcal{C}^{\mathsf{box}}(R) = \emptyset$ (remember that $\mathbf{PPS}_{\mathsf{DiLL}_0} \subseteq \mathbf{PS}$).

2. Let $v \in \mathcal{C}^{\mathsf{box}}(R)$: if p_v is the unique auxiliary port of v, then $\mathsf{depth}_R(p_v) = \mathsf{depth}_R(\mathsf{box}_R(v)) + 1$. Indeed if $\varphi_{\mathsf{P}^{\mathsf{pri}}_R(v)}$ is the unique box-crossing path from $\mathsf{c}_R(\mathsf{P}^{\mathsf{pri}}_R(v)) = \mathsf{c}_R(p_v)$ to $\mathsf{P}^{\mathsf{pri}}_R(v)$, then $\varphi_{\mathsf{P}^{\mathsf{pri}}_R(v)} \cdot p_v$ is the unique box-crossing path from $\mathsf{c}_R(p_v)$ to p_v and moreover $\varphi_{\mathsf{P}^{\mathsf{pri}}_R(v)} \cdot p_v$ crosses v whereas $\varphi_{\mathsf{P}^{\mathsf{pri}}_R(v)}$ does not.

3. Let $v \in \mathcal{C}^{\mathsf{box}}(R)$: if $p \in \mathsf{inbox}_R(v) \cap \mathcal{P}(\mathsf{box}_R(v))$ then $\mathsf{boxesof}_{\mathsf{box}_R(v)}(p) = \mathsf{boxesof}_R(p) \smallsetminus \{l \in \mathcal{C}^{\mathsf{box}}(R) \mid \mathsf{P}^{\mathsf{pri}}_R(l) \notin \mathsf{inbox}_R(v)\}$. Indeed, let $l \in \mathcal{C}^{\mathsf{box}}(R)$: $l \in \mathsf{boxesof}_{\mathsf{box}_R(v)}(p)$ iff $l \in \mathcal{C}^{\mathsf{box}}(\mathsf{box}_R(v))$ and $p \in \mathsf{inbox}_{\mathsf{box}_R(v)}(l) = \mathsf{inbox}_R(l)$ iff $\mathsf{P}^{\mathsf{pri}}_R(l) \in \mathsf{inbox}_R(v)$ and $l \in \mathsf{boxesof}_R(v)$.

The following lemma shows that the depth of a port p in a ps R is nothing but the number of boxes in R containing p, as in the usual inductive syntaxes of linear logic.

Lemma 61. *Let $R \in \mathbf{PS}$. For every $p \in \mathcal{P}(R)$, one has*

$$\mathsf{depth}_R(p) = \mathsf{card}(\mathsf{boxesof}_R(p))$$

In particular, $\mathsf{depth}_R(p) = 0$ iff $p \notin \mathsf{inbox}_R(v)$ for any $v \in \mathcal{C}^{\mathsf{box}}(R)$.

Proof. By induction on the $\mathsf{depth}(p) \in \mathbb{N}$. We denote by $\varphi_p = (p_i)_{0 \leq i \leq n}$ (with $n \in \mathbb{N}$) the unique box-crossing path from $\mathsf{c}_R(p)$ to p.

If $\mathsf{depth}_R(p) = 0$, then φ_p does not cross any $v \in \mathcal{C}^{\mathsf{prom}}(R) = \mathcal{C}^{\mathsf{box}}(R)$ (as $R \in \mathbf{PS}$) nor any $q \in \mathcal{A}uxdoors(R)$, according to definition 59. Hence, for every $v \in \mathcal{C}^{\mathsf{box}}(R)$, every path above v does not end in p, by condition 4 in definition 46. Thus, $\mathsf{card}(\mathsf{boxesof}_R(p)) = 0 = \mathsf{depth}_R(p)$.

If $\mathsf{depth}_R(p) > 0$, then φ_p crosses some $v \in \mathcal{C}^{\mathsf{box}}(R)$ or some $q \in \mathcal{A}uxdoors(R)$, according to definition 59. Hence, $n > 0$ and for some $v \in \mathcal{C}^{\mathsf{box}}(R)$, there exists a path above v ending in p (i.e. $p \in \mathsf{inbox}_R(v)$), by condition 4 in definition 46. According to definition 58 (i.e. thanks to nesting condition, as $R \in \mathbf{PS}$), $\mathsf{C}^{\mathsf{box}}_R(p)$ is defined. By condition 4 in definition 46 and lemma 57.4, φ_p crosses either $\mathsf{C}^{\mathsf{box}}_R(p)$ or $q \in \mathsf{auxd}_R(\mathsf{C}^{\mathsf{box}}_R(v))$: in both cases, there exists $k \in \{0, \ldots, n\}$ such that $p_k \in \mathsf{doors}_R(\mathsf{C}^{\mathsf{box}}_{\mathsf{R}}(\mathsf{p}))$ and the subpath $(p_i)_{k \leq i \leq n}$ of φ_p does not cross any $v \in \mathcal{C}^{\mathsf{box}}(R)$ nor any $q \in \mathcal{A}uxdoors(R)$, because of the minimality of $\mathsf{inbox}_R(\mathsf{C}^{\mathsf{box}}_R(p))$. So $\mathsf{depth}_R(p_{k-1}) = \mathsf{depth}_R(p) - 1$, hence $\mathsf{depth}_R(p_{k-1}) = \mathsf{card}(\mathsf{boxesof}_R(p_{k-1}))$ by induction hypothesis. Because of lemma 51, $p_{k-1} \notin \mathsf{inbox}_R(\mathsf{C}^{\mathsf{box}}_R(p))$ whereas $p_k \notin \mathsf{inbox}_R(\mathsf{C}^{\mathsf{box}}_R(p))$ and thus $\mathsf{card}(\mathsf{boxesof}_R(p_{k-1})) = \mathsf{card}(\mathsf{boxesof}_R(p)) - 1$, by the nesting condition. Therefore, $\mathsf{depth}_R(p) = \mathsf{card}(\mathsf{boxesof}_R(p))$. □

The following proposition reveals some intuitive properties of some notions already introduced.

Proposition 62. *Let* $R \in \mathbf{PS}$.

1. *If* $\{p,q\} \in \mathcal{W}(R)$ *then* $\mathsf{boxesof}_R(p) = \mathsf{boxesof}_R(q)$ *and* $\mathsf{depth}_R(p) = \mathsf{depth}_R(q)$.
2. *Let* $v \in \mathcal{C}^{\mathsf{box}}(R)$ *and let* p_v *be the unique auxiliary port of* v*: for every* $q \in \mathsf{bc}_R(v)$ *one has* $\mathsf{boxesof}_R(p_v) = \mathsf{boxesof}_R(q)$ *and* $\mathsf{depth}_R(q) = \mathsf{depth}_R(p_v)$.
3. *Let* $p \in \bigcup \mathcal{C}uts(R) \cup \mathcal{D}(R)$*:* $\mathsf{depth}_R(p) = 0$ *iff* $p \in \bigcup \mathcal{C}uts_0(R) \cup \mathcal{D}_0(R)$.
4. *Let* $v, v' \in \mathcal{C}^{\mathsf{box}}(R)$*: if* $\mathsf{inbox}_R(v') \subseteq \mathsf{inbox}_R(v)$*, then* $\mathsf{boxesof}_R(\mathsf{P}^{\mathsf{pri}}_R(v)) \subseteq \mathsf{boxesof}_R(\mathsf{P}^{\mathsf{pri}}_R(v'))$ *and* $\mathsf{depth}_R(\mathsf{box}_R(v)) \leq \mathsf{depth}_R(\mathsf{box}_R(v'))$.
5. *for every* $v \in \mathcal{C}^{\mathsf{box}}(R)$ *one has* $\mathsf{depth}(\mathsf{box}_R(v)) < \mathsf{depth}(R)$.

PROOF.

1. For every $v \in \mathcal{C}^{\mathsf{box}}(R)$, there exists a path above v ending in p iff there exists a path above v ending in q: this is evident when $\{p,q\}$ is neither a cut nor an axiom, this is due to condition 4 in definition 46 if $\{p,q\}$ is an axiom, and this is due to definition of bc_R if $\{p,q\}$ is a cut. Hence, $p \in \mathsf{inbox}_R(v)$ iff $q \in \mathsf{inbox}_R(v)$ and thus $\mathsf{boxesof}_R(p) = \mathsf{boxesof}_R(q)$. Therefore $\mathsf{depth}_R(p) = \mathsf{depth}_R(q)$ by lemma 61.
2. For every $l \in \mathcal{C}^{\mathsf{box}}(R)$, $l \in \mathsf{boxesof}_R(p_v)$ iff $p_v \in \mathsf{inbox}_R(l)$ iff there exists in R a path above l ending in p_v iff (by definition of ascending path) there exists in R a path above l ending in q iff $q \in \mathsf{inbox}_R(l)$ iff $l \in \mathsf{boxesof}_R(q)$. Therefore, $\mathsf{boxesof}_R(p_v) = \mathsf{boxesof}_R(q)$ and so $\mathsf{depth}_R(q) = \mathsf{depth}_R(p_v)$ by lemma 61.
3. If $\mathsf{depth}_R(p) = 0$ then the box-crossing path from $\mathsf{c}_R(p)$ to p crosses no promotion cells, hence there is no ports q such that $q \preceq^1_R p$, in particular $p \notin \mathsf{im}(\mathsf{bc}_R)$ and thus $p \in \mathcal{C}uts_0(R) \cup \mathcal{D}_0(R)$.

   Conversely, if $p \in \mathcal{C}uts_0(R) \cup \mathcal{D}_0(R)$ then $p = \mathsf{c}_R(p)$ by lemma 57.3, therefore $\mathsf{depth}_R(p) = 0$.
4. By remark 50.2, in R there exists a path above v ending in the unique auxiliary port of v'. If $\mathsf{P}^{\mathsf{pri}}_R(v') \notin \mathsf{inbox}_R(v)$ then there is no path above v ending in $\mathsf{P}^{\mathsf{pri}}_R(v')$, hence $v = v'$ and so $\mathsf{P}^{\mathsf{pri}}_R(v) = \mathsf{P}^{\mathsf{pri}}_R(v')$, whence $\mathsf{boxesof}_R(\mathsf{P}^{\mathsf{pri}}_R(v)) = \mathsf{boxesof}_R(\mathsf{P}^{\mathsf{pri}}_R(v'))$. Otherwise in R there exists a path φ above v ending in $\mathsf{P}^{\mathsf{pri}}_R(v')$: if $l \in \mathsf{boxesof}_R(\mathsf{P}^{\mathsf{pri}}_R(v))$ then $l \in \mathcal{C}^{\mathsf{box}}(R)$ and $\mathsf{P}^{\mathsf{pri}}_R(v) \in \mathsf{inbox}_R(l)$, thus there exists a path ψ above l ending in $\mathsf{P}^{\mathsf{pri}}_R(v)$ and so $\psi \cdot \varphi$ is a path in R above l ending in $\mathsf{P}^{\mathsf{pri}}_R(v')$,

whence $\mathsf{P}_R^{\mathsf{pri}}(v') \in \mathsf{inbox}_R(l)$ and thus $l \in \mathsf{boxesof}_R(\mathsf{P}_R^{\mathsf{pri}}(v'))$. Therefore $\mathsf{boxesof}_R(\mathsf{P}_R^{\mathsf{pri}}(v)) = \mathsf{boxesof}_R(\mathsf{P}_R^{\mathsf{pri}}(v'))$, so $\mathsf{depth}_R(\mathsf{box}_R(v)) \leq \mathsf{depth}_R(\mathsf{box}_R(v'))$ by lemma 61.

5. Let $p \in \mathcal{P}(\mathsf{box}_R(v))$ be such that $\mathsf{depth}(\mathsf{box}_R(v)) = \mathsf{depth}_{\mathsf{box}_R(v)}(p)$. If $p \in \mathcal{P}(\mathsf{box}_R(v)) \setminus \mathsf{inbox}_R(v)$ then p is the principal port of a unary ?-cell whose unique auxiliary port $q \in \mathsf{auxd}_R(v)$ and so $q \in \mathsf{inbox}_R(v)$ and $\mathsf{depth}_{\mathsf{box}_R(v)}(p) \leq \mathsf{depth}_{\mathsf{box}_R(v)}(q)$. Therefore, we can suppose without loss of generality that $p \in \mathsf{inbox}_R(v)$ and thus $\mathsf{boxesof}_{\mathsf{box}_R(v)}(p) = \mathsf{boxesof}_R(p) \setminus \{l \in \mathcal{C}^{\mathsf{box}}(R) \mid \mathsf{P}_R^{\mathsf{pri}}(l) \notin \mathsf{inbox}_R(v)\}$ by remark 60.3. As $\mathsf{P}_R^{\mathsf{pri}}(v) \notin \mathsf{inbox}_R(v)$ (by lemma 51), one has $\mathsf{boxesof}_{\mathsf{box}_R(v)}(p) \subsetneq \mathsf{boxesof}_R(p)$, whence $\mathsf{depth}(\mathsf{box}_R(v)) = \mathsf{depth}_{\mathsf{box}_R(v)}(p) < \mathsf{depth}_R(p) \leq \mathsf{depth}(R)$ by lemma 61.

□

According to proposition 62.1, we are entitled to talk about of the depth of a wire $\{p, q\}$ of a ps: it is the depth of p or q. Propositions 62.2-3 mean that in a ps R, for any $v \in \mathcal{C}^{\mathsf{box}}(R)$, the cuts and deadlocks pointed by the arrow function bc_R are the cuts and deadlock at depth 0 in $\mathsf{box}_R(v)$, in other words $\mathsf{box}_R(v)$ is the deepest (i.e. "smallest" in the sense of proposition 62.4) box containing them.

Notice that proposition 62.2 does not hold if we replace the hypothesis $q \in \mathsf{bc}_R(v)$ with $q \in \mathsf{auxd}_R(v)$ because in general, we can have $v, l \in \mathcal{C}^{\mathsf{box}}(R)$ such that $q \in \mathsf{auxd}_R(l) \cap \mathsf{auxd}_R(v)$ but $p_v \notin \mathsf{inbox}_R(l)$ (where p_v is the unique auxiliary port of v), whence $\mathsf{boxesof}_R(q) \not\subseteq \mathsf{boxesof}_R(p_v)$.

Proposition 62.5 allows to make easily induction on the depth of a ps.

1.6 Indexed ((pre-)pre-)proof-structures

We introduce the notion of indexed pseudo-structure (resp. ppps; pps; ps), i.e. a pseudo-structure (resp. ppps; pps; ps) with ordered conclusions. This is mandatory to fix an order on conclusions in order to define the interpretation of a proof-structure in the relational model.

Definition 63 (Indexed ((pre-)pre-)proof-structure)**.** *An* indexed pseudo-structure *is a pair* (Φ, ind) *such that* $\Phi \in \mathbf{PseudoPPPS}$ *and* $\mathsf{ind} : \mathcal{P}^{\mathsf{free}}(\Phi) \to \{1, \ldots, \mathsf{card}(\mathcal{P}^{\mathsf{free}}(\Phi))\}$ *is a bijection. We say then that* ind *is an* enumeration of $\mathcal{P}^{\mathsf{free}}(\Phi)$.

An indexed ppps *(resp.* indexed pps*;* indexed ps*) is an indexed pseudo-structure* (Φ, ind) *such that* $\Phi \in \mathbf{PPPS}$ *(resp.* $\Phi \in \mathbf{PPS}$*;* $\Phi \in \mathbf{PS}$*).*

We denote by $\mathbf{PseudoPPPS}^{\mathsf{ind}}$ *(resp.* $\mathbf{PPPS}^{\mathrm{ind}}$*;* $\mathbf{PPS}^{\mathrm{ind}}$*;* $\mathbf{PS}^{\mathrm{ind}}$*) the set of indexed pseudo-structures (resp. indexed ppps; indexed pps; indexed ps).*

We set $\mathbf{PPS}^{\mathrm{ind}}_{\mathsf{DiLL}_0} = \{(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}} \mid R \in \mathbf{PPS}_{\mathsf{DiLL}_0}\}$ *and* $\mathbf{PS}^{\mathrm{ind}}_{\mathsf{MELL}} = \{(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}} \mid R \in \mathbf{PS}_{\mathsf{MELL}}\}$.

We introduce the notion of "identity" (or better said isomorphism) between two indexed ppps (resp. pps; ps). The idea is that two corresponding conclusions of two indexed ppps (resp. pps; ps) have to be in the same order position.

Definition 64 (Isomorphism between indexed ((pre-)pre-)-proof-structures). *Let* $(\Phi, \mathsf{ind}), (\Phi', \mathsf{ind}') \in \mathbf{PPPS}^{\mathrm{ind}}$ *(resp.* $(\Phi, \mathsf{ind}), (\Phi', \mathsf{ind}) \in \mathbf{PPS}^{\mathrm{ind}}$*;* $(\Phi, \mathsf{ind}), (\Phi', \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}$*).*

An isomorphism from (Φ, ind) to (Φ', ind') *is a* $\varphi : \Phi \simeq \Phi'$ *such that the following diagram commutes:*

$$\begin{tikzcd}
\mathcal{P}^{\mathsf{free}}(\Phi) \arrow[r, "\mathsf{ind}"] \arrow[d, "\varphi_{\mathcal{P}}"'] & \{1, \ldots, \mathsf{card}(\mathcal{P}^{\mathsf{free}}(\Phi))\} \\
\mathcal{P}^{\mathsf{free}}(\Phi') \arrow[ur, "\mathsf{ind}'"'] &
\end{tikzcd}$$

We write then $\varphi : (\Phi, \mathsf{ind}) \simeq (\Phi', \mathsf{ind}')$*.*

If there exists an isomorphism from (Φ, ind) *to* (Φ', ind')*, then we say that* (Φ, ind) and (Φ', ind') are isomorphic *and we write* $(\Phi, \mathsf{ind}) \simeq (\Phi', \mathsf{ind}')$*.*

Remark 65. Let $\Phi, \Phi' \in \mathbf{PPPS}$ with $\varphi : R \simeq R'$. For every enumeration ind of $\mathcal{P}^{\mathsf{free}}(\Phi)$, there exists an enumeration ind' of $\mathcal{P}^{\mathsf{free}}(\Phi')$ such that $\varphi : (\Phi, \mathsf{ind}) \simeq (\Phi', \mathsf{ind}')$. Indeed, it suffices to take $\mathsf{ind}' : \mathcal{P}^{\mathsf{free}}(\Phi') \to \{1, \ldots, \mathsf{card}(\mathcal{P}^{\mathsf{free}}(\Phi'))\}$ such that $\mathsf{ind}'(p) = \mathsf{ind}(\varphi_{\mathcal{P}}^{-1}(p))$.

1.7 A non-inductive correctness criterion

A correctness criterion is a property fulfilled by all and only those proof-structures corresponding to a proof in the (multiplicative and exponential framework of) Linear Logic sequent calculus. This gives a geometrical account of the Linear Logic proofs. There is a multitude of equivalent correctness criteria for the multiplicative and exponential framework of Linear Logic, the most common one is the Danos-Regnier criterion (see for example [DR89, Tor03]), which is a simplification of the primary long trip criterion of Girard introduced in [Gir87].

All the well-known correctness criteria for the multiplicative and exponential framework of Linear Logic proof-structures are defined by induction on the depth of the proof-structure, so they can considered "purely geometrical" only in the case of a proof-structure without boxes (in particular, in the multiplicative framework). In our syntax we can reformulate the Danos-Regnier correctness criterion for the multiplicative and exponential framework of Linear Logic proof-structures in such a way that our criterion is completely "non-inductive", that is reinforces the idea of a "purely geometrical" characterization of (multiplicative and exponential) Linear Logic proofs.

Definition 66 (Snipping, linearization)**.** *Let* $\Phi \in \mathbf{Modules}$ *and let* $Q \subseteq \mathcal{P}^{\mathsf{aux}}(\Phi)$*. The* snipping of Q in Φ *is a* 6*-tuple* $\Phi' = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ *such that:*

- $\mathbb{C}' = (\mathsf{t}_\Phi, \mathcal{P}', \mathsf{C}_\Phi\restriction_{\mathcal{P}'}, \mathsf{P}^{\mathsf{pri}}_\Phi, \mathsf{P}^{\mathsf{left}}\restriction_{L_Q})$ *where* $\mathcal{P}' = \mathcal{P}(\Phi) \smallsetminus Q$ *and* $L_Q = \{l \in \mathcal{C}^{\otimes,\parr}(\Phi) \mid \mathsf{card}(\{p \in \mathcal{P}' \mid \mathsf{C}_\Phi(p) = l\}) = 3\}$*;*
- $\mathcal{I}' = \mathcal{I}(\Phi) \smallsetminus (Q \cap \bigcup \mathcal{A}x(\Phi))$*;*
- $\mathcal{D}' = \mathcal{D}(\Phi)$ *and* $\mathcal{W}' = \mathcal{W}(\Phi)$*;*
- $\mathsf{auxd}' = \emptyset = \mathsf{bc}'$ *(where* $\emptyset$ *is the empty function).*

If $Q = \emptyset$*, we say that the snipping of* Q *in* Φ *is the* linearization of Φ*, denoted by* $\mathsf{lin}(\Phi)$*.*

Remark 67. For every $\Phi \in \mathbf{Modules}$ and $Q \subseteq \mathcal{P}^{\mathsf{aux}}(\Phi)$, if $\Phi' = (\mathbb{C}', \mathcal{I}', \mathcal{D}', \mathcal{W}', \mathsf{auxd}', \mathsf{bc}')$ is the snipping of Q in Φ, then $\mathbb{C}' \in \mathbf{ModuleBases}$ and $\Phi' \in \mathbf{Module}$. Furthermore $\mathcal{C}(\Phi) = \mathcal{C}(\Phi')$, $\mathcal{P}(\Phi) = \mathcal{P}(\Phi')$, $\mathcal{P}^{\mathsf{pri}}(\Phi) = \mathcal{P}^{\mathsf{pri}}(\Phi')$, $\mathcal{D}(\Phi) = \mathcal{D}_0(\Phi')$ and $\mathcal{W}(\Phi) = \mathcal{W}(\Phi')$, but in general $\mathcal{I}(\Phi) \not\supseteq \mathcal{I}(\Phi')$, $\mathcal{P}^{\mathsf{free}}(\Phi) \not\supseteq \mathcal{P}^{\mathsf{free}}(\Phi')$ and $\mathcal{P}^{\mathsf{aux}}(\Phi) \not\subseteq \mathcal{P}^{\mathsf{aux}}(\Phi')$. Therefore, every path in Φ' is also a path in Φ (but the converse does not hold).

If $R \in \mathbf{PS}$, then $\mathsf{lin}(R) \in \mathbf{PPS}_{\mathsf{DiLL}_0}$, $\mathcal{I}(R) = \mathcal{I}(\mathsf{lin}(R))$, $\mathcal{P}^{\mathsf{free}}(R) = \mathcal{P}^{\mathsf{free}}(\mathsf{lin}(R))$ and $\mathcal{P}^{\mathsf{aux}}(R) = \mathcal{P}^{\mathsf{aux}}(\mathsf{lin}(R))$.

Roughly speaking, if Φ is a module and Q is a set of auxiliary ports of Φ, the snipping of Q in Φ is the module obtained from Φ by disconnecting the ports of Q (which become isolated ports) by their cells. This operation might transforms a ps in a module which is not a pseudo-structure, because some binary $\otimes$- or $\parr$-cells might get 0-ary or unary, or some wires might get hanging (i.e. they connect a principal port with an isolated port).

The linearization of a ps Φ has to be seen as the DiLL_0-proof-structure obtained from Φ by forgetting the boundaries of all the boxes of Φ (i.e. their arrows).

Definition 68 (Switching, correctness graph)**.** *Let* $R \in \mathbf{PS}$ *and let*

$$\mathcal{C}^{\parr,?\mathsf{c}}(R) = \mathcal{C}^{\parr}(R) \cup \{l \in \mathcal{C}(R) \mid \mathsf{t}_R(l) = ?,\ \mathsf{a}_R(l) \geq 2\}\,.$$

A switching of R *is a function associating with every* $l \in \mathcal{C}^{\parr,?\mathsf{c}}(R)$ *an auxiliary port of* l*.*

For every switching s *of* R*, we set* $\mathsf{off}_R(s) = \{p \in \mathcal{P}^{\mathsf{aux}}_l(R) \smallsetminus \mathsf{im}(s) \mid l \in \mathcal{C}^{\parr,?\mathsf{c}}(R)\}$*: the* s-correctness graph of Φ *is the snipping of* $\mathsf{off}_R(s)$ *in* R*.*

A correctness graph of R *is a* s*-correctness graph of* R *for some switching* s *of* R*.*

Definition 68 reformulates in our syntax some standard notions of Linear Logic proof-structures.

Remark 69. For every $R \in \mathbf{PS}$ and switching s of R, if G^s_R is the s-correction graph of R then $G^s_R \in$ **Modules**; moreover, if $l \in \mathcal{C}(R)$ is such that there exists a $p \in \mathsf{doors}_R(v) \cap \mathcal{P}^{\mathsf{aux}}_l(R)$ for some $v \in \mathcal{C}^{\mathsf{box}}(R)$, then $\mathsf{a}_{G^s_R}(l) = 1$.

Definition 70 (DR-path)**.** *Let $R \in \mathbf{PS}$ and let s be a switching of R.*

A DR-path in R according to s *is a path $(p_i)_{i\in I}$ (where I is an initial segment of $\mathbb{N}$) in the s-correction graph of R such that for every $v \in \mathcal{C}^{\mathsf{box}}(R)$ and $i, i+1 \in I$, if $p_i \in \mathsf{doors}_R(v) \cap \mathcal{P}^{\mathsf{aux}}_l(R)$ and $p_{i+1} = \mathsf{P}^{\mathsf{pri}}_R(l)$ for some $l \in \mathcal{C}(R)$, then for every $j \in I$ such that $j > i+1$ one has $p_j \notin \mathsf{doors}_R(v)$.*

For every $p, q \in \mathcal{P}(R)$, we say that p and q are DR-connected according to s *if there exists a DR-path in R according to s from p to q.*

The idea is that, given a ps R and a switching s of R, a DR-path in R according to s leaving a box cannot re-enter it. By means of DR-paths we can give a simple correctness criterion.

Definition 71 (DR-connection, DR-acyclicity, proof-net)**.** *Let $R \in \mathbf{PS}$.*

R is DR-connected *if for every switching s of R, all $p, q \in \mathcal{P}(R)$ are DR-connected according to s.*

R is DR-acyclic *if $\mathcal{D}(R) = \emptyset$ and, for every switching s of R, each path in the s-correction graph of R is not a cycle and it is a DR-path in R according to s.*

R is ACC *(or* R is a proof-net *or* R satisfies the correctness criterion*) if R is DR-connected and DR-acyclic.*

We denote by $\mathbf{PN}$ *the set of proof-nets. We set* $\mathbf{PN}_{\mathsf{MELL}} = \mathbf{PN} \cap \mathbf{PS}_{\mathsf{MELL}}$ *(resp.* $\mathbf{PN}_{\mathsf{DiLL}_0} = \mathbf{PN} \cap \mathbf{PPS}_{\mathsf{DiLL}_0}$*), whose elements are the* MELL-proof-nets *(resp.* DiLL_0-proof-nets*).*

We point out that our correctness criterion, as well as our definition of proof-structure, is completely non-inductive, by taking into account only some "geometrical informations" available in a proof-structure. Our correctness criterion can be seen as a non-inductive version of the Danos-Regnier correctness criterion [DR89, Tor03] for the multiplicative and exponential framework of Linear Logic (whic is non-inductive only in the multiplicative fragment). Intuitively, in our correctness criterion DR-paths play the role of induction on the depth of a proof-structure in the Danos-Regnier criterion.

In the case of a DR-connected ps, we can simplify the correctness criterion.

Proposition 72. *Let $R \in \mathbf{PS}$ be DR-connected. R is a proof-net iff $\mathcal{D}(R) = \emptyset$ and, for every switching s of R, the s-correction graph of R is acyclic.*

Proof. Let $R \in \mathbf{PS}$ be DR-connected such that $\mathcal{D}(R) = \emptyset$ and, for every switching s of R, the s-correctness graph of R is acyclic. We have to show that each path in the s-correction graph of R is a DR-path in R according to s. Let us suppose by absurd that for some switching s of R there exists a path in the s-correction graph G^s_R of R which is not a DR-path in R according to s.

Thus, there would exist $v \in \mathcal{C}^{\mathsf{box}}(R)$ and a path $(p_i)_{0 \leq i \leq n}$ (for some $n \in \mathbb{N}^*$) in G^s_R the such that $p_n \in \mathsf{doors}_R(v)$, $p_0 \in \mathsf{doors}_R(v) \cap \mathcal{P}^{\mathsf{aux}}_l(R)$ and $p_1 = \mathsf{P}^{\mathsf{pri}}_R(l)$ for some $l \in \mathcal{C}(R)$ and $p_i \notin \mathsf{doors}_R(v)$ for every $1 \leq i \leq n-1$. Since R is DR-connected, there would exist a DR-path $(q_j)_{0 \leq i \leq m}$ (for some $m \in \mathbb{N}^*$) in R according to s from p_n to p_0, hence $q_1 \in \mathsf{inbox}_R(v)$ (in particular, $q_1 \neq p_{n-1}$) by remark 69 and so $(p_i)_{0 \leq i \leq n} \cdot (q_j)_{1 \leq j \leq m}$ would be a cycle in G^s_R, that is impossible because of acyclicity of G^s_R. ☐

Empires are introduced by Girard in [Gir87] to prove the sequentialization's theorem for Linear Logic. We reformulate it for DiLL_0-ps: the intuition is that empires in DiLL_0-ps enjoys all the good properties of empires in multiplicative fragment of Linear Logic, if we see !-cells (resp. ?-cells) as generalized $\otimes$-links (resp. $⅋$-links).

Definition 73 (Empire). *Let $R \in \mathbf{PPS}_{\mathsf{DiLL}_0}$ and let $p \in \mathcal{P}(R)$.*

*For every switching s of R, let G^s_R be s-correction graph of R: the s-*correction graph of R rooted in *p is either the snipping of $\{p\}$ in G^s_R if $p \in \mathcal{P}^{\mathsf{aux}}(G^s_R)$, or G^s_R otherwise.*

The empire of p in R, *denoted by $\epsilon_R(p)$, is the set of $q \in \mathcal{P}(R)$ such that p and q are connected in all the s-correction graph of R rooted in p, for every switching s of R.*

The boundary of $\epsilon_R(p)$ *is*

$$\partial\epsilon_R(p) = \{q \in \mathcal{P}(R) \mid \exists\, q' \in \mathcal{P}(R) \smallsetminus \epsilon_R(p) \textit{ and } (q, q') \textit{ is a path in } R \textit{ of length } 1\}.$$

Chapter 2

Relational semantics

The general goal of denotational semantics is to give a "mathematical" counterpart to syntactical devices such as proofs and programs, bringing to the fore their essential properties: the basic pattern is to associate with every formula/type an object of some category and with every proof/program a morphism of this category (its interpretation or semantics).

Let us consider the $*$-autonomous category **Rel** of sets and relations: the Kleisli category of the comonad associated with the finite multisets functor on **Rel** is a Cartesian closed category, i.e. a denotational model for λ-calculus. Such an interpretation of λ-terms is the same as the interpretation of the Linear Logic proof-net translating the λ-term in the multiset based relational model of Linear Logic. This holds for both the typed and untyped case.

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 such that $D \Rightarrow D$ (the exponentiation of D) is a retract of D (i.e. there exist two morphisms $\mathsf{abs} : (D \Rightarrow D) \to D$ and $\mathsf{app} : D \to (D \Rightarrow D)$ such that $\mathsf{app} \circ \mathsf{abs} = id_{D \Rightarrow D}$). In the multiplicative and exponential framework of Linear Logic 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 74) in the category **Rel** in such a way that not only $D \times D$ and $\mathscr{M}_{\mathrm{fin}}(D)$ 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 proof-structures invariant under cut-elimination.

Notation. Let X be a set. We denote by $\mathscr{M}(X)$ (resp. $\mathscr{M}_{\mathrm{fin}}(X)$) the set of multisets (resp. finite multisets) on X. For every $m \in \mathscr{M}(X)$, we set $\mathsf{supp}(m) = \{x \in X \mid m(x) \neq 0\}$.

2.1 Relational spaces

We introduce a domain D to interpret (untyped) proof-structures as it is already defined in [dCPT11, dCT12]. All the following definitions are exactly the same as those ones in [dCT12].

In the definition of the domain D the set $\{+,-\}$ of polarities is used in order to "semantically distinguish" cells of dual types $1/\bot$, $\otimes/\parr$ and $!/?$, which is mandatory in an untyped framework.

Definition 74 (Atom, point)**.** *We fix a set A not containing any pair nor any 3-tuple and such that $* \notin A$; we call* atoms *the elements of A.*

We define D_n by induction on $n \in \mathbb{N}$:

- $D_0 = A \cup (\{+,-\} \times \{*\})$,
- $D_{n+1} = D_0 \cup (\{+,-\} \times D_n \times D_n) \cup (\{+,-\} \times \mathscr{M}_{\text{fin}}(D_n))$.

We set $D = \bigcup_{n\in\mathbb{N}} D_n$. The depth of an element $\alpha \in D$ *is the least $n \in \mathbb{N}$ such that $\alpha \in D_n$.*

We set $D^{<\omega} = \bigcup_{n\in\mathbb{N}} D^n$, whose elements are called points.

Remark 75.

1. $D_n \subseteq D_{n+1}$ for every $n \in \mathbb{N}$. The proof is by a straightforward induction on $n \in \mathbb{N}$.

2. Let $\alpha, \beta, \alpha_1, \ldots, \alpha_k \in D$ (for some $k \in \mathbb{N}$), let $\gamma \in A$ and $\iota \in \{+,-\}$:
   - $\mathsf{depth}(\gamma) = 0 = \mathsf{depth}(\iota, *)$, as $D_0 = A \cup (\{+,-\} \times \{*\})$;
   - $\mathsf{depth}(\iota, \alpha, \beta) = \max\{\mathsf{depth}(\alpha), \mathsf{depth}(\beta)\}+1$, indeed if $\mathsf{depth}(\alpha) = n$, $\mathsf{depth}(\beta) = m$ and $d = \mathsf{max}\{n, m\}$ then $(\alpha, \beta) \in D_d \times D_d$ and $(\alpha, \beta) \notin D_i \times D_i$ for any $0 \leq i \leq d-1$, hence $\mathsf{depth}(\iota, \alpha, \beta) = d+1$;
   - $\mathsf{depth}(\iota, [\alpha_1, \ldots, \alpha_k]) = \sup\{\mathsf{depth}(\alpha_i) \mid i \in \{1, \ldots, k\}\}+1$, indeed if $\mathsf{depth}(\alpha_i) = n_i$ for any $i \in \{0, \ldots, k\}$ and $d = \sup\{n_i \mid i \in \{1, \ldots, k\}\}$, then $(\alpha_1, \ldots, \alpha_k) \in D_d^k$ and $(\alpha_1, \ldots, \alpha_k) \notin D_j^k$ for any $0 \leq j \leq d-1$, hence $\mathsf{depth}(\iota, [\alpha_1, \ldots, \alpha_k]) = d+1$.

3. The conditions on A ensure that D satisfies the following equation

$$D = A \uplus (\{+,-\} \times \{*\}) \uplus (\{+,-\} \times D \times D) \uplus (\{+,-\} \times \mathscr{M}_{\text{fin}}(D))$$

   which means that A, $\{+,-\}\times\{*\}$, $\{+,-\}\times D\times D$ and $\{+,-\}\times\mathscr{M}_{\text{fin}}(D)$ are retracts of D.

Thanks to remark 75.2, we can easily define some notions and prove some propositions by induction on the depth of elements of D.

The function $(\,)^\perp$ (which is the semantic version of the linear negation) flips polarities.

Definition 76 (Dual). *We set* $+^\perp = -$ *and* $-^\perp = +$. *We define* $\alpha^\perp$ *for every* $\alpha \in D$, *by induction on* $\mathsf{depth}(\alpha) \in \mathbb{N}$ *as follows (where* $\gamma \in A$, $\alpha, \beta, \alpha_1, \ldots, \alpha_n \in D$ *for some* $n \in \mathbb{N}$, *and* $\iota \in \{+, -\}$*):*

- $\gamma^\perp = \gamma$ *and* $(\iota, *) = (\iota^\perp, *)$;
- $(\iota, \alpha, \beta)^\perp = (\iota^\perp, \alpha^\perp, \beta^\perp)$ *and* $(\iota, [\alpha_1, \ldots, \alpha_n])^\perp = (\iota^\perp, [\alpha_1^\perp, \ldots, \alpha_n^\perp])$.

Definition 77 (Substitution). *A* substitution *is a function* $\sigma : D \to D$ *induced by a function* $\sigma_A : A \to D$ *and defined by induction on the depth of elements of* D, *as follows (where* $\gamma \in A$, $\alpha, \beta, \alpha_1, \ldots, \alpha_n \in D$ *for some* $n \in \mathbb{N}$, *and* $\iota \in \{+, -\}$*):*

- $\sigma(\gamma) = \sigma_A(\gamma)$ *and* $\sigma(\iota, *) = (\iota, *)$;
- $\sigma(\iota, \alpha, \beta) = (\iota, \sigma(\alpha), \sigma(\beta))$;
- $\sigma(\iota, [\alpha_1, \ldots, \alpha_n]) = (\iota, [\sigma(\alpha_1), \ldots, \sigma(\alpha_n)])$.

Let $(\alpha_1, \ldots, \alpha_n) \in D^{<\omega}$ *for some* $n \in \mathbb{N}$*: we set* $\sigma(\alpha_1, \ldots, \alpha_n) = (\sigma(\alpha_1), \ldots, \sigma(\alpha_n))$.

If $\sigma_A : A \to D$ *is a function such that* $\mathsf{im}(\sigma_A) \subseteq A$ *(resp.* σ_A *is a bijection), then we say that the substitution* σ *induced by* σ_A *is* atomic *(resp.* bijective*).*

We denote by $\mathfrak{M}$ *(resp.* $\mathfrak{S}$*) the set of atomic (resp. bijective and atomic) substitutions.*

Intuitively, for every $\alpha, \beta \in D$, if there exists an atomic substitution σ such that $\alpha = \sigma(\beta)$, then α and β have the same "structure", the corresponding multisets have the same cardinality and α can differs from β only for the identification of distinct atoms of β.

Remark 78. By a straightforward induction on $\mathsf{depth}(\alpha) \in \mathbb{N}$, we can prove that $\sigma(\alpha)^\perp = \sigma(\alpha^\perp)$ for every substitution σ and $\alpha \in D$.

Definition 79 (Occurrences of an element of D). *For every* $\alpha \in D$, *we define* $\mathsf{sub}(\alpha) \in \mathscr{M}_{\text{fin}}(D)$ *by induction on* $\mathsf{depth}(\alpha) \in \mathbb{N}$ *as follows:*

- $\mathsf{sub}(\gamma) = [\gamma]$ *if* $\gamma \in A \cup (\{+, -\} \times \{*\})$;
- $\mathsf{sub}(\iota, \alpha, \beta) = [(\iota, \alpha, \beta)] + \mathsf{sub}(\alpha) + \mathsf{sub}(\beta)$;
- $\mathsf{sub}(\iota, [\alpha_1, \ldots, \alpha_n]) = [(\iota, [\alpha_1, \ldots, \alpha_n])] + \sum_{j=1}^n \mathsf{sub}(\alpha_j)$.

For every $n \in \mathbb{N}$ *and* $(\alpha_1, \ldots, \alpha_n) \in D^{<\omega}$, *we set* $\mathsf{sub}(\alpha_1, \ldots, \alpha_n) = \sum_{i=1}^n \mathsf{sub}(\alpha_i)$.

For every $\alpha \in D$ *and* $r \in D^{<\omega}$, *we say that* α occurs in r *if* $\alpha \in \mathsf{supp}(\mathsf{sub}(r))$, *and that* there are exactly m occurrences of α in r *if* $\mathsf{sub}(r)(\alpha) = m$.

In the sequel we need the notion of injective point, k-point of $D^{<\omega}$ for any $k \in \mathbb{N}$, and for $E \subseteq D^{<\omega}$ the notion of E-atomic.

Definition 80 (Injective point, k-point, E-atomic point)**.** *$r \in D^{<\omega}$ is* injective *if for every $\gamma \in A$, either γ does not occur in r or there are exactly* 2 *occurrences of γ in r. For every $E \subseteq D^{<\omega}$, we set $E_{\text{inj}} = \{r \in E \mid r \text{ is injective}\}$.*

Given $k \in \mathbb{N}$, we say that $r \in D^{<\omega}$ is a k-point *if, for every $m \in \mathbb{N}$ and $\alpha_1, \ldots, \alpha_m \in D$ such that $(+, [\alpha_1, \ldots, \alpha_m])$ occurs in r, we have $m = k$.*

Let $E \subseteq D^{<\omega}$. $r \in E$ is E-atomic *if for every $r' \in E$ and every substitution σ such that $\sigma(r') = r$ one has $\sigma(\gamma) \in A$ for every $\gamma \in A$ occurring in r'. We set $E_{\text{at}} = \{r \in E \mid r \text{ is } E\text{-atomic}\}$.*

Once the subset E of $D^{<\omega}$ is fixed, it makes sense for $r \in E$ to say that it is E-atomic: this means that no other element of E is "more atomic" than r. In a typed framework, we would not have to define the notion of E-atomic point, but in our untyped framework we need that: the reason will be explained after definition 92.

Remark 81.

1. Let id be the substitution induced by the identity functions $id_A \colon A \to A$: then $id \in \mathfrak{S}$ and $id(\alpha) = \alpha$ for any $\alpha \in D$ (i.e. id is the identity function on D). Thus, $id(r) = (id(\alpha_1), \ldots, id(\alpha_n)) = r$ for any $r = (\alpha_1, \ldots, \alpha_n) \in D^{<\omega}$ with $n \in \mathbb{N}$ (i.e. id is the identity function on $D^{<\omega}$).

2. Let $\sigma, \tau \in \mathfrak{M}$ (resp. $\sigma, \tau \in \mathfrak{S}$) be induced as in definition 77 respectively by some functions (resp. bijections) $\sigma_A, \tau_A : A \to A$, let $\tau \circ \sigma$ be the function induced as in definition 77 by the functions $\tau_A \circ \sigma_A$: then $\tau \circ \sigma \in \mathfrak{M}$ (resp. $\tau \circ \sigma \in \mathfrak{S}$) and $(\tau \circ \sigma)(\alpha) = \tau(\sigma(\alpha))$ for any $\alpha \in D$ (i.e. $\circ$ is the composition of functions from D to D). Thus $(\tau \circ \sigma)(r) = ((\tau \circ \sigma)(\alpha_1), \ldots, (\tau \circ \sigma)(\alpha_n)) = \tau(\sigma(r))$ for any $r = (\alpha_1, \ldots, \alpha_n) \in D^{<\omega}$ with $n \in \mathbb{N}$ (i.e. $\circ$ is the composition of functions from $D^{<\omega}$ to $D^{<\omega}$).

3. Let $\sigma \in \mathfrak{S}$ be induced as in definition 77 by some bijection $\sigma_A : A \to A$, let σ^{-1} be the function induced as in the definition 77 by the inverse σ_A^{-1} of σ_A: then $\sigma^{-1} \in \mathfrak{S}$ and $\sigma^{-1}(\sigma(\alpha)) = \alpha = \sigma(\sigma^{-1}(\alpha))$ for any $\alpha \in D$ (i.e. σ is a bijection from D to D and σ^{-1} is the inverse function of σ). Thus $\sigma^{-1}(\sigma(r)) = (\sigma^{-1}(\sigma(\alpha_1)), \ldots, \sigma^{-1}(\sigma(\alpha_n))) = r = \sigma(\sigma^{-1}(r))$ for any $r = (\alpha_1, \ldots, \alpha_n) \in D^{<\omega}$ with $n \in \mathbb{N}$ (i.e. σ is a bijection from $D^{<\omega}$ to $D^{<\omega}$ and σ^{-1} is the inverse function of σ).

Proposition 82.

1. *$\mathfrak{M}$ is a monoid where the binary operation is the composition and the identity is $id \in \mathfrak{M}$.*

2. *$\mathfrak{S}$ is a group where the binary operation is the composition, the identity is id $\in \mathfrak{S}$, the inverse of an injective substitution σ is σ^{-1}.*

PROOF.

1. By remarks 81.1 and 81.2, because the composition of function is associative.
2. By proposition 82.1 and by remark 81.3.

$\square$

Lemma 83. *Let $\sigma \in \mathfrak{M}$.*

- *If $\alpha \in D$ and $x \in A$, then $\mathsf{sub}(\sigma(\alpha))(x) = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha)(x')$ where σ_A^{-1} is the inverse image of the function σ_A inducing σ.*
- *If $r \in D^{<\omega}$ and $x \in A$, then $\mathsf{sub}(\sigma(r))(x) = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(r)(x')$ where σ_A^{-1} is the inverse image of the function σ_A inducing σ.*

PROOF. The second part of this lemma is an immediate consequence of the first one. The proof of the first statement is by induction on $\mathsf{depth}(\alpha)$. Let $\iota \in \{+, -\}$.

- If $\alpha \in A$: $\sigma(\alpha) = \sigma_A(\alpha) \in A$ and so:
  - if $\sigma_A(\alpha) = x$ then $\mathsf{sub}(\sigma(\alpha))(x) = 1 = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha)(x')$ because $\mathsf{card}(\mathsf{supp}(\mathsf{sub}(\alpha))) = 1$,
  - otherwise $\mathsf{sub}(\sigma(\alpha))(x) = 0 = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha)(x')$.
- If $\alpha = (\iota, *)$ then $\sigma(\alpha) = (\iota, *)$, hence $\mathsf{sub}(\sigma(\alpha))(x) = 0 = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(a)(x')$, since $* \notin A$.
- if $\alpha = (\iota, \beta, \gamma)$ where $\beta, \gamma \in D$, then $\sigma(\alpha) = (\iota, \sigma(\beta), \sigma(\gamma))$. By induction hypothesis, $\mathsf{sub}(\sigma(\beta))(x) = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\beta)(x')$ and $\mathsf{sub}(\sigma(\gamma))(x) = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\gamma)(x')$, so $\mathsf{sub}(\sigma(\alpha))(x) = \mathsf{sub}(\sigma(\beta))(x) + \mathsf{sub}(\sigma(\gamma))(x) = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\beta)(x') + \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\gamma)(x') = \sum_{x' \in \sigma_A^{-1}(x)} (\mathsf{sub}(\beta)(x') + \mathsf{sub}(\gamma)(x')) = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha)(x')$.
- if $\alpha = (\iota, [\alpha_1, \ldots, \alpha_n]$ where $n \in \mathbb{N}$ and $\alpha_i \in D$ for any $1 \leq i \leq n$, then $\sigma(\alpha) = (\iota, [\sigma(\alpha_1), \ldots, \sigma(\alpha_n)])$. By induction hypothesis, $\mathsf{sub}(\sigma(\alpha_i))(x) = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha_i)(x')$ for any $1 \leq i \leq n$, thus $\mathsf{sub}(\sigma(\alpha))(x) = \sum_{i=1}^{n} \mathsf{sub}(\sigma(\alpha_i))(x) = \sum_{i=1}^{n} \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha_i)(x') = \sum_{x' \in \sigma_A^{-1}(x)} \sum_{i=1}^{n} \mathsf{sub}(\alpha_i)(x') = \sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha)(x')$.

$\square$

With reference to notation of lemma 83, notice that in the sum $\sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha)(x')$ there are only finitely many nonzero summands (and so $\sum_{x' \in \sigma_A^{-1}(x)} \mathsf{sub}(\alpha)(x')$ is a finite number) because $\mathsf{supp}(\mathsf{sub}(\alpha)) \cap A$ is a finite set.

Corollary 84. *Let $\sigma \in \mathfrak{S}$. If $r \in D^{<\omega}$ and $x \in A$, then $\mathsf{sub}(\sigma(r))(x) = \mathsf{sub}(r)(\sigma_A^{-1}(x))$ where σ_A^{-1} is the inverse of the function σ_A inducing σ. In particular, $r \in D_{\text{inj}}^{<\omega}$ iff $\sigma(r) \in D_{\text{inj}}^{<\omega}$.*

Proof. As $\sigma \in \mathfrak{S}$, σ_A is invertible and so $\sigma_A^{-1}(x)$ is a singleton if σ_A^{-1} is the inverse image of the function σ_A, hence by lemma 83 $\mathsf{sub}(\sigma(r))(x) = \sum_{x' \in \sigma_X^{-1}(x)} \mathsf{sub}(r)(x') = \mathsf{sub}(r)(\sigma_A^{-1}(x))$ where σ_A^{-1} in the rightmost side of the equality is the inverse of the function σ_A. □

Proposition 85.

1. *The binary operation $\bullet : \mathfrak{M} \times D^{<\omega} \to D^{<\omega}$ defined by $\bullet(\sigma, r) := \sigma(r)$ is an action of the monoid $\mathfrak{M}$ on the set $D^{<\omega}$. In particular, the binary relation $\precsim$ on $D^{<\omega}$ defined by:*

$$\text{for any } r, s \in D^{<\omega}, \quad r \precsim s \quad \text{iff} \quad \exists\, \sigma \in \mathfrak{M} : r = \sigma(s)$$

is a preorder relation.

2. *The binary operation $\bullet : \mathfrak{S} \times D_{\text{inj}}^{<\omega} \to D_{\text{inj}}^{<\omega}$ (resp. $\bullet : \mathfrak{S} \times D^{<\omega} \to D^{<\omega}$) defined by $\bullet(\sigma, r) := \sigma(r)$ is an action of the group $\mathfrak{S}$ on the set $D_{\text{inj}}^{<\omega}$ (resp. $D^{<\omega}$). In particular, the binary relation $\sim$ on $D_{\text{inj}}^{<\omega}$ (resp. $D^{<\omega}$) defined by:*

$$\text{for any } r, s \in D_{\text{inj}}^{<\omega} \text{ (resp. } r, s \in D^{<\omega}\text{)}, \quad r \sim s \quad \text{iff} \quad \exists\, \sigma \in \mathfrak{S} : r = \sigma(s)$$

is an equivalence relation.

Proof. By proposition 82, it is sufficient to show that the axioms of the action of a monoid or group on a set hold (in the case where the domain of $\bullet$ is $\mathfrak{S} \times D_{\text{inj}}^{<\omega}$, corollary 84 ensures that the codomain of $\bullet$ is $D_{\text{inj}}^{<\omega}$). Let $r \in D^{<\omega}$:

- if $id : D \to D$ is the identity substitution, induced by the identity function $id_A : A \to A$, then $id \bullet r = id(r) = r$ (see remark 81.1);
- if $\sigma, \tau \in \mathfrak{S}$ then $\sigma \bullet (\tau \bullet r) = \sigma(\tau(r)) = (\sigma \circ \tau) \bullet r$ (see remark 81.2). □

Remark 86. The binary operation $\bullet : \mathfrak{M} \times D^{<\omega} \to D^{<\omega}$ is not stable for the injective points. For instance, let $x, x' \in A$ be such that $x \neq x'$, let $r = ((+, x, x'), (-, x, x'))$, let σ be the substitution induced as in definition 77 by

$$\sigma_A(z) \quad = \quad \begin{cases} x & \text{if } z = x' \\ z & \text{otherwise} \end{cases}$$

then $r \in D_{\text{inj}}^{<\omega}$ but $\sigma \bullet r = \sigma(r) = ((+, x, x), (-, x, x)) \notin D_{\text{inj}}^{<\omega}$.

Roughly speaking, for every $r, s \in D^{<\omega}$, $r \sim s$ means that r and s are equal up to a "coherent renaming" of atoms, where coherent means that in r we can neither identify distinct atoms of s nor diversify identical atoms of s.

Remark 87. Let $r, r' \in D^{<\omega}$ and $r \sim r'$: if r is an injective point then r' is an injective point (by corollary 84).

2.2 Experiments

Like in [Tor03, dCPT11, dCT12], we use experiments, introduced by Girard in [Gir87] to compute the interpretation of a proof-net in the coherent and relational semantics and deeply studied by Tortora de Falco in [Tor00, Tor03]. An experiment (definition 88) can be thought as objects between syntax and semantics allowing to associate with every ps R a point of $D^{<\omega}$ (called result of the experiment, see definition 89) which is an element of the interpretation of R in the relational semantics. The interpretation of R in the relational semantics is the set of results of all the experiments of R (definition 90). Experiments are deeply related to non-idempotent intersection types and their derivations in the λ-calculus (see [dC07, dC09, Ehr12]): an experiment corresponds to a type derivation and the result of an experiment corresponds to a type. The intersection types system considered in [dC07, dC09, Ehr12] lacks idempotency and this corresponds to the fact that we use multisets for interpreting exponentials and not sets as in the set based coherent semantics introduced by Girard in [Gir87].

The definition of experiment of a ps (the same as that one in [dCT12]) is inductive and it uses the nesting condition.

Definition 88 (Experiment)**.** *Let $R \in \mathbf{PS}$. An* experiment e of R*, denoted by $e : R$, is a function associating with every $p \in \mathcal{P}(R)$ an $\alpha \in \mathscr{M}_{\mathrm{fin}}(D)$ and with every $v \in \mathcal{C}^{\mathsf{box}}(R)$ a finite multiset of finite multisets of experiments of $\mathsf{box}_R(v)$. The definition is by induction on $\mathsf{depth}(R) \in \mathbb{N}$, and we ask that $\mathsf{card}(e(v)) = 1$ for every $v \in \mathcal{C}^{\mathsf{box}}(R)$ such that $\mathsf{depth}_R(v) = 0$, and $\mathsf{card}(e(p)) = 1$ for every $p \in \mathcal{P}(R)$ such that $\mathsf{depth}_R(p) = 0$. Furthermore the following conditions are to be fulfilled.*

1. *For every $\{p, q\} \in \mathcal{W}(R)$ such that $\mathsf{depth}_R(p) = 0 = \mathsf{depth}_R(q)$:*
   - *if $\{p, q\} \in \mathcal{A}x(R) \cup \mathcal{C}uts(R)$, $e(p) = [\alpha]$ and $e(q) = [\beta]$, then $\alpha = \beta^{\perp}$;*
   - *if $\{p, q\} \in \mathcal{W}(R) \setminus (\mathcal{A}x(R) \cup \mathcal{C}uts(R))$, then $e(p) = e(q)$.*
2. *For every $l \in \mathcal{C}(R)$ such that $\mathsf{depth}_R(\mathsf{P}_R^{\mathsf{pri}}(l)) = 0$:*
   - *if $l \in \mathcal{C}^{\otimes}(R)$ (resp. $l \in \mathcal{C}^{\parr}(R)$), $e(\mathsf{P}_R^{\mathsf{left}}(l)) = [\alpha]$ and $e(\mathsf{P}_R^{\mathsf{right}}(l)) = [\beta]$, then $e(\mathsf{P}_R^{\mathsf{pri}}(l)) = [(+, \alpha, \beta)]$ (resp. $e(\mathsf{P}_R^{\mathsf{pri}}(l)) = [(-, \alpha, \beta)]$);*

- *if* $l \in \mathcal{C}^1(R)$ *(resp.* $l \in \mathcal{C}^\perp(R)$*), then* $e(\mathsf{P}^{\mathsf{pri}}_R(l)) = [(+,*)],)$ *(resp.* $e(\mathsf{P}^{\mathsf{pri}}_R(l)) = [(-,*)]$*);*
- *if* $l \in \mathcal{C}^?(R)$*, then* $e(\mathsf{P}^{\mathsf{pri}}_R(l)) = [(-, \sum_{p \in \mathcal{P}^{\mathsf{aux}}_l(R)} e(p))]$*.*

3. *For every* $v \in \mathcal{C}^{\mathsf{box}}(R)$ *such that* $\mathsf{depth}_R(\mathsf{box}_R(v)) = 0$*, let* $e(v) = [[e_1, \dots, e_{n_v}]]$*:*
   - *if* p *is the conclusion*[1] *of* $\mathsf{box}_R(v)$ *such that* $\mathsf{P}^{\mathsf{pri}}_R(v) <_R p$*, then* $e(\mathsf{P}^{\mathsf{pri}}_R(l)) = [(+, \sum_{i=1}^{n_v} e_i(p))]$*;*
   - *if* $w \in \mathcal{C}^{\mathsf{box}}(\mathsf{box}_R(v))$*, then* $e(w) = \sum_{i=1}^{n_v} e_i(w)$*;*[2]
   - *if* $p \in \mathsf{inbox}_R(v)$*, then* $e(p) = \sum_{i=1}^{n_v} e_i(p)$*.*[3]

Let $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}$*. An* experiment of (R, ind) *is an experiment of* R*.*

Given a ps R, experiments of R are functions defined on R allowing to compute the interpretation of R pointwise. Indeed, for every experiment e of R, the labels associated by e with the conclusions of R form a tuple called the result of e representing which is a point of $D^{<\omega}$, so the result of an experiment is a truly semantic object. The set of results of all the experiments of R is the interpretation of R in the (muliset based) relational semantics.

Definition 89 (Result of an experiment)**.** *Let* $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}$ *with* $n = \mathsf{card}(\mathcal{P}^{\mathsf{free}}(R))$ *and let* e *be an experiment of* R*. The* result of e in (R, ind) *is* $|e|_{\mathsf{ind}} = (\alpha_1, \dots, \alpha_n) \in D^n$ *such that* α_i *is the unique element of the multiset* $e(\mathsf{ind}^{-1}(i))$*, for every* $i \in \{1, \dots, n\}$*.*

Definition 90 (Interpretation of a proof-structure)**.** *Let* $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}$ *and let* $n = \mathsf{card}(\mathcal{P}^{\mathsf{free}}(R))$*. The* interpretation of (R, ind) *is*

$$[\![(R, \mathsf{ind})]\!] = \{|e|_{\mathsf{ind}} \in D^n \mid e \textit{ is some experiment of } R\}.$$

The crucial result proven in [Gir87] is that if π' is a proof-net obtained by applying to the proof-net π some steps of cut-elimination, then $[\![\pi]\!] = [\![\pi]\!]$ (invariance under cut-elimination). This result is extended in [dCPT11] to untyped nets, moreover it holds also for differential nets. Therefore, relational semantics is a denotational model for **PS**.

Experiments are defined for whatever ps, including DiLL_0-ps. Any denotational semantics of DiLL_0-ps provides a semantics for MELL-ps through the Taylor expansion. This is what the following proposition says in the case of relational semantics.

Proposition 91. *For every* $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}$*, one has* $[\![(R, \mathsf{ind})]\!] = \bigcup_{\rho \in R^*} [\![(\rho, \mathsf{ind})]\!]$*.*

---

[1] See remark 50.1.

[2] This is well defined thanks to the nesting condition and because each promotion cell in $\mathsf{box}_R(v)$ is a cell of R by proposition 49, since $\mathsf{t}_R(l) = ?$ for every $l \in \mathcal{L}_0$.

[3] This is well defined thanks to the nesting condition.

Proof. By straightforward induction on $\mathsf{card}(\mathcal{C}(R)) \in \mathbb{N}$. $\square$

Definition 92 (Atomic experiment)**.** *Let $R \in \mathbf{PS}$. An experiment e of R is* atomic *if for every $p \in \bigcup \mathcal{A}x(R)$, one has $e(p) \in \mathscr{M}_{\text{fin}}(A)$.*

In our untyped framework we need to restrict the set E of all results of all experiments of a ps to the set of the results of the atomic experiments of this ps, in order to avoid the problem of "infinite η-expansions" which are semantically "invisible". Of course, a given point of $D^{<\omega}$ can be the result of an atomic experiment of a ps and the result of a non-atomic experiment of another ps. However, given $(R, \mathsf{ind}) \in \mathbf{PS}^{\text{ind}}$, it makes sense for $r \in [\![(R, \mathsf{ind})]\!]$ to say that it is $[\![(R, \mathsf{ind})]\!]$-atomic: this means that no other element of $[\![(R, \mathsf{ind})]\!]$ is "more atomic" than r.

Lemma 93. *For every $(R, \mathsf{ind}) \in \mathbf{PS}^{\text{ind}}$ cut-free, one has*

$$[\![(R, \mathsf{ind})]\!]_{\mathsf{at}} = \{|e|_{\mathsf{ind}} \mid e \textit{ is an atomic experiment of } R\}\,.$$

Proof. By straightforward induction on $\mathsf{card}(\mathcal{C}(R)) \in \mathbb{N}$. $\square$

Therefore, the set of results of atomic experiment of a cut-free ps R is the atomic restriction of the interpretation of R. The inclusion $\subseteq$ does not hold if R is not cut-free.

2.3 The relationship between Taylor expansion and relational semantics

A MELL-ps contains additional information with respect to a DiLL_0-ps, namely those carried by boxes (the arrows functions auxd and bc). However, these additional informations can be accommodated in $\mathbf{PS}_{\mathsf{DiLL}}$ thanks to the Taylorexpansion of a MELL-ps, which is the re-formulation in terms of nets of the usual Taylor expansion of analytic functions around the origin ([Ehr05]).

The Taylor expansion R^* of $R \in \mathbf{PS}_{\mathsf{MELL}}$ is a set of DiLL_0-ps having the same conclusions as R and such that (the definition is by induction on $\mathsf{depth}(R)$) if $\mathsf{depth}(R) = 0$ then $R^* = \{R\}$, otherwise if $B_1, \ldots, B_n$ are the boxes of R at depth 0 then R^* is the set of the DiLL_0-nets obtained by taking, for every $1 \leq i \leq n$, k_i DiLL_0-ps of B_i^*, for any $k_i \in \mathbb{N}$. We does not give a formal definition of Taylor expansion in our syntax because it would be very complicate. We still refer to the definition given in [MP07] (definition 9).

Given $R \in \mathbf{PS}_{\mathsf{MELL}}$ and $\Phi \in R^*$, it is easy to define a function $\tau_{R,\Phi}$ which associates with every ports (resp. cell) of R the set corresponding ports (resp. cell) of Φ. Obviously, if $p \in \mathcal{P}(R)$ (resp. $l \in \mathcal{C}(R)$) has depth 0 in R then $\tau_{R,\Phi}(p)$ (resp. $\tau_{R,\phi}(l)$) is a singleton.

In the intuition of many specialists, (a result of) an experiment of a MELL-ps R is seen as a DiLL_0-ps in the Taylor expansion R^* of R, and the interpretation of R in the relational semantics is seen as R^*. But this relationship between Taylor expansion and relational semantics has been never formulated precisely. This is what we aim at doing here. Quite surprisingly, the relationship between a result of an experiment of a MELL-ps R and a DiLL_0-ps in R^* can be stated in the expected intuitive way only when R is cut-free. This is due to the fact that two distinct DiLL_0-ps in the Taylor expansion R^* of a MELL-ps R (with cuts) might have the same normal form.

Definition 94 (From points to pseudo-structures)**.** *Let $\alpha \in D$. We define by induction on $\mathsf{depth}(\alpha)$ a pair $(\widetilde{\alpha}, ax_\alpha)$ such that $\widetilde{\alpha}$ is a pseudo-structure having only one conclusion (denoted by $\mathsf{c}(\alpha)$) and ax_α is a function associating with every $l \in \mathcal{C}^{ax}(\widetilde{\alpha})$ some $\gamma \in A$, called the* label of l *as follows:*

- *if $\alpha \in A$ then $\widetilde{\alpha} = (\mathbb{C}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ where $\mathbb{C}$ is the pseudo-cell-base consisting only of a ax-cell l, and $ax_\alpha(l) = \alpha$;*
- *if $\alpha = (+, *)$ (resp. $\alpha = (-, *)$) then $\widetilde{\alpha} = (\mathbb{C}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ where $\mathbb{C}$ is the pseudo-cell-base consisting only of a 1-cell (resp. $\perp$-cell), and $ax_\alpha = \emptyset$ (the empty function);*
- *if $\alpha = (+, \alpha_1, \alpha_2)$ (resp. $\alpha = (-, \alpha_1, \alpha_2)$) then $\widetilde{\alpha} = (\mathbb{C}, \emptyset, \emptyset, \mathcal{W}(\widetilde{\alpha_1}) \uplus \mathcal{W}(\widetilde{\alpha_2}) \uplus \{w_1, w_2\}, \emptyset, \emptyset)$, where $\mathbb{C}$ is the pseudo-cell-base consisting of the disjoint union of $\mathbb{C}(\widetilde{\alpha_1})$, $\mathbb{C}(\widetilde{\alpha_2})$ and a $\otimes$-cell (resp. $⅋$-cell) l, and $w_1 = \{\mathsf{c}(\alpha_1), \mathsf{P}^{\mathsf{left}}_{\mathbb{C}}(l)\}$ and $w_2 = \{\mathsf{c}(\alpha_2), \mathsf{P}^{\mathsf{right}}_{\mathbb{C}}(l)\}$; moreover $ax_\alpha = ax_{\alpha_1} \uplus ax_{\alpha_2}$;*
- *if $\alpha = (+, [\alpha_1, \ldots, \alpha_n])$ (resp. $\alpha = (-, [\alpha_1, \ldots, \alpha_n])$) with $n \in \mathbb{N}$, then $\widetilde{\alpha} = (\mathbb{C}, \emptyset, \emptyset, \biguplus_{0 \leq i \leq n} \mathcal{W}(\widetilde{\alpha_i}) \uplus \{w_1, \ldots, w_n\}, \emptyset, \emptyset)$, where $\mathbb{C}$ is the pseudo-cell-base consisting of the disjoint union of $\mathbb{C}(\widetilde{\alpha_1}), \ldots, \mathbb{C}(\widetilde{\alpha_n})$ and a n-ary $!$-cell (resp. $?$-cell) l with $\mathcal{P}^{\mathsf{aux}}_l(\mathbb{C}) = \{p_1, \ldots, p_n\}$, and $w_i = \{\mathsf{c}(\alpha_i), p_i\}$ for every $1 \leq i \leq n$; moreover $ax_\alpha = \biguplus_{i=1}^n ax_{\alpha_i}$.*

For every $n \in \mathbb{N}$ and $r = (\alpha_1, \ldots, \alpha_n) \in D^n$, we set $\widetilde{r} = \biguplus_{i=1}^n \widetilde{\alpha_i}$ and $ax_r = \biguplus_{i=1}^n ax_{\alpha_i}$, and we define the function $\mathsf{ind}_r : \{\mathsf{c}(\alpha_1), \ldots, \mathsf{c}(\alpha_n)\} \to \{1, \ldots, n\}$ by $\mathsf{ind}_r(\mathsf{c}(\alpha_i)) = i$ for every $1 \leq i \leq n$.

Let $r \in D^{<\omega}$. We denote by $\mathsf{pseudo}(r)$ the set defined as follows: $\Phi \in \mathsf{pseudo}(r)$ iff Φ is obtained from $\widetilde{r}$ by connecting $(l_1, l'_1), \ldots, (l_n, l'_n)$ (for some $n \in \mathbb{N}$) where $l_1, l'_1, \ldots, l_n, l'_n$ are pairwise distinct ax-cells of $\widetilde{r}$ such that $ax_r(l_i) = ax_r(l'_i)$.

Remark 95. Let $r \in D^{<\omega}$.

$\widetilde{r}$ (resp. $(\widetilde{r}, \mathsf{ind}_r)$) is a cut-free and deadlock-free pseudo-structure (resp. indexed pseudo-structure) such that $\mathcal{I}(\Phi) = \emptyset$ and $\mathcal{C}^{\mathsf{prom}}(\Phi) = \emptyset$.

If r is injective then there exists exactly one $\Phi \in \mathsf{pseudo}(r)$ such that $\Phi \in \mathbf{PPS}_{\mathsf{DiLL}_0}$: this is the $\Phi \in \mathsf{pseudo}(r)$ obtained from $\widetilde{r}$ by connecting all the pairs of distinct ax-cells with the same label in $\widetilde{r}$ (there is exactly one way to do that because of the injectivity of r). We denote such a $\Phi \in \mathsf{pseudo}(r)$ by $\overline{r}$. Clearly, $(\overline{r}, \mathsf{ind}_r) \in \mathbf{PPS}^{\mathrm{ind}}_{\mathsf{DiLL}_0}$.

The following proposition says that, for every injective point r, $\overline{r}$ can be seen as a sort of canonical representative of the equivalence class of r modulo $\sim$ (i.e. modulo a coherent renaming of atoms occurring in r).

Proposition 96. *Let $r, r' \in D^{<\omega}_{\mathrm{inj}}$. If $r \sim r'$ then $\overline{r} \simeq \overline{r'}$.*

Proof. Evident. ☐

Lemma 97. *Let $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}$ be cut-free and deadlock-free. For every $r \in [\![(R, \mathsf{ind})]\!]_{\mathrm{inj,at}}$, if r is a 1-point then $(\overline{r}, \mathsf{ind}_r) \simeq (\mathsf{lin}(R), \mathsf{ind})$.*

Proof. This is a reformulation in our syntax of theorem 2.4.8 of [Tor03]. ☐

Roughly speaking, lemma 97 says that, given $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}$, an injective and atomic 1-point in the interpretation of (R, ind) is the same as (R, ind) but the arrows of R.

Lemma 98. *Let $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}_{\mathsf{MELL}}$ be cut-free and deadlock-free. For every $\rho \in (R, \mathsf{ind})^*$ and for every atomic experiment e of ρ whose result is r, if r is injective then $(\overline{r}, \mathsf{ind}_r) \simeq \rho$.*

Proof. By induction on $\mathsf{card}(\mathcal{C}(R))$. ☐

Lemma 99. *Let $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}_{\mathsf{MELL}}$ be cut-free and deadlock-free. For every $r \in [\![(R, \mathsf{ind})]\!]_{\mathrm{at,inj}}$, one has $(\overline{r}, \mathsf{ind}_r) \in (R, \mathsf{ind})^*$.*

Proof. By induction on $\mathsf{card}(\mathcal{C}(R))$. ☐

Theorem 100. *For every $(R, \mathsf{ind}) \in \mathbf{PS}^{\mathrm{ind}}_{\mathsf{MELL}}$, If R is cut-free and deadlock-free then*

$$(R, \mathsf{ind})^* = \{(\overline{r}, \mathsf{ind}_r) \mid r \in [\![(R, \mathsf{ind})]\!]_{\mathrm{at,inj}}\}.$$

Proof. Immediate consequence of lemmas 98 and 99. ☐

Theorem 100 and proposition 96 say that the Taylor expansion of a cut-free and deadlock-free MELL-ps R is the set of canonical representatives of equivalence classes modulo $\sim$ of injective points (which are results of atomic experiments) in the interpretation (for relational semantics) of R.

It remains to be investigated the meaning (with respect to the relational semantics) of a DiLL_0-ps ρ in the Taylor expansion of a MELL-ps R with cuts. In this case the difficulty is that ρ can reduce itself in several cut-free and deadlock-free DiLL_0-ps, i.e. ρ corresponds to several injective points of the interpretation of R.

2.4 Injectivity of relational semantics through Taylor expansion

Separation is an important mathematical property, and several theorems are often referred to as "separation theorems". In theoretical computer science, one of the most well-known examples of separation theorem is Böhm's theorem ([Böh68]) for pure λ-calculus: if t, t' are two distinct closed $\beta\eta$-normal terms, then there exists a context $C[\ ]$ such that $C[t]\ \beta^*\ 0$ and $C[t']\ \beta^*\ 1$.

In Linear Logic, a semantic approach to the question of separation is developed by Tortora de Falco in [Tor00], [Tor03] and [dCT12], where the (very natural) question of "injectivity" of the semantics is addressed: do the equivalence relation on proofs defined by the cut-elimination procedure and the one defined by a given denotational model (sometimes/always) coincide? When the answer is positive one says that the model is injective (it separates syntactically different proofs). Indeed, two proofs are "syntactically equivalent" when (roughly speaking) they have the same cut-free form (in a confluent and weakly normalizing system), and they are "semantically equivalent" in a given denotational model when they have the same interpretation.

The works [Tor00] and [Tor03] give partial results and counterexamples to the question of injectivity, mainly for the (multiset based) coherent model: in particular the counterexamples show that this model is not injective for MELL. Also, it was conjectured that the (multiset based) relational model is injective for MELL. Proofs of injectivity of the relational model is given for some fragment of MELL. The most interesting result is obtained by de Carvalho and Tortora de Falco in [dCT12]: for MELL without weakenings (and without the multiplicative unit $\bot$) relational semantics is injective.

In this section we show that relational semantics is injective for MELL-proof-nets, where a MELL-proof-net is a MELL-ps satisfying the ACC correctness criterion (see definition 71). The proof intuitively simplify that one in [dCT12] for the connected case: it is based on the analysis of the Taylor expansion showed in the previous section which gives a "geometrical account" of the interpretation of a ps. This allows, for example, to use some "geometrical tools" as the notion of empires (definition 73).

We give a notion of measure of a ps.

Definition 101 (*aux*-measure)**.** *Let* $R \in \mathbf{PS}$ *and let* $m^?_R$ *be a multiset on* $\mathbb{N}$ *defined by* $m^?_R(i) = \mathsf{card}(\{l \in \mathcal{C}^?(R) \mid \mathsf{a}_R(l) = i\})$ *for every* $i \in \mathbb{N}$*. The* *aux*-measure *of* R *is* $\#(R) = (m^?_R, \mathsf{card}(\mathcal{C}(R)), \mathsf{card}(\mathcal{P}(R)))$*.*

Remark 102.

1. Given $R \in \mathbf{PS}$, $m^?_R$ is a finite multiset, since $\mathcal{C}^?(R)$ is a finite set.

2. For every $R, R' \in \mathbf{PS}$, we can establish an order relation between $m^?_R$

and $m^?_{R'}$, given by the usual multiset order:

$$m^?_R \leq m^?_{R'} \Leftrightarrow \text{for any } i \in \mathbb{N}, \text{ if } m^?_R(i) > m^?_{R'}(i)$$
$$\text{then there is } j > i \text{ such that } m^?_R(j) < m^?_{R'}(j)$$

Definition 103 (n-DiLL_0-ps of a MELL-ps)**.** *Let $R \in \mathbf{PS}_{\mathsf{MELL}}$, let $\Phi \in R^*$ and $n \in \mathbb{N}$.*

Φ is the n-DiLL_0-ps of R *if for every $l \in \mathcal{C}^!(\Phi)$, one has $\mathsf{a}_\Phi(l) = n$.*

Intuitively, given $R \in \mathbf{PS}$ and $n \in \mathbb{N}$, the n-DiLL_0-ps of R is the element of R^* obtained from R by taking, for every promotion cell v of R, n copies of the box of v, starting from the deepest promotion cell of R.

Remark 104. Let $R \in \mathbf{PS}_{\mathsf{MELL}}$.

1. $\mathsf{lin}(R)$ is the 1-DiLL_0-ps of R.

2. If Φ is the n-DiLL_0-ps of R for some $n \in \mathbb{N}$ then, for every $l \in \mathcal{C}(\Phi) \cup \mathcal{P}(\Phi)$, one has $\mathsf{card}(\tau_{R,\Phi}(l)) = n^{\mathsf{depth}_R(l)}$;[4]

3. If Φ is the n-DiLL_0-ps of R for some $n \in \mathbb{N}$ and if $v \in \mathcal{C}^{\mathsf{box}}(R)$ then, for every $u \in \tau_{R,\Phi}(v)$ and $p, p' \in \mathcal{P}^{\mathsf{aux}}_u(\Phi)$, one has $\mathsf{module}(\epsilon_\Phi(p)) \simeq \mathsf{module}(\epsilon_\Phi(p'))$.

Let $\Phi \in \mathbf{PS}$, let $p_1, \ldots, p_n \in \mathcal{P}^{\mathsf{free}}(\Phi)$ and let v be a cell such that $v \notin \mathcal{C}^{\mathsf{term}}(\Phi)$. In the sequel we will use the notion of "adding v on $p_1, \ldots, p_n$ in Φ" which we does not define explicitly. The intuition is that the "add v on $p_1, \ldots, p_n$ in Φ" is the ps Φ' obtained from Φ by adding the cell v in such a way that $p_1, \ldots, p_n$ becomes the "auxiliary ports" of v in Φ' (possibly by adding also the necessary wires if some p_i is the principal port of a terminal cell in Φ).

Definition 105 (1-projection)**.** *Let $R \in \mathbf{PS}_{\mathsf{MELL}}$ and let Φ be the* 2-DiLL_0*-ps of R. The* 1-projection of Φ *is a 3-tuple $(1(\Phi), \pi_{\mathcal{P}(\Phi)}, \pi_{\mathcal{C}(\Phi)})$ defined as follows by induction on $\#(R)$ with the lexicographical order on $\mathbb{N}^3$, where $1(\Phi)$ is a* DiLL_0*-ps such that $\mathsf{a}_{1(\Phi)}(l) = 1$ for every $l \in \mathcal{C}^!(1(\Phi))$, $\pi_{\mathcal{P}(\Phi)} : \mathcal{P}(\Phi) \to \mathcal{P}(1(\Phi))$ and $\pi_{\mathcal{C}(\Phi)} : \mathcal{C}(\Phi) \to \mathcal{C}(1(\Phi))$*

- *If R is the empty ps, then $1(\Phi)$ is the empty ps, $\pi_{\mathcal{P}(\Phi)}$ and $\pi_{\mathcal{C}(\Phi)}$ are the empty functions.*

- *If $\mathcal{A}x^{\mathsf{isol}}(R) \neq \emptyset$, let $\{p, q\} \in \mathcal{A}x^{\mathsf{isol}}(R)$, let R' be the erasure*[5] *of $\{p, q\}$ in R and let Φ' be the erasure of $\{p_0, q_0\}$ in Φ where $\tau_{R,\Phi}(p) = \{p_0\}$ and $\tau_{R,\Phi}(q) = q_0$. It is immediate to check that Φ' is the* 2-DiLL_0*-ps of R'. Since $\#(R') < \#(R)$, $(1(\Phi'), \pi_{\mathcal{P}(\Phi')}, \pi_{\mathcal{C}(\Phi')})$ is defined. We set:*

---

[4] For the definition of $\tau_{R,\Phi}$, see page 61.

[5] See definition 24.

  - $1(\Phi)$ *is the disjoint union of* $1(\Phi')$ *and the ps consisting only of the axiom* $\{p,q\}$;
  - $\pi_{\mathcal{P}(\Phi)} = \pi_{\mathcal{P}(\Phi')} \cup id_{\tau_{R,\Phi}(p) \cup \tau_{R,\Phi}(q)}$;
  - $\pi_{\mathcal{C}(\Phi)} = \pi_{\mathcal{C}(\Phi')}$.

- *If there exists* $l \in \mathcal{C}^{\mathsf{term}}(R)$ *such that either* $l \in \mathcal{C}^{\otimes,\parr}(R) \cup \mathcal{C}^{1,\perp}(R)$, *or* $l \in \mathcal{C}^{?}(R)$ *with* $\mathsf{a}_R(l) \leq 1$ *and* $\mathcal{P}^{\mathsf{aux}}_l(R) \cap \mathcal{A}uxdoors(R) = \emptyset$, *then let* R' *be the erasure of* l *in* R *and let* Φ' *be the erasure*[6] *of* l_0 *in* Φ *where* $\tau_{R,\Phi}(l) = \{l_0\}$. *It is immediate to check that* Φ' *is the* $2\text{-}\mathsf{DiLL}_0$*-ps of* R'. *As* $\#(R') < \#(R)$, $(1(\Phi'), \pi_{\mathcal{P}(\Phi')}, \pi_{\mathcal{C}(\Phi')})$ *is defined. We set:*
  - $1(\Phi)$ *is the "add of* l_0 *on* $\mathcal{P}^{\mathsf{aux}}_{l_0}(\Phi)$ *in* $1(\Phi')$*";*
  - $\pi_{\mathcal{P}(\Phi)} = \pi_{\mathcal{P}(\Phi')} \cup id_{\{\mathsf{P}^{\mathsf{pri}}_{\Phi}(l_0)\}}$;
  - $\pi_{\mathcal{C}(\Phi)} = \pi_{\mathcal{C}(\Phi')} \cup id_{\tau_{R,\Phi}(l)}$.

- *If there exists* $l \in \mathcal{C}^{\mathsf{term}}(R) \cap \mathcal{C}^{?}(R)$ *such that* $\mathsf{a}_R(l) \geq 2$, *let* $p \in \mathcal{P}^{\mathsf{aux}}_l(R)$. *It is immediate to check that* $\mathsf{sep}_\Phi(p_0)$ *is*[7] *the* $2\text{-}\mathsf{DiLL}_0$*-ps of* $\mathsf{sep}_R(p)$, *where* $\tau_{R,\Phi}(p) = \{p_0\}$. *As* $\#(\mathsf{sep}_R(p)) < \#(R)$, $(1(\Phi'), \pi_{\mathcal{P}(\Phi')}, \pi_{\mathcal{C}(\Phi')})$ *is defined. We set:*
  - $1(\Phi) = \mathsf{join}_{1(\Phi')}(l_0, \mathsf{newsep}_\Phi(p_0))$ *where* $\tau_{R,\Phi}(l) = \{l_0\}$;[8]
  - $\pi_{\mathcal{P}(\Phi)} = \pi_{\mathcal{P}(\Phi')} \restriction_{\mathcal{P}(\Phi') \setminus \{\mathcal{P}^{\mathsf{pri}}_{\Phi'}(\mathsf{newsep}_\Phi(p_0))\}}$;
  - $\pi_{\mathcal{C}(\Phi)} = \pi_{\mathcal{C}(\Phi')} \restriction_{\mathcal{C}(\Phi') \setminus \{\mathsf{newsep}_\Phi(p_0)\}}$.

- *Otherwise* R *is a non empty ps, without isolated axioms, such that* $\mathcal{C}^{\mathsf{box}}(R) \cap \mathcal{C}^{\mathsf{term}}(R) \neq \emptyset$ *and every terminal cell* l *of* R *is either a promotion cell, or a* $?$*-cell such that* $\mathsf{a}_R(l) = 1$ *and* $\mathcal{P}^{\mathsf{aux}}_l(R) \cap \mathsf{auxd}_R(v) \neq \emptyset$ *for some* $v \in \mathcal{C}^{\mathsf{box}}(R)$.[9] *Let* $v \in \mathcal{C}^{\mathsf{box}}(R) \cap \mathcal{C}^{\mathsf{term}}(R)$ *and let* p_v *be its unique auxiliary port, let* $\tau_{R,\Phi}(p_v) = \{q, q'\}$, *let* R' *be the erasure of* v *in* R, *let* Φ'' *be the erasure*[10] *of* $\epsilon_\Phi(q)$ *in* Φ *(where* $\epsilon_\Phi(q)$ *is the empire*[11] *of* q *in* Φ*) and let* Φ' *be the erasure of* v *in* Φ''. *It is immediate to check that* Φ' *is the* $2\text{-}\mathsf{DiLL}_0$*-ps of* R'. *As* $\#(R') < \#(R)$, $(1(\Phi'), \pi_{\mathcal{P}(\Phi')}, \pi_{\mathcal{C}(\Phi')})$ *is defined. One has*[12] $\varphi : \mathsf{module}_\Phi(\epsilon_\Phi(q)) \simeq \mathsf{module}_\Phi(\epsilon_\Phi(q'))$. *We set:*
  - $1(\Phi)$ *is the "add of* v_0 *on* q' *in* $1(\Phi')$*", where* $\tau_{R,\Phi}(v) = \{v_0\}$;

---

[6] See definition 21.

[7] See definition 32.

[8] See definition 30.

[9] Roughly speaking, in this case there are only (and at least one) terminal boxes such that all the auxiliary doors are auxiliary ports of unary ?-cells: in particular, all the terminal ?-cells are unary.

[10] See definition 24.

[11] See definition 73.

[12] See definition 26.

- $\pi_{\mathcal{P}(\Phi)}(p) = \begin{cases} \pi_{\mathcal{P}(\Phi')}(p) & \text{if } p \in \mathcal{P}(\Phi') \\ p & \text{if } p = \mathcal{P}^{\mathsf{pri}}_{\Phi}(v_0) \\ \pi_{\mathcal{P}(\Phi')}(\varphi_{\mathcal{P}}(p)) & \text{if } p \in \epsilon_{\Phi}(q) \end{cases}$
- $\pi_{\mathcal{C}(\Phi)}(l) = \begin{cases} \pi_{\mathcal{C}(\Phi')}(l) & \text{if } l \in \mathcal{C}(\Phi') \\ l & \text{if } l = v_0 \\ \pi_{\mathcal{C}(\Phi')}(\varphi_{\mathcal{C}}(l)) & \text{if } l \in \mathcal{C}(\mathsf{module}_{\Phi}(\epsilon_{\Phi}(q))) \end{cases}$

Given a MELL-ps R and a 2-DiLL_0-ps Φ of R, the 1-projection of Φ morally is the DiLL_0-ps obtained from Φ by erasing, for every $l \in \mathcal{C}^{!}(\Phi)$, the empire of one of the two auxiliary ports of l. The function $\pi_{\mathcal{P}(\Phi)}$ (resp. $\pi_{\mathcal{C}(\Phi)}$) associates with every port (resp. cell) l of Φ the corresponding port (resp. cell) in $1(\Phi)$ in such a way that if l is erased in $1(\Phi)$ then

Lemma 106. *Let $R \in \mathbf{PS}_{\mathsf{MELL}}$, let Φ be the* 2-DiLL_0*-ps of R and let $(1(\Phi), \pi_{\mathcal{P}(\Phi)}, \pi_{\mathcal{C}(\Phi)})$ be the* 1*-projection of Φ with $1(\Phi) \in R^*$. For every $R' \in \mathbf{PS}_{\mathsf{MELL}}$, if $\Phi, 1(\Phi) \in R'^*$ then $R \simeq R'$.*

Proof. We prove that $R = R'$ up to isomorphisms. For the sake of simplicity, we ignore in this proof all the problems related to isomorphisms.

By lemma 97 and theorem 100, $\mathsf{lin}(R) = 1(\Phi) = \mathsf{lin}(R')$.

It is enough to show that, given $v \in \mathcal{C}^{\mathsf{box}}(R)$ whose unique auxiliary port is p_v, if $q \in \tau_{R,\Phi}(p_v)$ and $\partial\epsilon_{\Phi}(q) = \{q_1, \ldots, q_n\}$ for some $n \in \mathbb{N}$, then $\mathsf{auxd}_R(v) = \{\pi_{\mathcal{P}(\Phi)}(q_1), \ldots, \pi_{\mathcal{P}(\Phi)}(q_n)\}$. Indeed (since $\mathsf{lin}(R) = \mathsf{lin}(R')$) we can derive from this equality that $\mathsf{auxd}_R(v) = \mathsf{auxd}_{R'}(v)$ for every $v \in \mathcal{C}^{\mathsf{box}}(R) = \mathcal{C}^{\mathsf{box}}(R')$.

The proof is by induction on $\#(R)$ with the lexicographical order on $\mathbb{N}^3$.

The only interesting case is when R is a non empty ps, without isolated axioms, such that $\mathcal{C}^{\mathsf{box}}(R) \cap \mathcal{C}^{\mathsf{term}}(R) \neq \emptyset$, every terminal cell l is either a promotion cell with $\mathsf{auxd}_R(l) \neq \emptyset$, or a ?-cell such that $\mathsf{a}_R(l) = 1$ and there exists $v \in \mathcal{C}^{\mathsf{box}}(R)$ with $\mathcal{P}^{\mathsf{aux}}_l(R) \cap \mathsf{auxd}_R(v) \neq \emptyset$. Let $v \in \mathcal{C}^{\mathsf{box}}(R) \cap \mathcal{C}^{\mathsf{term}}(R)$ and let p_v be its unique auxiliary port, let $\tau_{R,\Phi}(p_v) = \{q, q'\}$ and let $\partial\epsilon_{\Phi}(q) = \{q_1, \ldots, q_n\}$ and $\partial\epsilon_{\Phi}(q') = \{q'_1, \ldots, q'_n\}$ for some $n \in \mathbb{N}$.

Let $p \in \mathsf{auxd}_R(v)$. Then $p \in \mathcal{P}(\mathsf{lin}(R))$ and since $1(\Phi) = \mathsf{lin}(R) \in R^*$ one has $p \in \mathsf{im}(\pi_{\mathcal{P}(\Phi)})$. For such a port $p \in \mathsf{auxd}_R(v)$ one has $\mathsf{a}_{\Phi}(\mathsf{C}_{\Phi}(\mathsf{q}'')) \geq 2$ for every $q'' \in \pi^{-1}_{\mathcal{P}(\Phi)}(p)$ and either $q'' \in \partial\epsilon_{\Phi}(q) \smallsetminus \epsilon_{\Phi}(q')$ or $q'' \in \partial\epsilon_{\Phi}(q') \smallsetminus \epsilon_{\Phi}(q)$, hence $\mathsf{auxd}_R(v) \subseteq \{\pi_{\mathcal{P}(\Phi)}(q_1), \ldots, \pi_{\mathcal{P}(\Phi)}(q_n)\}$.

Let $q'' \in \partial\epsilon_{\Phi}(q)$. Necessarily $\pi_{\mathcal{P}(\Phi)}(q'') \in \mathsf{inbox}_R(v)$. If $\pi_{\mathcal{P}(\Phi)}(q'') \notin \mathsf{auxd}_R(v)$ then there would exist $p \in \mathsf{inbox}_R(v) \cap \mathsf{auxd}_R(v)$ such that $p <_R \pi_{\mathcal{P}(\Phi)}(q'')$, so there would exist $q''' \in \pi^{-1}_{\mathcal{P}(\Phi)}(p) \cap \epsilon_{\Phi}(q)$ such that $q''' <_{\Phi} q''$, that is impossible. Therefore $\pi_{\mathcal{P}(\Phi)}(q'') \in \mathsf{auxd}_R(v)$ and so $\mathsf{auxd}_R(v) \supseteq \{\pi_{\mathcal{P}(\Phi)}(q_1), \ldots, \pi_{\mathcal{P}(\Phi)}(q_n)\}$. ☐

Lemma 107. *Let $R \in \mathbf{PN}_{\mathsf{MELL}}$. If Φ is the* 2-DiLL_0*-ps of R and $(1(\Phi), \pi_{\mathcal{P}(\Phi)}, \pi_{\mathcal{C}(\Phi)})$ is the* 1*-projection of Φ, then $1(\Phi)$ is the* 1-DiLL_0*-ps of R.*

Proof. By induction on $\#(R)$ with the lexicographical order on $\mathbb{N}^3$. The only interesting case is when R is a non empty ps, without isolated axioms, such that $\mathcal{C}^{\mathsf{box}}(R) \cap \mathcal{C}^{\mathsf{term}}(R) \neq \emptyset$, every the terminal cell l is either a promotion cell with $\mathsf{auxd}_R(l) \neq \emptyset$, or a ?-cell such that $\mathsf{a}_R(l) = 1$ and there exists $v \in \mathcal{C}^{\mathsf{box}}(R)$ with $\mathcal{P}_l^{\mathsf{aux}}(R) \cap \mathsf{auxd}_R(v) \neq \emptyset$. Let $v \in \mathcal{C}^{\mathsf{box}}(R) \cap \mathcal{C}^{\mathsf{term}}(R)$ and let p_v be its unique auxiliary port, let $\tau_{R,\Phi}(p_v) = \{q, q'\}$ and let $\partial\epsilon_\Phi(q) = \{q_1, \ldots, q_n\}$ and $\partial\epsilon_\Phi(q') = \{q'_1, \ldots, q'_n\}$ for some $n \in \mathbb{N}$. Let R' be the erasure of v in R, let Φ'' be the erasure of $\epsilon_\Phi(q)$ in Φ and let Φ' be the erasure of u in Φ'' where $\tau_{R,\Phi}(v) = \{u\}$. Since R is ACC, Φ' is the 2-DiLL_0-ps of R'. By induction hypothesis $1(\Phi')$ is the 1-DiLL_0-ps of R'. Hence $1(\Phi)$ (which is the "add of u on $\pi_{\mathcal{P}(\Phi)}(q)$ in $1(\Phi')$") is the 1-DiLL_0-ps of R. □

The following theorem says that a cut-free and deadlock-free MELL-proof-net R (i.e. a MELL-ps satisfying the ACC correctness criterion) is completely characterized by its atomic 2-point in its interpretation in the relational semantics.

Theorem 108. *Let $r, r' \in D^{<\omega}$ be 2-points, let $(R, \mathsf{ind}), (R', \mathsf{ind}') \in \mathbf{PN}^{\mathrm{ind}}$ be cut-free, deadlock-free and such that $r \in [\![(R, \mathsf{ind})]\!]_{\mathrm{inj,at}}$ and $r' \in [\![(R', \mathsf{ind}')]\!]_{\mathrm{inj,at}}$. If $\overline{r} = \overline{r'}$ then $(R, \mathsf{ind}) \simeq (R', \mathsf{ind}')$.*

Proof. By lemmas 106 and 107. □

Corollary 109 (Injectivity)**.** *Let $(R, \mathsf{ind}), (R', \mathsf{ind}') \in \mathbf{PN}^{\mathrm{ind}}$ be cut-free and deadlock-free. If $[\![(R, \mathsf{ind})]\!] = [\![(R', \mathsf{ind}')]\!]$ then $(R, \mathsf{ind}) \simeq (R', \mathsf{ind}')$.*

Part II

Call-by-value lambda calculus

Chapter 3

About a call-by-value λ-calculus

First formulated by Alonzo Church in 1936, λ-calculus is a formal system in mathematical logic and theoretical computer science for expressing computation by way of variable binding and substitution. It found early successes in the area of computability theory, such as a negative answer to Hilbert's Entscheidungsproblem. As pointed out by Peter Landin's 1965 paper [Lan65], sequential procedural programming languages can be understood in terms of the λ-calculus, which provides the basic mechanisms for procedural abstraction and procedure (subprogram) application. The λ-calculus may be seen as the idealized prototype of functional programming languages, like Lisp, Haskell or the various dialects of ML. Under this view, β-reduction (the operation performing substitution of a bound variable for an argument) corresponds to a computational step.

Because of the importance of the notion of variable binding and substitution, there is not just one system of λ-calculus, and in particular there are typed and untyped variants. Historically, the most important system was the untyped λ-calculus, in which function application has no restrictions (so the notion of the domain of a function is not built into the system). In the Church–Turing Thesis, the untyped lambda calculus is claimed to be capable of computing all effectively calculable functions; actually untyped λ-calculus is equivalent to all the models of computation having the highest expressive power nowadays known, like Turing machines and recursive functions. The typed λ-calculus is a variety that restricts function application, so that functions can only be applied if they are capable of accepting the given input's "type" of data.

Another variant of λ-calculus is the "call-by-value" λ-calculus. The most commonly used parameter passing policy for programming languages is call-by-value (CBV). Landin in [Lan65] pioneered a CBV formal evaluation for a lambda-core of ALGOL60 (named ISWIM) via the SECD abstract machine.

Ten years later, Plotkin in [Plo75] introduced the λ_{β_v}-calculus in order to grasp the CBV paradigm in a pure lambda-calculus setting. The λ_{β_v}-calculus narrows the β-reduction rule by allowing the reduction of a redex $(\lambda x\, t)u$, only in case u is a value, i.e. a variable or an abstraction.

3.1 A call-by-value λ-calculus

We will study now Λ_{CBV}, a call-by-value λ-calculus introduced in [Ehr12] by Ehrhard and inspired by his analysis of the relational model for Linear Logic.

3.1.1 The syntax of Λ_{CBV}

Let $\mathcal{V}$ be a countable set whose elements, denoted by $x, y, z, \ldots$, are called *variables*.

Definition 110. *We define the elements of the sets* Λ_{t} *(*terms*),* Λ_{v} *(*values*),* Λ_{CBV} *(*expressions*) by mutual induction as follows:*

$$\begin{array}{llll} \Lambda_{\mathsf{t}} & L, M, N & ::= (M)N \mid (V)^! & \textit{terms} \\ \Lambda_{\mathsf{v}} & U, V, W & ::= x \mid \lambda x M & \textit{values} \\ \Lambda_{\mathsf{CBV}} & D, E, F & ::= M \mid V & \textit{expressions} \end{array}$$

Note that $\Lambda_{\mathsf{CBV}} = \Lambda_{\mathsf{t}} \uplus \Lambda_{\mathsf{v}}$. Terms of the shape $(V)^!$ (resp. $(M)N$) for some value V (resp. terms M and N) are called *promoted values* (resp. *applications*). Terms of the shape $(M)N$ for some terms M and N are called *applications*, M (resp. N) is in *function* (resp. *argument*) position. Values of the shape $\lambda x M$ for some term M are called *abstractions*.

Notation. We follows the Krivine's notation (see [Kri93]) for applications, where the parentheses are on the function. For instance, the term $(M)(N)L$ according to our notation is the term $M(NL)$ according to Barendregt's notation (see [Bar84]).

Let $M, N_1, \ldots, N_n$ be terms, with $n \in \mathbb{N}$: if no ambiguity arise, often we use the notation $(M)N_1 \ldots N_n$ or $MN_1 \ldots N_n$ for $(\ldots((M)N_1)\ldots)N_n$, in particular if $n = 0$ then it stands for M.

If $n = 0$, $(N_1)\ldots(N_n)M$ stands for M.

If V is a value, often we write $V^!$ instead of $(V)^!$.

Definition 111. *With every expression E is associated its* size $\mathsf{size}(E) \in \mathbb{N}^*$, *defined by induction on E as follows:*

- $\mathsf{size}(x) = 1$;
- $\mathsf{size}(V^!) = \mathsf{size}(V) + 1$;
- $\mathsf{size}(\lambda x M) = \mathsf{size}(M) + 1$;
- $\mathsf{size}(MN) = \mathsf{size}(M) + \mathsf{size}(N) + 1$.

For every expression E, $\mathsf{size}(E)$ is the number of rules of definition 110 used to build E, in other words it is the sum of the nodes in the usual tree-like representation of M.

Due to the presence of constructor $(\,)^!$ (which allows to separate terms and values into two distinct sets), the set Λ_{CBV} of expressions does not coincide with the set Λ of ordinary λ-terms. By the way, there is an obvious "forgetful functor" $(\,)^{\mathrm{F}}$ from Λ_{CBV} to Λ, defined as follows (by induction on the expression in Λ_{CBV}):

$$\begin{array}{rclcrcl} (x)^{\mathrm{F}} & = & x & \qquad & (\lambda x M)^{\mathrm{F}} & = & \lambda x\,(M)^{\mathrm{F}} \\ (V^!)^{\mathrm{F}} & = & V^{\mathrm{F}} & \qquad & (MN)^{\mathrm{F}} & = & (M)^{\mathrm{F}}(N)^{\mathrm{F}} \end{array}$$

Definitions of free variables, α-equivalence and substitution (avoiding variable capture) are extended to expressions as expected. For instance, the *free occurrences* of a variable x in an expression E are defined, by induction on E, as follows :

- if E is the variable x, then the occurrence of x in E is free;
- if $E = (M)N$ for some terms M and N, then the free occurrences of x in E are those of x in M and N;
- if $E = \lambda y M$ for some term M, the free occurrences of x in E are those of x in M, except if $x = y$; in that case, no occurrence of x in E is free.
- if $E = (V)^!$ for some value V, the free occurrences of x in E are those of x in V.

A *free variable* in an expression E is a variable which has at least one free occurrence in E; the set of free variables in E is denoted by $\mathsf{fv}(E)$. An expression which has no free variable is said *closed.* A *bound variable* in an expression E is a variable which occurs in E just after the symbol λ. In an expression $\lambda x M$ the λx before M *binds* the free occurrences of x in M.

We work up to α-equivalence.

As another example of notion coming from ordinary λ-calculus trivially adapted to Λ_{CBV}, the operation of substitution avoiding variable capture is extended by setting

$$V^![W_1/x_1, \ldots, W_n/x_n] = (V[W_1/x_1, \ldots, W_n/x_n])^!$$

for any values $V, W_1, \ldots, W_n$, variables $x_1, \ldots, x_n$ and $n \in \mathbb{N}$. Notice that the substitution is defined only for values replacing variables. The following lemma extends at Λ_{CBV} a substitution lemma of ordinary λ-calculus.

Lemma 112.

1. *Let E be an expression, let V, W be values and let x, y be variables. If $x \notin \mathsf{fv}(W) \cup \{y\}$ then $E[V/x][W/y] = E[W/y][V[W/y]/x]$.*

2. *If E is a vale (resp. a term), V is a value and x is a variable, then $E[V/x]$ is a value (resp. a term).*

PROOF.

1. By induction on the expression E. The only novelty is the case where $E = U^!$ for some value U: by applying induction hypothesis the identity holds.

2. By a straightforward induction on the expression E.

$\square$

Remark 113. It follows immediately from definition that:

1. Every term is in the form

$$(V^!)M_1 \dots M_m \qquad \text{and} \qquad (N_1) \cdots (N_n)W^!$$

where V and W are values, $m, n \in \mathbb{N}$, M_i and N_j are terms for every $1 \leq i \leq m$ and $1 \leq j \leq n$; both of these forms are unique, and $m = 0$ iff $n = 0$; moreover if $m = 0$ (i.e. $n = 0$) then $V = W$, otherwise $M_m = (N_2) \cdots (N_n)W^!$ and $N_1 = (V^!)M_1 \dots M_{m-1}$.

2. As any term has always a finite length, applying recursively the left (resp. right) decomposition in remark 113.1 yields that every term M either is such that $M = V^!$ for some value V or there exist $\ell \in \mathbb{N}$, values $V, V_0, \dots, V_\ell$ and terms $L_0, \dots, L_\ell$ such that:

   - $L_0 = (V^!)V_0^! L_{01} \cdots L_{0k_0}$ (resp. $L_0 = (L_{01}) \cdots (L_{0k_0})(V^!)V_0^!$) for some $k_0 \in \mathbb{N}$ and terms $L_{01}, \dots, L_{0k_0}$;
   - for every $1 \leq i \leq \ell$, we have $L_i = (V_i^!)L_{i-1}L_{i1} \cdots L_{ik_i}$ (resp. $L_i = (L_{i1}) \cdots (L_{ik_i})(L_{i-1})V_i^!$) for some $k_i \in \mathbb{N}$ and terms $L_{i1}, \dots, L_{ik_i}$;
   - $M = L_\ell$.

   Both of this decompositions are unique. It is more natural to consider these decompositions as a binary tree, see §3.4.

3. Every closed value is in the form $\lambda x N$, where N is a term such that $\mathsf{fv}(N) \subseteq \{x\}$.

Definition 114. *For every expression E, we define by induction on E the set $\mathsf{sub}(E)$ of the* subexpressions *of E as follows:*

$$\begin{array}{rclcrcl} \mathsf{sub}(x) & = & \{x\} & \qquad & \mathsf{sub}(\lambda x M) & = & \mathsf{sub}(M) \cup \{\lambda x M\} \\ \mathsf{sub}(V^!) & = & \mathsf{sub}(V) \cup \{V^!\} & & \mathsf{sub}(MN) & = & \mathsf{sub}(M) \cup \mathsf{sub}(N) \cup \{(M)N\} \end{array}$$

A subterm *(resp.* subvalue*) of an expression E is a term (resp. value) which is a subexpression of E.*

Definition 115 (β_{v}- and $\hat{\beta}_{\mathsf{v}}$-redex)**.** *A* β-redex *is a term of the shape* $(\lambda xM)^{!}N$ *for some terms* M *and* N*. A* β_{v}-redex *is a term of the shape* $(\lambda xM)^{!}V^{!}$ *for some value* V *and term* M*, its* contractum *is the term* $M[V/x]$*.*

A σ_1 *(resp.* σ'_3*)*-redex *is a term of the shape* $(\lambda xM)^{!}NL$ *(resp.* $(M)(\lambda xL)^{!}N$*) for some terms* M*,* N *and* L *with* $x \notin \mathsf{fv}(L)$ *(resp.* $x \notin \mathsf{fv}(M)$*), its* contractum *is the term* $(\lambda xML)^{!}N$*. A* σ_3-redex *is a* σ'_3*-redex* $(M)(\lambda xL)^{!}N$ *such that* $M = V^{!}$ *for some value* V*.*

A σ_{v} *(resp.* σ'_{v}*)*-redex *is either a* σ_1*-redex or a* σ_3 *(resp.* σ'_3*)-redex. A* $\beta_{\mathsf{v}\sigma}$ *(resp.* $\beta_{\mathsf{v}\sigma'}$*)*-redex *is either a* β_{v}*-redex or a* σ_{v} *(resp.* σ'_{v}*)-redex.*

Let E *be an expression and let* $R \in \{\beta, \beta_{\mathsf{v}}, \sigma_1, \sigma_3, \sigma'_3, \sigma_{\mathsf{v}}, \sigma'_{\mathsf{v}}, \beta_{\mathsf{v}\sigma}, \beta_{\mathsf{v}\sigma'}\}$*. A* R-redex in E *is an occurrence in* E *of a subterm of* E *which is a* R*-redex. A* $\hat{R}$-redex in E *is a* R*-redex in* E *which is not in any subvalue of* E*. We say that* E contains a R*(resp.* $\hat{R}$*)*-redex *if there is a* R*(resp.* $\hat{R}$*)-redex in* E*.*

In order to compare β_{v}-redexes in Λ_{CBV} with β-redexes in ordinary (call-by-name) λ-calculus by means of the "forgetful functor" $(\,)^{\mathrm{F}}$, we observe that if a term $M \in \Lambda_{\mathsf{CBV}}$ is a β_{v}-redex then M^{F} is a β-redex (in Λ), but the converse does not hold: for instance, $M = (\lambda x\, x^{!})^{!}(y^{!})z^{!}$ is not a β_{v}-redex, differently from $M^{\mathrm{F}} = (\lambda x\, x)(y)z$ which is a β-redex. Indeed for every term $t \in \Lambda$ there exists a β_{v}-redex $M \in \Lambda_{\mathsf{CBV}}$ such that $M^{\mathrm{F}} = t$ iff $t = (\lambda x\, u)v$ for some $u, v \in \Lambda$ such that v is a variable or an abstraction. Essentially, modulo the "forgetful functor" $(\,)^{\mathrm{F}}$, a β_{v}-redex is a β-redex such that its argument is a value.

A $\hat{\beta}_{\mathsf{v}}$-redex can be seen as a "outermost" β_{v}-redex, that is a β_{v}-redex not contained in any other β_{v}-redex.

Remark 116.

1. Every term of the shape $V^{!}$ for some value V contains no $\hat{\beta}_{\mathsf{v}}$-redexes, indeed a variable is not a $\hat{\beta}_{\mathsf{v}}$-redex and, for every term N, any possible $\hat{\beta}_{\mathsf{v}}$-redex in N is "invisible" in $(\lambda xN)^{!}$ since λxN is a subvalue of $(\lambda xN)^{!}$.

   On the contrary, a value V (and so a term $V^{!}$) might contains a β_{v}-redex. For instance $(\lambda y\, x^{!})^{!}x'^{!}$ is a β_{v}-redex in the value $\lambda d(\lambda y\, x^{!})^{!}z^{!}$ (and in the term $(\lambda d(\lambda y\, x^{!})^{!}z^{!})^{!}$).

2. If a term contains several $\hat{\beta}_{\mathsf{v}}$-redexes then they are non-overlapping, i.e. they have no common occurrences of subexpressions. Indeed, a $\hat{\beta}_{\mathsf{v}}$-redex is of the shape $(\lambda xN)^{!}V^{!}$ for some term N and value V, where $(\lambda xN)^{!}$ and $V^{!}$ contain no $\hat{\beta}_{\mathsf{v}}$-redexes (see remark 116.1).

The following definitions will be used to define and characterize binary relations on the set Λ_{CBV} of expressions.

Definition 117 (Contextual and applicative closure)**.** *Let* R *be a binary relation on* Λ_{CBV}*.*

We say that R passes to context *(resp.* R passes to applicative contexts*) if* R *is such that the following conditions 1, 2, 3 and 4 (resp. 1 and 2) hold, for any terms* M, M', N *and values* V, V'*:*

1. *if* $M\ R\ M'$ *then* $MN\ R\ M'N$*;*
2. *if* $M\ R\ M'$ *then* $NM\ R\ NM'$*;*
3. *if* $V\ R\ V'$ *then* $V^!\ R\ V'^!$*;*
4. *if* $M\ R\ M'$ *then* $\lambda xM\ R\ \lambda xM'$*.*

The contextual closure *of* R *is the binary relation* R' *on* Λ_{CBV} *defined by applying, a finite number of times, the following rules:*

$$\frac{M\ R\ N}{M\ R'\ N}\,R \qquad \frac{M\ R'\ M'}{MN\ R'\ M'N}\,@_{\mathsf{l}} \qquad \frac{N\ R'\ N'}{MN\ R'\ MN'}\,@_{\mathsf{r}}$$

$$\frac{M\ R'\ M'}{\lambda xM\ R'\ \lambda xM'}\,\lambda \qquad \frac{V\ R'\ W}{V^!\ R'\ W^!}\,!$$

The applicative closure *of* R *is the binary relation* R' *on* Λ_{CBV} *defined by applying, a finite number of times, the following rules:*

$$\frac{M\ R\ N}{M\ R'\ N}\,R \qquad \frac{M\ R'\ M'}{MN\ R'\ M'N}\,@_{\mathsf{l}} \qquad \frac{N\ R'\ N'}{MN\ R'\ MN'}\,@_{\mathsf{r}}$$

In the sequel we will consider contextual or applicative closures R' of relations R defined only by axiom rules. Therefore, thanks to the R-rule, we are entitled to talk about the axiom rules of R as derivation rules of the relation R'.

Notation. Let R be a binary relation on a set X.

We denote by $R^=$ (resp. R^+; R^*; R^T) the reflexive (resp. transitive; reflexive-transitive; symmetric) closure of R. We denote by $\simeq_R$ the symmetric and reflexive-transitive closure of R, i.e. $\simeq_R = (R^T)^*$.

Let $E, F \in X$ and $n \in \mathbb{N}$: we say that E *R-reduces to F in n steps* (and we write $E\ R^n\ F$) if there exists a finite sequence $(E_i)_{0\leq i\leq n}$ of elements of X such that $E = E_0$, $F = E_n$ and $E_i\ R\ E_{i+1}$ for every $0 \leq i < n$.

Remark 118.

1. If $R \subseteq \Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}$ and R' is the contextual (resp. applicative) closure of R, then $R' \subseteq (\Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}) \cup (\Lambda_{\mathsf{v}} \times \Lambda_{\mathsf{v}})$ (resp. $R' \subseteq \Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}$). The proof is by a straightforward induction on the derivation of $E\ R'\ F$, where E and F are expressions.
2. If R is a binary relation on Λ_{CBV} passing to context (resp. passing to applicative context) then $R^=$, R^+, R^*, R^T and $\simeq_R$ pass to context (resp. pass to applicative context).

We recall some standard definitions in term rewriting systems.

Definition 119. *Let R be a binary relation on a set X and let $E \in X$.*

E *is a* R-normal form *or is* R-normal *if there is no expression E' such that $E\ R\ E'$.*

A R-normal form of E *is a R-normal form E' such that $E\ \beta_{\mathsf{v}}^*\ E'$.*

E *is* R-normalizable *if there exists a R-normal form of E.*

E *is* R-strongly normalizable *if there is no infinite sequence $(E_i)_{i\in\mathbb{N}}$ of elements of X such that $E_0 = E$ and $E_i\ R\ E_{i+1}$ for every $i \in \mathbb{N}$.*

Definition 120. *Let R be a binary relation on a set X.*

R *is* strongly *(resp.* locally*)* confluent *if for every $E, E_1, E_2 \in X$ such that $E\ R\ E_i$ for $i \in \{1, 2\}$ there exists E' such that $E_i\ R\ E'$ (resp. $E_i\ R^*\ E'$) for $i \in \{1, 2\}$.*

R *is* confluent *if R^* is strongly confluent.*

We recall a well-known result of term rewriting system.

Theorem 121. *Let R be a confluent binary relation on a set X and let $E_1, E_2 \in X$. If $E_1 \simeq_R E_2$ then there exists $E \in X$ such that $E_1\ R^*\ E$ and $E_2\ R^*\ E$.*

3.1.2 Some call-by-value β-reductions

The following notions of β_{v}- and $\hat{\beta}_{\mathsf{v}}$-reduction are introduced by Ehrhard in [Ehr12]. They formulate respectively the the well-known ([Plo75]) call-by-value and lazy (or weak) call-by-value β-reduction for the syntax presented in §3.1.1. Some results of this section are nothing but a reformulation in Λ_{CBV} of well-known results for call-by-value λ-calculus, the novelty is in pointing out the deep symmetries in β_{v}- and especially $\hat{\beta}_{\mathsf{v}}$-reduction.

Definition 122 (β_{v}- and $\hat{\beta}_{\mathsf{v}}$-reduction)**.** *The* β_{v}-reduction *(resp.* weak β_{v}-reduction *or* $\hat{\beta}_{\mathsf{v}}$-reduction*), denoted by β_{v} (resp. $\hat{\beta}_{\mathsf{v}}$), is the contextual (resp. applicative) closure of the binary relation $\to_{\beta_{\mathsf{v}}}$ on Λ_{t} defined by the following rule:*

$$\frac{}{(\lambda x\, M)^! V^! \to_{\beta_{\mathsf{v}}} M[V/x]}\ \beta$$

where M is a term and V is a value.

Remark 123. By remark 118, $\beta_{\mathsf{v}} \subseteq (\Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}) \cup (\Lambda_{\mathsf{v}} \times \Lambda_{\mathsf{v}})$ and $\hat{\beta}_{\mathsf{v}} \subseteq \Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}$.

In order to compare our call-by-value λ-calculus with ordinary (call-by-name) λ-calculus with respect to reductions by means of the "forgetful functor" $(\,)^{\mathrm{F}}$ (see p. 73), we can prove, by straightforward induction on $E \in \Lambda_{\mathsf{CBV}}$, that if E and E' are expressions such that $E\ \beta_{\mathsf{v}}\ E'$ then $E^{\mathrm{F}}\ \beta\ E'^{\mathrm{F}}$, but the converse does not hold: for instance $M = (\lambda x\, x^!)(y^!)z^!$ is β_{v}-normal (in Λ_{CBV}), on the contrary $M^{\mathrm{F}} = (\lambda x\, x)(y)z \;\beta\; (y)z$ (in Λ). In other words modulo the "forgetful functor" $(\,)^{\mathrm{F}}$, the call-by-value λ-calculus allows to reduce a β-redex only if its argument is a value, i.e. a variable or an abstraction (see the β-rule for β_{v}- and $\hat{\beta}_{\mathsf{v}}$-reductions), whilst there is no such a restriction in ordinary λ-calculus.

Remark 124.

1. It is immediate to check that for every expression E (resp. term M), there exists an expression E' (resp. a term M') such that $E \;\beta_{\mathsf{v}}\; E'$ (resp. $M \;\hat{\beta}_{\mathsf{v}}\; M'$) iff E (resp. M) contains a β_{v}(resp. $\hat{\beta}_{\mathsf{v}}$)-redex. Therefore, an expression (resp. a term) is β_{v}(resp. $\hat{\beta}_{\mathsf{v}}$)-normal iff it contains no β_{v}(resp. $\hat{\beta}_{\mathsf{v}}$)-redex.

2. It is easy to verify that for all expressions E, E' (resp. terms M, M'), $E \;\beta_{\mathsf{v}}\; E'$ (resp. $M \;\hat{\beta}_{\mathsf{v}}\; M'$) iff E' (resp. M') is obtained from E (resp. M) by replacing exactly one β_{v}-(resp. $\hat{\beta}_{\mathsf{v}}$-)redex in E (resp. M) with its contractum.

3. Clearly, $\hat{\beta}_{\mathsf{v}} \subseteq \beta_{\mathsf{v}}$ (the proof is by induction on the length of the derivation of $M \;\hat{\beta}_{\mathsf{v}}\; M'$). More precisely, weak β_{v}-reduction is the β_{v}-reduction with the restriction that it does not reduce under the λ's (whence the word "weak"): $\hat{\beta}_{\mathsf{v}}$-reduction reduces a β_{v}-redex only if there is no λ in front of it. In particular, every β_{v}-normal form is $\hat{\beta}_{\mathsf{v}}$-normal; the converse fails to hold: for instance, $(\lambda d(\lambda y\, x^!)^! z^!)^!$ is $\hat{\beta}_{\mathsf{v}}$-normal but not β_{v}-normal since $(\lambda d(\lambda y\, x^!)^! z^!)^! \;\beta_{\mathsf{v}}\; (\lambda x z^!)^!$.

4. Terms of the shape $V^!$ where V is a value are $\hat{\beta}_{\mathsf{v}}$-normal forms; on the contrary, a value V and so a term $V^!$ are not necessarily β_{v}-normal (see remarks 116.1 and 124.1)

5. All the critical pairs for $\hat{\beta}_{\mathsf{v}}$-reduction (i.e. terms M, M_1, M_2 such that $M \;\hat{\beta}_{\mathsf{v}}\; M_1$ and $M \;\hat{\beta}_{\mathsf{v}}\; M_2$ with $M_1 \neq M_2$) arise from non-overlapping $\hat{\beta}_{\mathsf{v}}$-redexes in the same term (see remarks 116.2 and 124.2)

6. For every expressions E and E', if $E \;\beta_{\mathsf{v}}\; E'$ then $\mathsf{fv}(E') \subseteq \mathsf{fv}(E)$ (the proof is by a straightforward induction on E). In particular, for every closed expression E, if $E \;\beta_{\mathsf{v}}\; E'$ then E' is closed.

In [Ehr12] Ehrhard showed that β_{v}-reduction is confluent and that $\hat{\beta}_{\mathsf{v}}$-reduction enjoys the following propriety: a term is $\hat{\beta}_{\mathsf{v}}$-normalizable iff its interpretation in the relational model for Λ_{CBV} defined in [Ehr12] is not empty. The latter result is analogous to that one in ordinary λ-calculus stating that a (ordinary) term is head-normalizable iff its interpretation in the Engler model is empty. This allows to draw a parallel between $\hat{\beta}_{\mathsf{v}}$-reduction and heand reduction in ordinary λ-calculus. The most apparent difference between these two things is that $\hat{\beta}_{\mathsf{v}}$-reduction is not a reduction strategy, that is a term in Λ_{CBV} might contains several $\hat{\beta}_{\mathsf{v}}$-redexes, whilst every term in ordinary λ-calculus can have at most one head redex. We will show that this is only a seeming difference.

The following notions of size will be used several times, they are well-defined for all terms by remark 113.1.

Definition 125. *For every term M, their sizes $\#_{\mathsf{l}}M \in \mathbb{N}$ and $\#_{\mathsf{r}}M \in \mathbb{N}$ are defined by induction on M as follows:*

$$\#_{\mathsf{l}}M = \begin{cases} 0 & \textit{if } M = V^{!} \textit{ for some value } V; \\ 0 & \textit{if } M = (V^{!})W^{!}N_1 \dots N_n \textit{ for some } n \in \mathbb{N}, \textit{ terms } N_1, \dots, N_n \textit{ and} \\ & \textit{values } V, W; \\ \#_{\mathsf{l}}L_1 + \#_{\mathsf{l}}L_2 + 1 & \textit{if } M = ((V^{!})(L_1)L_2)N_1 \dots N_n \textit{ for some } n \in \mathbb{N}, \textit{ terms } N_1, \dots, N_n, \\ & L_1, L_2 \textit{ and value } V. \end{cases}$$

$$\#_{\mathsf{r}}M = \begin{cases} 0 & \textit{if } M = V^{!} \textit{ for some value } V; \\ 0 & \textit{if } M = (N_1) \cdots (N_n)(W^{!})V^{!} \textit{ for some } n \in \mathbb{N}, \textit{ terms } N_1, \dots, N_n \textit{ and} \\ & \textit{values } V, W; \\ \#_{\mathsf{r}}L_1 + \#_{\mathsf{r}}L_2 + 1 & \textit{if } M = (N_1) \cdots (N_n)((L_1)L_2)V^{!} \textit{ for some } n \in \mathbb{N}, \textit{ terms } N_1, \dots, N_n, \\ & L_1, L_2 \textit{ and value } V. \end{cases}$$

The closed $\hat{\beta}_{\mathsf{v}}$-normal forms are promoted values easily characterizable.

Proposition 126. *Let M be a closed term: M is a $\hat{\beta}_{\mathsf{v}}$-normal form iff $M = (\lambda x N)^{!}$ for some term N with $\mathsf{fv}(N) \subseteq \{x\}$.*

PROOF.

- $\Leftarrow$: Trivial (it is not necessary to suppose M is closed, see also remark 124.4).
- $\Rightarrow$: The proof is by induction on the size $\#_{\mathsf{l}}M \in \mathbb{N}$. By remark 113.1, $M = (V^{!})M_1 \dots M_m$ for some $m \in \mathbb{N}$, terms $M_1, \dots, M_m$ and value V. As M is closed, V is a closed value, thus $V = \lambda x N$ for some term N with $\mathsf{fv}(N) \subseteq \{x\}$.

  If $\#_{\mathsf{l}}M = 0$ then $m = 0$ and so $M = (\lambda x N)^{!}$, otherwise it should be $m > 0$ and $M_1 = W^{!}$ for some value W and so $M = (\lambda x N)^{!}W^{!}M_2 \dots M_m$, that is impossible because M is $\hat{\beta}_{\mathsf{v}}$-normal.

  If $\#_{\mathsf{l}}M > 0$ then it should be $m > 0$ and $M_1 = (L_1)L_2$ for some closed terms L_1, L_2 which are $\hat{\beta}_{\mathsf{v}}$-normal forms (since M is a $\hat{\beta}_{\mathsf{v}}$-normal form), hence $L_1 = (\lambda x_1 N_1)^{!}$ and $L_2 = (\lambda x_2 N_2)^{!}$ for some terms N_1, N_2 by induction hypothesis, thus $M = ((V^{!})((\lambda x_1 N_1)^{!})(\lambda x_2 N_2)^{!})M_2 \dots M_m$, that is impossible because M is $\hat{\beta}_{\mathsf{v}}$-normal.

  Therefore the only possibility is that $M = (\lambda x N)^{!}$ for some term N with $\mathsf{fv}(N) \subseteq \{x\}$.

□

Theorem 127 (Strong confluence for $\hat{\beta}_{\mathsf{v}}$)**.** *Let* M, M_1, M_2 *be terms: if* $M\ \hat{\beta}_{\mathsf{v}}\ M_1$ *and* $M\ \hat{\beta}_{\mathsf{v}}\ M_2$ *with* $M_1 \neq M_2$*, then there exists a term* N *such that* $M_1\ \hat{\beta}_{\mathsf{v}}\ N$ *and* $M_2\ \hat{\beta}_{\mathsf{v}}\ N$*.*

Proof. By induction on the term M. Let us consider the last rule of the derivation of $M\ \hat{\beta}_{\mathsf{v}}\ M_1$.

If it is the β-rule, then $M = (\lambda x N_1)^! V^!$ and $M_1 = N[V/x]$, so there is no $M_2 \neq M_1$ such that $M\ \hat{\beta}_{\mathsf{v}}\ M_2$, since $(\lambda x N_1)^!$ and $V^!$ are $\hat{\beta}_{\mathsf{v}}$-normal forms (see remark 124.4).

If it is the $@_{\mathsf{l}}$-rule, then $M = N_1 N_2$ and $M_1 = N_1' N_2$ with $N_1\ \hat{\beta}_{\mathsf{v}}\ N_1'$, hence $N_1 \neq (\lambda x M')^!$ for any term M' (see remark 124.4). Thus there are only two cases : either $M_2 = N_1 N_2'$ with $N_2\ \hat{\beta}_{\mathsf{v}}\ N_2'$ and then $M_1\ \hat{\beta}_{\mathsf{v}}\ N$ and $M_2\ \hat{\beta}_{\mathsf{v}}\ N$ where $N = N_1' N_2'$; or $M_2 = N_1'' N_2$ with $N_1\ \hat{\beta}_{\mathsf{v}}\ N_1'' \neq N_1'$ by hypothesis, and then there exists a term L such that $N_1'\ \hat{\beta}_{\mathsf{v}}\ L$ and $N_1''\ \hat{\beta}_{\mathsf{v}}\ L$ by induction hypothesis, so $M_1 = N_1' N_2\ \hat{\beta}_{\mathsf{v}}\ N$ and $M_2 = N_1'' N_2\ \hat{\beta}_{\mathsf{v}}\ N$ where $N = L N_2$.

If it is the $@_{\mathsf{r}}$-rule, then $M = N_1 N_2$ and $M_1 = N_1 N_2'$ with $N_2\ \hat{\beta}_{\mathsf{v}}\ N_2'$, hence $N_2 \neq V^!$ for any value V (see remark 124.4). Thus there are only two cases: either $M_2 = N_1' N_2$ with $N_1\ \hat{\beta}_{\mathsf{v}}\ N_1'$ and then $M_1\ \hat{\beta}_{\mathsf{v}}\ N$ and $M_2\ \hat{\beta}_{\mathsf{v}}\ N$ where $N = N_1' N_2'$; or $M_2 = N_1 N_2''$ with $N_2\ \hat{\beta}_{\mathsf{v}}\ N_2'' \neq N_2'$ by hypothesis, and then there exists a term L such that $N_2'\ \hat{\beta}_{\mathsf{v}}\ L$ and $N_2''\ \hat{\beta}_{\mathsf{v}}\ L$ by induction hypothesis, so $M_1 = N_1 N_2'\ \hat{\beta}_{\mathsf{v}}\ N$ and $M_2 = N_1 N_2''\ \hat{\beta}_{\mathsf{v}}\ N$ where $N = N_1 L$.
□

The following corollary of theorem 127 is a well known result which holds for every strongly confluent term rewriting system.

Corollary 128 (Confluence, uniqueness of normal form, number of steps)**.**

1. $\hat{\beta}_{\mathsf{v}}$ *is confluent. More precisely, let* M, M_1, M_2 *be terms: if* $M\ \hat{\beta}_{\mathsf{v}}^*\ M_1$ *in* $m_1 \in \mathbb{N}$ *steps and* $M\ \hat{\beta}_{\mathsf{v}}^*\ M_2$ *in* $m_2 \in \mathbb{N}$ *steps, then there exists a term* N *such that* $M_1\ \hat{\beta}_{\mathsf{v}}^*\ N$ *in* $n_1 \leq m_2$ *steps and* $M_2\ \hat{\beta}_{\mathsf{v}}^*\ N$ *in* $n_2 \leq m_1$ *steps.*

2. *Every term* M *has at most a* $\hat{\beta}_{\mathsf{v}}$*-normal form, and if that exists then all the* $\hat{\beta}_{\mathsf{v}}$*-reductions from* M *to its* $\hat{\beta}_{\mathsf{v}}$*-normal form have the same number of steps.*

3. *Every term* M *is* $\hat{\beta}_{\mathsf{v}}$*-strongly normalizable iff it is* $\hat{\beta}_{\mathsf{v}}$*-normalizable.*

Proof.

1. By induction on $m_1 + m_2 \in \mathbb{N}$.

   If $m_2 = 0$ then $M = M_2$, hence $M_2\ \hat{\beta}_{\mathsf{v}}^*\ M_1$ in m_1 steps (and $M_1\ \hat{\beta}_{\mathsf{v}}^*\ M_1$ in 0 steps).

If $m_1 = 0$ then $M = M_1$, hence $M_1 \ \hat{\beta}_{\mathsf{v}}^* \ M_2$ in m_2 steps (and $M_2 \ \hat{\beta}_{\mathsf{v}}^* \ M_2$ in 0 steps).

If $m_1, m_2 > 0$ then there exist terms L_1, L_2 such that $M \ \hat{\beta}_{\mathsf{v}} \ L_1$ and $M \ \hat{\beta}_{\mathsf{v}} \ L_2$: by theorem 127, there exist a term L such that $L_1 \ \hat{\beta}_{\mathsf{v}}^* \ L$ and $L_2 \ \hat{\beta}_{\mathsf{v}}^* \ L$ in at most one step. By induction hypothesis (as $L_2 \ \hat{\beta}_{\mathsf{v}}^* \ M_2$ in $m_2 - 1$ steps), there exists a term N' such that $L \ \hat{\beta}_{\mathsf{v}}^* \ N'$ in $\ell \leq m_2 - 1$ steps and $M_2 \ \hat{\beta}_{\mathsf{v}}^* \ N'$ in at most one step. Therefore $L_1 \ \hat{\beta}_{\mathsf{v}}^* \ M_1$ in $m_1 - 1$ steps and $L_1 \ \hat{\beta}_{\mathsf{v}}^* \ N'$ in $\ell' \leq \ell + 1 \leq m_2$ steps, so there exists a term N such that $M_1 \ \hat{\beta}_{\mathsf{v}}^* \ N$ in $n_1 \leq \ell' \leq m_2$ steps and $N' \ \hat{\beta}_{\mathsf{v}}^* \ N$ in $n \leq m_1 - 1$ steps by induction hypothesis, thus $M_2 \ \hat{\beta}_{\mathsf{v}}^* \ N$ in $n_2 \leq n + 1 \leq m_1$ steps.

2. If $M \ \hat{\beta}_{\mathsf{v}}^* \ M_1$ and $M \ \hat{\beta}_{\mathsf{v}}^* \ M_2$ where M_1 and M_2 are $\hat{\beta}_{\mathsf{v}}$-normal forms, then there exists a term N such that $M_1 \ \hat{\beta}_{\mathsf{v}}^* \ N$ and $M_2 \ \hat{\beta}_{\mathsf{v}}^* \ N$ by corollary 128.1, so $M_1 = N = M_2$ since M_1 and M_2 are $\hat{\beta}_{\mathsf{v}}$-normal forms.

   Let M' be the $\hat{\beta}_{\mathsf{v}}$-normal form of M. We prove by induction on $m \in \mathbb{N}$ that if $M \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in m steps, then every $\hat{\beta}_{\mathsf{v}}$-reduction from M to M' has length m.

   - If $m = 0$ then $M = M'$ and so M is a $\hat{\beta}_{\mathsf{v}}$-normal form, hence the $\hat{\beta}_{\mathsf{v}}$-reduction of 0 steps is the only $\hat{\beta}_{\mathsf{v}}$-reduction from M to M'.
   - If $m > 0$ then there exists a term M_1 such that $M \ \hat{\beta}_{\mathsf{v}} \ M_1$ and $M_1 \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in $m - 1$ steps. We show that for every term M_2 and $m_2 \in \mathbb{N}$, if $M \ \hat{\beta}_{\mathsf{v}}^* \ M_2$ in m_2 steps and $M_2 \ \hat{\beta}_{\mathsf{v}}^* \ M'$ then $M_2 \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in m' steps with $m = m_2 + m'$.

     If $m_2 = 0$ then $M_2 = M \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in m steps by hypothesis, so we conclude by taking $m' = m$.

     If $m_2 > 0$ then there exists a term N such that $M \ \hat{\beta}_{\mathsf{v}} \ N$ and $N \ \hat{\beta}_{\mathsf{v}}^* \ M_2$ in $m_2 - 1$ steps. If $N = M_1$ then $N \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in $m - 1$ steps by hypothesis, so $M_2 \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in m' steps with $m - 1 = m_2 - 1 + m'$ by induction hypothesis applied to M_1, thus $m = m_2 + m'$. If $N \neq M_1$ then there exists a term N' such that $M_1 \ \hat{\beta}_{\mathsf{v}} \ N'$ and $N \ \hat{\beta}_{\mathsf{v}} \ N'$ by theorem 127, hence $N' \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in n steps with $n + 1 = m - 1$ by induction hypothesis applied to M_1, so $N \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in $n + 1 = m - 1$ steps, therefore by applying the induction hypothesis to N we conclude that $M_2 \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in m' steps with $m - 1 = m_2 - 1 + m'$, hence $m = m_2 + m'$.

3. The left-to-right direction is obvious. For the right-to-left direction, let us suppose by absurd that there exists a $\hat{\beta}_{\mathsf{v}}$-normalizable term M which is not $\hat{\beta}_{\mathsf{v}}$-strongly normalizable: then there should exist the $\hat{\beta}_{\mathsf{v}}$-normal form M' of M and an infinite sequence of terms $(M_i)_{i \in \mathbb{N}}$ such that $M = M_0$ and $M_i \ \hat{\beta}_{\mathsf{v}} \ M_{i+1}$; if $M \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in m steps, then $M_m = M'$ by corollary 128.2, that is impossible because $M_m \ \hat{\beta}_{\mathsf{v}} \ M_{m+1}$ and so M_m is not $\hat{\beta}_{\mathsf{v}}$-normal.

$\square$

The $\hat{\beta}_{\mathsf{v}}$-reduction is not necessarily normalizing: if $M = (\lambda x(x^!)x^!)^!(\lambda x(x^!)x^!)^!$ then $M \;\hat{\beta}_{\mathsf{v}}\; M$ and there is only one $\hat{\beta}_{\mathsf{v}}$-redex in M, so M is a not $\hat{\beta}_{\mathsf{v}}$-normalizable (closed) term. Moreover, the fact that a term is strongly $\hat{\beta}_{\mathsf{v}}$-normalizable does not imply that it is β_{v}-normalizable: for instance, if M is as above then $(\lambda zM)^!$ is a $\hat{\beta}_{\mathsf{v}}$-normal form but $(\lambda zM)^! \;\beta_{\mathsf{v}}\; (\lambda zM)^!$.

Definition 129 (Leftmost and rightmost(-outermost) reduction)**.** *We define two binary relations on* Λ_{t}*:*

- *the* weak leftmost(-outermost) β_{v}-reduction*, denoted by* $\hat{\beta}_{\mathsf{vl}}$*, whose rules are:*

$$\frac{}{(\lambda x\, M)^! V^! \;\hat{\beta}_{\mathsf{vl}}\; M[V/x]}\,\beta \qquad\qquad \frac{M \;\hat{\beta}_{\mathsf{vl}}\; M'}{MN \;\hat{\beta}_{\mathsf{vl}}\; M'N}\,@_{\mathsf{l}}$$

$$\frac{N \;\hat{\beta}_{\mathsf{vl}}\; N'}{V^!N \;\hat{\beta}_{\mathsf{vl}}\; V^!N'}\,@_{\mathsf{rv}}$$

- *the* weak rightmost(-outermost) β_{v}-reduction*, denoted by* $\hat{\beta}_{\mathsf{vr}}$*, whose rules are:*

$$\frac{}{(\lambda x\, M)^! V^! \;\hat{\beta}_{\mathsf{vr}}\; M[V/x]}\,\beta \qquad\qquad \frac{M \;\hat{\beta}_{\mathsf{vr}}\; M'}{MV^! \;\hat{\beta}_{\mathsf{vr}}\; M'V^!}\,@_{\mathsf{lv}}$$

$$\frac{N \;\hat{\beta}_{\mathsf{vr}}\; N'}{MN \;\hat{\beta}_{\mathsf{vr}}\; MN'}\,@_{\mathsf{r}}$$

Remark 130. Clearly, $\hat{\beta}_{\mathsf{vl}}, \hat{\beta}_{\mathsf{vr}} \subseteq \hat{\beta}_{\mathsf{v}}$ (the proof is by induction on the length of the derivation of $M \;\hat{\beta}_{\mathsf{vl}}\; M'$ or $M \;\hat{\beta}_{\mathsf{vr}}\; M'$ respectively). As a consequence, every $\hat{\beta}_{\mathsf{v}}$-normal form is $\hat{\beta}_{\mathsf{vl}}$- and $\hat{\beta}_{\mathsf{vr}}$-normal (in particular, for every value V, $V^!$ is $\hat{\beta}_{\mathsf{vl}}$- and $\hat{\beta}_{\mathsf{vr}}$-normal, see remark 124.4). The converse fails to hold; for instance, if $I = (\lambda x\, x^!)^!$, x_i is a variable and V_i is a value for $i \in \{1, 2\}$, then

$$M = \big((x_1^!)V_1^!\big)\big((II)(I)I\big)(x_2^!)V_2^!$$

is a $\hat{\beta}_{\mathsf{vl}}$- and $\hat{\beta}_{\mathsf{vr}}$-normal form but $M \;\hat{\beta}_{\mathsf{v}}^{+}\; ((x_1^!)V_1^!)(I)(x_2^!)V_2^!$.

We can characterize terms which are not $\hat{\beta}_{\mathsf{vl}}$- or $\hat{\beta}_{\mathsf{vr}}$-normal (see also remark 113.2).

Theorem 131. *Let* M, M' *be terms:* $M \;\hat{\beta}_{\mathsf{vl}}$ *(resp.* $\hat{\beta}_{\mathsf{vr}}$*)* M' *iff there exist* $\ell \in \mathbb{N}$*, values* $V_0, \ldots, V_\ell$ *and terms* $L_0, \ldots, L_\ell, L'_0, \ldots, L'_\ell$ *such that:*

- $(\lambda xN)^!V_0^!L_{01} \cdots L_{0k_0}$ *(resp.* $L_0 = (L_{01}) \cdots (L_{0k_0})(\lambda xN)^!V_0^!$*) and* $(N[V_0/x])L_{01} \cdots L_{0k_0}$ *(resp.* $L'_0 = (L_{01}) \cdots (L_{0k_0})\, N[V_0/x]$*) for some* $k_0 \in \mathbb{N}$ *and terms* $N, L_{01}, \ldots, L_{0k_0}$*;*

- *for every* $1 \leq i \leq \ell$, *we have* $(V_i^!)L_{i-1}L_{i1}\cdots L_{ik_i}$ *(resp.* $L_i = (L_{i1})\cdots(L_{ik_i})(L_{i-1})V_i^!$*) and* $(V_i^!)L'_{i-1}L_{i1}\cdots L_{ik_i}$ *(resp.* $L'_i = (L_{i1})\cdots(L_{ik_i})(L'_{i-1})V_i^!$*) for some* $k_i \in \mathbb{N}$ *and terms* $L_{i1},\ldots,L_{ik_i}$*;*
- $M = L_\ell$ *and* $M' = L'_\ell$*.*

Furthermore, both of these decompositions, if any, are unique.

PROOF. We prove the statement about $\hat{\beta}_{\mathsf{vr}}$, the proof for the $\hat{\beta}_{\mathsf{vl}}$ case is perfectly symmetric.

$\Leftarrow$: Proof by induction on $\ell \in \mathbb{N}$.

If $\ell = 0$, then $M = (L_{01})\cdots(L_{0k_0})(\lambda xN)^!V_0^!$ and $M' = (L_{01})\cdots(L_{0k_0})\,N[V_0/x]$ for some $k_0 \in \mathbb{N}$, value V_0 and terms $N, L_{01},\ldots,L_{0k_0}$; hence $M\ \hat{\beta}_{\mathsf{vr}}\ M'$ by applying the β-rule and k_0 times the $@_{\mathsf{r}}$-rule.

If $\ell > 0$, then $M = (L_{\ell 1})\cdots(L_{\ell k_\ell})(L_{k_\ell - 1})V_\ell^!$ and $M' = (L_{\ell 1})\cdots(L_{\ell k_\ell})(L'_{k_\ell - 1})V_\ell^!$ for some $k_\ell \in \mathbb{N}$, value V_ℓ and terms $L_{\ell 1},\ldots,L_{\ell k_\ell}$; by induction hypothesis, $L_{\ell-1}\ \hat{\beta}_{\mathsf{vr}}\ L'_{\ell-1}$, so $M\ \hat{\beta}_{\mathsf{vr}}\ M'$ by applying the $@_{\mathsf{vr}}$-rule and k_ℓ times the $@_{\mathsf{r}}$-rule.

$\Rightarrow$: The uniqueness is obvious. The proof for the existence is by induction on the length of the derivation of $M\ \hat{\beta}_{\mathsf{v}}\ M'$. Let us consider the last rule.

If the last rule is β, then $M = (\lambda xN)^!V^!$ and $M' = N[V/x]$ for some term N and value V, so we conclude by taking $\ell = 0 = k_0$.

If the last rule is $@_{\mathsf{lv}}$, then $M = NV^!$ and $M' = N'V^!$ for some value V and terms N, N' such that $N\ \hat{\beta}_{\mathsf{vr}}\ N'$; by induction hypothesis, there exist $\ell \in \mathbb{N}$, terms $L_0,\ldots,L_\ell,L'_0,\ldots,L'_\ell$ and values $V_0,\ldots,V_\ell$ such that:

- $L_0 = (L_{01})\cdots(L_{0k_0})(\lambda xL)^!V_0^!$ and $L'_0 = (L_{01})\cdots(L_{0k_0})\,L[V_0/x]$ for some $k_0 \in \mathbb{N}$ and terms $L, L_{01},\ldots,L_{0k_0}$;
- for every $1 \leq i \leq \ell$, $L_i = (L_{i1})\cdots(L_{ik_i})(L_{i-1})V_i^!$ and $L'_i = (L_{i1})\cdots(L_{ik_i})(L'_{i-1})V_i^!$ for some $k_i \in \mathbb{N}$ and terms $L_{i1},\ldots,L_{ik_i}$;
- $N = L_\ell$ and $N' = L'_\ell$.

We conclude by taking $L_{\ell+1} = M$, $L'_{\ell+1} = M'$ and $k_{\ell+1} = 0$.

If the last rule is $@_{\mathsf{r}}$, then $M = N_1N_2$ and $M' = N_1N'_2$ for some terms N_1, N_2, N'_2 such that $N_2\ \hat{\beta}_{\mathsf{vr}}\ N'_2$; by induction hypothesis, there exist $\ell \in \mathbb{N}$, terms $L_0,\ldots,L_\ell,L'_0,\ldots,L'_\ell$ and values $V_0,\ldots,V_\ell$ such that:

- $L_0 = (L_{01})\cdots(L_{0k_0})(\lambda xL)^!V_0^!$ and $L'_0 = (L_{01})\cdots(L_{0k_0})\,L[V_0/x]$ for some $k_0 \in \mathbb{N}$ and terms $L, L_{01},\ldots,L_{0k_0}$;
- for every $1 \leq i \leq \ell$, $L_i = (L_{i1})\cdots(L_{ik_i})(L_{i-1})V_i^!$ and $L'_i = (L_{i1})\cdots(L_{ik_i})(L'_{i-1})V_i^!$ for some $k_i \in \mathbb{N}$ and terms $L_{i1},\ldots,L_{ik_i}$;

- $N_2 = L_\ell$ and $N_2' = L_\ell'$.

We can conclude by replacing in the sequence of L_i's (resp. L_i''s), L_ℓ (resp. L_ℓ') with $M_\ell = (M_0)\cdots(M_{m_\ell})(L_{\ell-1})V_\ell^!$ (resp. $M_\ell' = (M_0)\cdots(M_{m_\ell})(L_{\ell-1}')V_\ell^!$), with $m_\ell = k_\ell + 1$, $M_0 = N_1$ and $M_j = L_{\ell j-1}$ for every $1 \leq j \leq m_\ell$, thus $M = M_\ell$ and $M' = M_\ell'$.

□

Theorem 131 says that in every term there exists at most one $\hat{\beta}_{\mathsf{v}}$-redex that can be reduced by $\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)reduction: theorem 131 might be seen also as a sort of definition of "$\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)redex".

Corollary 132.

1. *There are no critical pairs for the $\hat{\beta}_{\mathsf{vl}}$(resp. $\hat{\beta}_{\mathsf{vr}}$)-reduction: if M, N_1, N_2 are terms such that $M\ \hat{\beta}_{\mathsf{vl}}$ (resp. $\hat{\beta}_{\mathsf{vr}}$) N_1 and $M\ \hat{\beta}_{\mathsf{vl}}$ (resp. $\hat{\beta}_{\mathsf{vr}}$) N_2, then $N_1 = N_2$.*

2. *For every term M, it is $\hat{\beta}_{\mathsf{vl}}$(resp. $\hat{\beta}_{\mathsf{vr}}$)-normal iff either $M = V^!$ for some value V or there exist $\ell \in \mathbb{N}$, a variable x, values $V_0, \ldots, V_\ell$ and terms $L_0, \ldots, L_\ell$ such that:*
   - *$L_0 = (x^!)V_0^! L_{01} \cdots L_{0k_0}$ (resp. $L_0 = (L_{01})\cdots(L_{0k_0})(x^!)V_0^!$) for some $k_0 \in \mathbb{N}$ and terms $L_{01}, \ldots, L_{0k_0}$;*
   - *for every $1 \leq i \leq \ell$, we have $L_i = (V_i^!)L_{i-1}L_{i1}\cdots L_{ik_i}$ (resp. $L_i = (L_{i1})\cdots(L_{ik_i})(L_{i-1})V_i^!$) for some $k_i \in \mathbb{N}$ and terms $L_{i1}, \ldots, L_{ik_i}$;*
   - *$M = L_\ell$.*

3. *Every closed term is a $\hat{\beta}_{\mathsf{v}}$-normal form iff it is a $\hat{\beta}_{\mathsf{vl}}$-normal form iff it is a $\hat{\beta}_{\mathsf{vr}}$-normal form.*

4. *For every closed $\hat{\beta}_{\mathsf{v}}$-normalizable term M, if M' is the $\hat{\beta}_{\mathsf{v}}$-normal form of M then $M\ \hat{\beta}_{\mathsf{vl}}^*\ M'$ and $M\ \hat{\beta}_{\mathsf{vr}}^*\ M'$.*

5. *For every term M, it is $\hat{\beta}_{\mathsf{v}}$-normal iff either $M = V^!$ for some value V or there exist $\ell \in \mathbb{N}$, a variable x, values $V_0, \ldots, V_\ell$ and terms $L_0, \ldots, L_\ell$ such that:*
   - *$L_0 = (x^!)V_0^! L_{01} \cdots L_{0k_0}$ (resp. $L_0 = (L_{01})\cdots(L_{0k_0})(x^!)V_0^!$) for some $k_0 \in \mathbb{N}$ and $\hat{\beta}_{\mathsf{v}}$-normal terms $L_{01}, \ldots, L_{0k_0}$;*
   - *for every $1 \leq i \leq \ell$, we have $L_i = (V_i^!)L_{i-1}L_{i1}\cdots L_{ik_i}$ (resp. $L_i = (L_{i1})\cdots(L_{ik_i})(L_{i-1})V_i^!$) for some $k_i \in \mathbb{N}$ and $\hat{\beta}_{\mathsf{v}}$-normal terms $L_{i1}, \ldots, L_{ik_i}$;*
   - *$M = L_\ell$.*

PROOF.

1. As every non-$\hat{\beta}_{\mathsf{vl}}$(resp. $\hat{\beta}_{\mathsf{vr}}$)-normal term M can be written in a unique way in the forms of theorem 131, there is exactly one $\hat{\beta}_{\mathsf{v}}$-redex in M that can be reduced by $\hat{\beta}_{\mathsf{vl}}$(resp. $\hat{\beta}_{\mathsf{vr}}$)-reduction.

2. It is an immediate consequence of theorem 131 and remark 113.2.

3. Every $\hat{\beta}_{\mathsf{v}}$-normal form is obviously a $\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)normal form (it is not necessary to suppose the term be closed, see remark 130).

   Conversely, let M be a closed $\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)normal term: by corollary 132.2 and since M is closed, the only possibility is that $M = V^!$ for some value V, so M is a $\hat{\beta}_{\mathsf{v}}$-normal form by remark 124.4.

4. Proof by induction on the number $n \in \mathbb{N}$ of steps of the $\hat{\beta}_{\mathsf{v}}$-reduction from M to M' (this number is well-defined by corollary 128.2).

   If $n = 0$ then $M = M'$, hence $M \ \hat{\beta}_{\mathsf{vl}}^* \ M'$ and $M \ \hat{\beta}_{\mathsf{vr}}^* \ M'$ (in 0 steps).

   If $n > 0$ then M is not $\hat{\beta}_{\mathsf{v}}$-normal. By corollary 132.3, M is neither a $\hat{\beta}_{\mathsf{vl}}$- nor a $\hat{\beta}_{\mathsf{vr}}$-normal form, hence there exist terms N_{l} and N_{r} such that $M \ \hat{\beta}_{\mathsf{vl}} \ N_{\mathsf{l}}$ and $M \ \hat{\beta}_{\mathsf{vr}} \ N_{\mathsf{r}}$. As $\hat{\beta}_{\mathsf{vl}}, \hat{\beta}_{\mathsf{vr}} \subseteq \hat{\beta}_{\mathsf{v}}$, both $N_{\mathsf{l}} \ \hat{\beta}_{\mathsf{v}}^* \ M'$ and $N_{\mathsf{r}} \ \hat{\beta}_{\mathsf{v}}^* \ M'$ in $n-1$ steps by corollaries 128.1-2. By induction hypothesis, $N_{\mathsf{l}} \ \hat{\beta}_{\mathsf{vl}}^* \ M'$ and $N_{\mathsf{r}} \ \hat{\beta}_{\mathsf{vr}}^* \ M'$, thus $M \ \hat{\beta}_{\mathsf{vl}}^* \ M'$ and $M \ \hat{\beta}_{\mathsf{vr}}^* \ M'$ (in n steps).

5. If M is $\hat{\beta}_{\mathsf{v}}$-normal then it is $\hat{\beta}_{\mathsf{vl}}$(resp. $\hat{\beta}_{\mathsf{vr}}$)-normal. By corollary 132.2, either $M = V^!$ for some value V or there exist $\ell \in \mathbb{N}$, a variable x, values $V_0, \dots, V_\ell$ and terms $L_0, \dots, L_\ell$ such that:

   - $L_0 = (x^!)V_0^! L_{01} \cdots L_{0k_0}$ (resp. $L_0 = (L_{01}) \cdots (L_{0k_0})(x^!)V_0^!$) for some $k_0 \in \mathbb{N}$ and terms $L_{01}, \dots, L_{0k_0}$;
   - for every $1 \leq i \leq \ell$, we have $L_i = (V_i^!)L_{i-1}L_{i1} \cdots L_{ik_i}$ (resp. $L_i = (L_{i1}) \cdots (L_{ik_i})(L_{i-1})V_i^!$) for some $k_i \in \mathbb{N}$ and terms $L_{i1}, \dots, L_{ik_i}$;
   - $M = L_\ell$;

   moreover, for every $1 \leq i \leq \ell$ and $1 \leq j \leq k_\ell$, L_{ij} is $\hat{\beta}_{\mathsf{v}}$-normal (again since M is $\hat{\beta}_{\mathsf{v}}$-normal) and so L_{ij} is $\hat{\beta}_{\mathsf{vl}}$(resp. $\hat{\beta}_{\mathsf{vr}}$)-normal by remark 130.

   Conversely, let M be a term. If $M = V^!$ for some value V then M is $\hat{\beta}_{\mathsf{v}}$-normal (see remark 124.4).

   If there exist $\ell \in \mathbb{N}$, a variable x, values $V_0, \dots, V_\ell$ and terms $L_0, \dots, L_\ell$ such that:

   - $L_0 = (x^!)V_0^! L_{01} \cdots L_{0k_0}$ (resp. $L_0 = (L_{01}) \cdots (L_{0k_0})(x^!)V_0^!$) for some $k_0 \in \mathbb{N}$ and $\hat{\beta}_{\mathsf{v}}$-normal terms $L_{01}, \dots, L_{0k_0}$,
   - for every $1 \leq i \leq \ell$, we have $L_i = (V_i^!)L_{i-1}L_{i1} \cdots L_{ik_i}$ (resp. $L_i = (L_{i1}) \cdots (L_{ik_i})(L_{i-1})V_i^!$) for some $k_i \in \mathbb{N}$ and $\hat{\beta}_{\mathsf{v}}$-normal terms $L_{i1}, \dots, L_{ik_i}$,

- $M = L_\ell$,

then we show by induction on $\ell \in \mathbb{N}$ that M is $\hat\beta_{\mathsf{v}}$-normal and L_i is an application for every $1 \leq i \leq \ell$. If $\ell = 0$ then $M = L_0 = (x^!)V_0^! L_{01} \cdots L_{0k_0}$ which is an application (for any $k_0 \in \mathbb{N}$) and a $\hat\beta_{\mathsf{v}}$-normal form since L_{0j} is so for every $1 \leq j \leq k_0$ by hypothesis. If $\ell > 0$, then $M = L_\ell = (V_\ell^!)L_{\ell-1}L_{\ell 1} \cdots L_{\ell k_\ell}$ (resp. $M = L_\ell = (L_{\ell 1}) \cdots (L_{\ell k_\ell})(L_{\ell-1})V_\ell^!$); by induction hypothesis, $L_{\ell-1}$ is a $\hat\beta_{\mathsf{v}}$-normal application, hence $(V_\ell^!)L_{\ell-1}$ (resp. $(L_{\ell-1})V_\ell^!$) is a $\hat\beta_{\mathsf{v}}$-normal application; thus M is a $\hat\beta_{\mathsf{v}}$-normal (since $L_{\ell j}$ is so for every $1 \leq j \leq k_\ell$ by hypothesis) application.

□

In other words, according to corollary 132.1, the $\hat\beta_{\mathsf{vl}}$-(resp. $\hat\beta_{\mathsf{vr}}$-)reduction is "strongly deterministic" i.e. it is a partial map from Λ_{t} to Λ_{t}: any term M has at most one $\hat\beta_{\mathsf{vl}}$-(resp. $\hat\beta_{\mathsf{vr}}$-)redex, if any it is the "leftmost-(resp. rightmost-)outermost" β_{v}-redex in M and there exists a unique term M' such that $M \; \hat\beta_{\mathsf{vl}}$ (resp. $\hat\beta_{\mathsf{vr}}$) M', otherwise if M is closed then it is $\hat\beta_{\mathsf{v}}$-normal.

Corollary 132.2 provides a characterization of $\hat\beta_{\mathsf{vl}}$- and $\hat\beta_{\mathsf{vr}}$-normal forms. Corollary 132.5 claims that a term is $\hat\beta_{\mathsf{v}}$-normal iff it is "hereditarily" $\hat\beta_{\mathsf{vl}}$-normal iff it is "hereditarily" $\hat\beta_{\mathsf{vr}}$-normal. This characterizations are more comprehensible by decomposing terms as binary trees (see §3.4).

The equivalences stated by corollary 132.3 have to be read together with the characterization given by proposition 126.

Corollary 132.4 provides two perfectly symmetric "$\hat\beta_{\mathsf{v}}$-normalizing strategies", which can be used for any $\hat\beta_{\mathsf{v}}$-normalizable closed term.

Note that the hypothesis that the term is closed is necessary in corollaries 132.3-4: a term with some free variable might have a $\hat\beta_{\mathsf{v}}$-redex without having neither "$\hat\beta_{\mathsf{vl}}$-" nor "$\hat\beta_{\mathsf{vr}}$-redex", see for example the term M in remark 130, which is a $\hat\beta_{\mathsf{vl}}$- and $\hat\beta_{\mathsf{vr}}$-normal form but not a $\hat\beta_{\mathsf{v}}$-normal form, so its $\hat\beta_{\mathsf{v}}$-normal form cannot be reached by either $\hat\beta_{\mathsf{vl}}$- or $\hat\beta_{\mathsf{vr}}$-reduction.

We introduce now a "$\hat\beta_{\mathsf{v}}$-normalization strategy" for which it is not necessary to assume that the term is closed.

Definition 133 ($\hat\beta_{\mathsf{vt}}$-reduction)**.** *We define a relation $\hat\beta_{\mathsf{vt}} \subseteq \Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}$, called* turbo weak β_{v}-reduction *or $\hat\beta_{\mathsf{vt}}$-reduction, by the following rules:*

$$\frac{}{(\lambda x\, M)^! V^! \;\hat\beta_{\mathsf{vt}}\; M[V/x]}\,\beta \qquad\qquad \frac{M \;\hat\beta_{\mathsf{vt}}\; M' \qquad N \;\hat\beta_{\mathsf{vt}}\; N'}{MN \;\hat\beta_{\mathsf{vt}}\; M'N'}\,@$$

$$\frac{M \;\hat\beta_{\mathsf{vt}}\; M' \qquad N \textit{ is } \hat\beta_{\mathsf{v}}\textit{-normal}}{MN \;\hat\beta_{\mathsf{vt}}\; M'N}\,@_{\mathsf{ln}}$$

$$\frac{N \;\hat\beta_{\mathsf{vt}}\; N' \qquad M \textit{ is } \hat\beta_{\mathsf{v}}\textit{-normal}}{MN \;\hat\beta_{\mathsf{vt}}\; MN'}\,@_{\mathsf{rn}}$$

*A term M is a $\hat{\beta}_{\mathsf{vt}}$-*normal form *or is $\hat{\beta}_{\mathsf{vt}}$-*normal *if there is no term M' such that $M \ \hat{\beta}_{\mathsf{vt}} \ M'$.*

The following proposition clarifies the intuitive meaning of the $\hat{\beta}_{\mathsf{vt}}$-reduction.

Proposition 134. *Let M, M' be terms:*

- *$M \ \hat{\beta}_{\mathsf{vt}} \ M'$ iff M contains at least one $\hat{\beta}_{\mathsf{v}}$-redex and M' is obtained from M by replacing all the $\hat{\beta}_{\mathsf{v}}$-redexes in M with their contractums;*
- *if $M \ \hat{\beta}_{\mathsf{vt}} \ M'$ then $M \ \hat{\beta}_{\mathsf{v}}^{+} \ M'$ in n steps, where n is the number of $\hat{\beta}_{\mathsf{v}}$-redexes in M.*

PROOF.

$\Rightarrow$: Proof by induction on the length of the derivation of $M \ \hat{\beta}_{\mathsf{vt}} \ M'$. Let us consider the last rule of this derivation.

If it is the β-rule, then $M = (\lambda x N)^! V^!$ and $M' = N[V/x]$ for some term N and value V, so M is the only $\hat{\beta}_{\mathsf{v}}$-redex in M, M' is its contractum and $M \ \hat{\beta}_{\mathsf{v}} \ M'$ (in one step) by the β-rule.

If it is the @-rule, then $M = M_1 M_2$ and $M' = M_1' M_2'$ for some terms M_1, M_2, M_1', M_2' with $M_i \ \hat{\beta}_{\mathsf{vt}} \ M_i'$ for $i \in \{1, 2\}$; by induction hypothesis, M_i contains a $\hat{\beta}_{\mathsf{v}}$-redex, M_i' is obtained from M_i by replacing all the $\hat{\beta}_{\mathsf{v}}$-redexes in M_i with their contractums and $M_i \ \hat{\beta}_{\mathsf{v}}^{+} \ M_i'$ in n_i steps where n_i is the number of $\hat{\beta}_{\mathsf{v}}$-redexes in M_i, for $i \in \{1, 2\}$. M is not a $\hat{\beta}_{\mathsf{v}}$-redex, otherwise $M_2 = V^!$ for some value V that is impossible by remark 116.1 since M_2 contains a $\hat{\beta}_{\mathsf{v}}$-redex. Hence M' is obtained from M by replacing all the $\hat{\beta}_{\mathsf{v}}$-redexes in M with their contractums, moreover $(M_1)M_2 \ \hat{\beta}_{\mathsf{v}}^{+} \ (M_1')M_2$ in n_1 steps and $(M_1')M_2 \ \hat{\beta}_{\mathsf{v}}^{+} \ (M_1')M_2'$ in n_2 steps, thus $M \ \hat{\beta}_{\mathsf{v}}^{+} \ M'$ in $n_1 + n_2$ steps, where $n_1 + n_2$ is the number of $\hat{\beta}_{\mathsf{v}}$-redexes in M.

If it is the $@_{\mathsf{ln}}$-rule, then $M = M_1 M_2$ and $M' = M_1' M_2$ for some terms M_1, M_2, M_1' where $M_1 \ \hat{\beta}_{\mathsf{vt}} \ M_1'$ and M_2 is $\hat{\beta}_{\mathsf{v}}$-normal; by induction hypothesis, M_1 contains a $\hat{\beta}_{\mathsf{v}}$-redex, M_1' is obtained from M_1 by replacing all the $\hat{\beta}_{\mathsf{v}}$-redexes in M_1 with their contractums and $M_1 \ \hat{\beta}_{\mathsf{v}}^{+} \ M_1'$ in n steps where n is the number of $\hat{\beta}_{\mathsf{v}}$-redexes in M_1. M_2 contains no $\hat{\beta}_{\mathsf{v}}$-redexes (by remark 124.1) and M is not a $\hat{\beta}_{\mathsf{v}}$-redex (otherwise $M_1 = (\lambda x N)^!$ for some term N that is impossible by remark 116.1 since M_1 contains a $\hat{\beta}_{\mathsf{v}}$-redex). Hence M' is obtained from M by replacing all the $\hat{\beta}_{\mathsf{v}}$-redexes in M with their contractums, moreover $M \ \hat{\beta}_{\mathsf{v}}^{+} \ M'$ in n steps, where n is the number of $\hat{\beta}_{\mathsf{v}}$-redexes in M.

If it is the $@_{\mathsf{rn}}$-rule, the proof is analogous to the previous case.

$\Leftarrow$: Proof by induction on the term M. As M contains a $\hat{\beta}_{\mathsf{v}}$-redex, $M = M_1 M_2$ for some terms M_1 and M_2 by remark 116.1.

If M_1 and M_2 contain no $\hat{\beta}_{\mathsf{v}}$-redexes then M is the only $\hat{\beta}_{\mathsf{v}}$-redex in M, so $M_1 = (\lambda x N)^!$, $M_2 = V^!$ and $M' = N[V/x]$ (since M' is obtained by M by replacing all the $\hat{\beta}_{\mathsf{v}}$-redexes in M with their contractums) for some term N and value V, thus $M \; \hat{\beta}_{\mathsf{vt}} \; M'$ by the β-rule.

If M_1 and M_2 contain a $\hat{\beta}_{\mathsf{v}}$-redex, then by induction hypothesis $M_i \; \hat{\beta}_{\mathsf{vt}} \; M'_i$ where M'_i is obtained from M_i by replacing all the $\hat{\beta}_{\mathsf{v}}$-redexes in M_i with their contractums, for $i \in \{1, 2\}$. Hence $M' = M'_1 M'_2$ since M is not a $\hat{\beta}_{\mathsf{v}}$-redex (otherwise $M_2 = V^!$ for some value V that is impossible by remark 116.1 since M_2 contains a $\hat{\beta}_{\mathsf{v}}$-redex). Thus $M \; \hat{\beta}_{\mathsf{vt}} \; M'$ by the @-rule.

If M_1 contains a $\hat{\beta}_{\mathsf{v}}$-redex and M_2 does not, then M_2 is $\hat{\beta}_{\mathsf{v}}$-normal (by remark 116.1) and $M_1 \; \hat{\beta}_{\mathsf{vt}} \; M'_1$ where M'_1 is obtained from M_1 by replacing all the $\hat{\beta}_{\mathsf{v}}$-redexes in M_1 with their contractums (by induction hypothesis). Hence $M' = M'_1 M_2$ since M_2 contains no $\hat{\beta}_{\mathsf{v}}$-redexes (by remark 124.1) and M is not a $\hat{\beta}_{\mathsf{v}}$-redex (otherwise $M_1 = (\lambda x N)^!$ for some term N that is impossible by remark 116.1 since M_1 contains a $\hat{\beta}_{\mathsf{v}}$-redex). Thus $M \; \hat{\beta}_{\mathsf{vt}} \; M'$ by the $@_{\mathsf{ln}}$-rule.

If M_2 contains a $\hat{\beta}_{\mathsf{v}}$-redex and M_1 does not, the proof is analogous to the previous case, in particular we conclude that $M \; \hat{\beta}_{\mathsf{vt}} \; M'$ by applying the $@_{\mathsf{rn}}$-rule.

□

Corollary 135.

1. *Every term is $\hat{\beta}_{\mathsf{v}}$-normal iff it is $\hat{\beta}_{\mathsf{vt}}$-normal.*
2. *Terms of the shape $V^!$ for some value V are $\hat{\beta}_{\mathsf{vt}}$-normal.*
3. *There are no critical pairs for the $\hat{\beta}_{\mathsf{vt}}$-reduction: if M, N_1, N_2 are terms such that $M \; \hat{\beta}_{\mathsf{vt}} \; N_1$ and $M \; \hat{\beta}_{\mathsf{vt}} \; N_2$, then $N_1 = N_2$.*

PROOF.

1. By proposition 134 and remark 124.1.
2. By corollary 135.1 ($\Rightarrow$) and remark 124.4.
3. Immediate consequence of proposition 134.

□

Corollary 135.3 says that the $\hat{\beta}_{\mathsf{vt}}$-reduction is "strongly deterministic" (i.e. it is a partial map from Λ_{t} to Λ_{t}): if a term M is not $\hat{\beta}_{\mathsf{v}}$-normal, then there exists a unique term M' such that $M \; \hat{\beta}_{\mathsf{vt}} \; M'$.

Obviously, the fact that $M\ \hat{\beta}_{\mathsf{vt}}\ M'$ does not entail that M' is $\hat{\beta}_{\mathsf{v}}$-normal, since the $\hat{\beta}_{\mathsf{vt}}$-reduction might create new $\hat{\beta}_{\mathsf{v}}$-redexes. For instance

$$(\lambda x(x^{!})x^{!})^{!}(\lambda x\, x^{!})^{!}\ \hat{\beta}_{\mathsf{vt}}\ (\lambda x\, x^{!})^{!}(\lambda x\, x^{!})^{!}$$

where $(\lambda x\, x^{!})^{!}(\lambda x\, x^{!})^{!}$ is not $\hat{\beta}_{\mathsf{v}}$-normal.

Theorem 136. *For every $\hat{\beta}_{\mathsf{v}}$-normalizable term M, if M' is the $\hat{\beta}_{\mathsf{v}}$-normal form of M then $M\ \hat{\beta}^{*}_{\mathsf{vt}}\ M'$.*

Proof. By induction on the number $m \in \mathbb{N}$ of steps of the $\hat{\beta}_{\mathsf{v}}$-reduction $M\ \hat{\beta}^{*}_{\mathsf{v}}\ M'$.

If $m = 0$ then $M = M'$, therefore $M\ \hat{\beta}^{*}_{\mathsf{vt}}\ M'$ by reflexivity of $\hat{\beta}^{*}_{\mathsf{vt}}$.

If $m > 0$ then M is not $\hat{\beta}_{\mathsf{v}}$-normal, thus M is not $\hat{\beta}_{\mathsf{vt}}$-normal by corollary 135.1, hence there exists a term N such that $M\ \hat{\beta}_{\mathsf{vt}}\ N$. By proposition 134, $M\ \hat{\beta}^{+}_{\mathsf{v}}\ N$ in $n > 0$ steps, with $n \leq m$ by corollary 128.2. By corollary 128.1 there exists a term N' such that $M'\ \hat{\beta}^{*}_{\mathsf{v}}\ N'$ and $N\ \hat{\beta}^{*}_{\mathsf{v}}\ N'$, so $M' = N'$ since M' is $\hat{\beta}_{\mathsf{v}}$-normal and thus $N\ \hat{\beta}^{*}_{\mathsf{v}}\ M'$ in $m - n < m$ by corollary 128.2. Therefore $N\ \hat{\beta}^{*}_{\mathsf{vt}}\ M'$ by induction hypothesis, hence $M\ \hat{\beta}^{*}_{\mathsf{vt}}\ M'$ by transitivity of $\hat{\beta}^{*}_{\mathsf{vt}}$. ☐

Theorem 136 provides a “$\hat{\beta}_{\mathsf{v}}$-normalizing strategy”, which can be used for any $\hat{\beta}_{\mathsf{v}}$-normalizable (not necessarily closed) term.

3.1.3 Some problems with η-reduction

Definition 137 (η-reduction)**.** *We define a relation $\eta \subseteq (\Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}) \cup (\Lambda_{\mathsf{v}} \times \Lambda_{\mathsf{v}})$, called* η-reduction*, by the following rules:*

$$\frac{}{(\lambda x\, M x^{!})^{!}\ \eta\ M}\ \eta \qquad \frac{M\ \eta\ M'}{MN\ \eta\ M'N}\ @_{\mathsf{l}} \qquad \frac{N\ \eta\ N'}{MN\ \eta\ MN'}\ @_{\mathsf{r}}$$

$$\frac{M\ \eta\ M'}{\lambda x M\ \eta\ \lambda x M'}\ \lambda \qquad \frac{V\ \eta\ W}{V^{!}\ \eta\ W^{!}}\ !$$

where in η-rule the variable x is not free in the term M.

Remark 138. Let $M = (\lambda d\, y^{!})^{!}\big(\lambda z\big((\lambda x\, x^{!}x^{!})^{!}(\lambda x\, x^{!}x^{!})^{!}\big)z^{!}\big)^{!}$: then $M\ \eta\ (\lambda d\, y^{!})\big((\lambda x\, x^{!}x^{!})^{!}(\lambda x\, x^{!}x^{!})^{!}\big)$ $M\ \hat{\beta}_{\mathsf{v}}\ y^{!}$ where $y^{!}$ and $(\lambda d\, y^{!})^{!}\big((\lambda x\, x^{!}x^{!})^{!}(\lambda x\, x^{!}x^{!})^{!}\big)$ are $\beta_{\mathsf{v}}\eta$-normal forms. Therefore, neither the $\beta_{\mathsf{v}}\eta$-reduction nor the $\hat{\beta}_{\mathsf{v}}\eta$-reduction are confluent.

3.2 A “completion” of β_{v}-reduction

A solid theory of call-by-value λ-calculus requires an operational characterization of solvability, i.e. to find a strategy which computes the results of the represented functions. Following [PR04], a term t is CBV-solvable whenever there is an head context H s.t. $H[t]\ \rightarrow^{*}_{\beta_v}\ I$ where $I = \lambda x x$ and $\rightarrow_{\beta_v}$

is the call-by-value β-reduction in Plotkin's λ_{β_v}-calculus. An operational characterization has been provided in [PR99, PR04] but, unfortunately, it is obtained through call-by-name β-reduction, which is disappointing and not satisfying. If it is not possible to get an internal characterization, i.e. one which uses the rules of the calculus itself, then there is an inherent weakness in the rewriting rules of the calculus. For Plotkin's call-by-value λ_{β_v}-calculus [Plo75] it is indeed the case, let us illustrate the point with an example. Let $\Delta = \lambda x\,(x)x$. There is no head context sending (via β_v-reduction) the following term to the identity:

$$t = \big((\lambda y \Delta)(x)z\big)\Delta$$
$$t' = (\Delta)(\lambda y \Delta)(x)z$$

and – as a consequence – t and t' should be unsolvable and divergent in a good call-by-value calculus, while they are in β_v-normal form. The weakness of β_v-reduction is a fact widely recognized and accepted, indeed there have been many proposals of alternative call-by-value λ-calculi, see for instance [Mog89, Hof95, DL07, HZ09, AP12]. All these different versions of call-by-value λ-calculi extend the syntax of λ-calculus with an explicit substitution constructor $t\{u/x\}$ (which is equivalent to use `let`...`in` expressions) defined in the syntax, but these substitutions are just delayed, they are not propagated in a small-steps way.

In particular, Accattoli and Paolini introduced in [AP12] the value-substitution lambda-calculus, a simple call-by-value λ-calculus with explicit substitutions borrowing ideas from Herbelin and Zimmerman's lambda-CBV calculus ([HZ09]) and from Accattoli and Kesner's structural lambda-calculus ([AK10]), both with explicit substitutions. Interestingly, in this new setting, Accattoli and Paolini characterized solvable terms as those terms having normal form with respect to a suitable contextual closure of its (call-by-value) reduction rules, thus improving over the previous characterization.

We aim at showing that we can characterize CBV-solvable terms without using explicit substitutions, by only adding some simple reduction rules in our syntax. These supplementary rules are nothing but an orientation of the two orientable rules σ_1 and σ_3 generating the σ_{v}-equivalence (see section 5.2): they are a reformulation in our syntax without explicit substitutions of the let_{let}- and let_{app}-rules of the Herbelin's and Zimmerman's calculus (see [HZ09]).

Definition 139 (σ- and σ'-reduction). *σ_1 is the contextual closure of the binary relation $\to_{\sigma_1}$ on $\Lambda_{\mathtt{t}}$ defined by the following rule:*

$$\frac{}{(\lambda x\, M)^! N L \;\to_{\sigma_1}\; (\lambda x M L)^! N}\,\sigma_1$$

where M, N, L are terms and $x \notin \mathsf{fv}(L)$.

σ_3 is the contextual closure of the binary relation $\to_{\sigma_3}$ on $\Lambda_{\mathtt{t}}$ defined by the following rule:

$$\frac{}{(V^!)\big((\lambda x\, L)^!\big)N \;\to_{\sigma_3}\; (\lambda x V^! L)^! N}\,\sigma_3$$

where N and L are terms, V is a value and $x \notin \mathsf{fv}(V)$.

σ_3' is the contextual closure of the binary relation $\to_{\sigma_3'}$ on Λ_{t} defined by the following rule:

$$\frac{}{(M)((\lambda x\, L)^!)N \;\to_{\sigma_3'}\; (\lambda x M L)^! N}\,\sigma_3'$$

where M, N, L are terms and $x \notin \mathsf{fv}(M)$.

The σ_{v}-reduction *(resp.* σ_{v}'-reduction*) is $\sigma_{\mathsf{v}} = \sigma_1 \cup \sigma_3$ (resp. $\sigma_{\mathsf{v}}' = \sigma_1 \cup \sigma_3'$).*

The variable condition on σ_1-, σ_3- and σ_3'-rules can be always fulfilled by α-conversion.

The σ_3-rule is a weakened version of the σ_3'-rule, i.e. it is the σ_3'-rule limited to the case where $M = V^!$ for some value V.

σ_1- and σ_3'-rules above are just an orientation of respective rules in the definition of σ_{v}-equivalence. Note the left-right symmetry of σ_1- and σ_3'-rules: in σ_1(resp. σ_3')-rule, the σ-redex is an an application of a β-redex (resp. term) to a term (resp. β-redex). In remark 148, we will see a reason to like the σ_3-rule more than its generalization σ_3'.

Remark 140.

1. By remark 118, one has $\sigma_{\mathsf{v}}, \sigma_{\mathsf{v}}' \subseteq (\Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}) \cup (\Lambda_{\mathsf{v}} \times \Lambda_{\mathsf{v}})$. Moreover, σ_{v} (resp. σ_{v}') is the contextual closure of the relations $\to_{\sigma_3}$ (resp. $\to_{\sigma_3'}$) and $\to_{\sigma_1}$ (the proof is by straightforward induction on the derivations).

2. It is immediate to check that for every expression E, there exists an expression E' such that $E\ \sigma_{\mathsf{v}}\ E'$ (resp. $E\ \sigma_{\mathsf{v}}'\ E'$) iff E contains a σ_{v}(resp. σ_{v}')-redex. Therefore, an expression is σ_{v}(resp. σ_{v}')-normal iff it contains no σ_{v}(resp. σ_{v}')-redex.

3. It is easy to verify that for all expressions E, E', $E\ \sigma_{\mathsf{v}}\ E'$ (resp. $E\ \sigma_{\mathsf{v}}'\ E'$) iff E' is obtained from E by replacing exactly one σ_{v}(resp. σ_{v}')-redex in E with its contractum.

4. Clearly, $\sigma_{\mathsf{v}} \subseteq \sigma_{\mathsf{v}}'$. The converse does not hold, for instance take $M = (z_1^!)z_2^!$, $N = (y_1^!)y_2^!$ and $L = (M)\big((\lambda x_1 x_2^!)^!\big)N$ where $x_1, x_2, y_1, y_2, z_1, z_2$ are pairwise distinct variables: then L is σ_{v}-normal but $L\ \sigma_3'\ (\lambda x_1 (M) x_2^!)^! N$.

We can merge the σ_{v}- and σ_{v}'-reduction into the β_{v}- and $\hat{\beta}_{\mathsf{v}}$-reduction, in order to get a sort of "completion" of the β_{v}- and $\hat{\beta}_{\mathsf{v}}$-reduction.

Definition 141. *We set $\beta_{\mathsf{v}\sigma} = \beta_{\mathsf{v}} \cup \sigma_{\mathsf{v}}$ (resp. $\beta_{\mathsf{v}\sigma'} = \beta_{\mathsf{v}} \cup \sigma_{\mathsf{v}}'$), called $\beta_{\mathsf{v}\sigma}$(resp. $\beta_{\mathsf{v}\sigma'}$)*-reduction*, and $\hat{\beta}_{\mathsf{v}\sigma} = \hat{\beta}_{\mathsf{v}} \cup \sigma_{\mathsf{v}}$ (resp. $\hat{\beta}_{\mathsf{v}\sigma'} = \hat{\beta}_{\mathsf{v}} \cup \sigma_{\mathsf{v}}'$), called $\hat{\beta}_{\mathsf{v}\sigma}$*(resp. $\hat{\beta}_{\mathsf{v}\sigma'}$)-reduction*.*

Intuitively, σ_{v}-reduction might enable a β_{v}-redex in an expression E which is hidden by the inessential sequential structure of E. For instance, if $N = (z^!)z^!$ and $M = (\lambda y(\lambda x\, z_0^!)^! x^!)^! N$ where x, y, z, z_0 are pairwise distinct variables, then $M_1 = (\lambda y(\lambda x\, z_0^!)^!)^! N x^!$ and $M_2 = (\lambda x\, z_0^!)^!((\lambda y\, x^!)^!)N$ are β_{v}-normal, but $M_1\ \sigma_{\mathsf{v}}\ M$ (by the σ_1-rule) and $M_2\ \sigma_{\mathsf{v}}\ M$ (by the σ_3-rule), where M is not β_{v}-normal.

Remark 142.

1. By remark 118, one has $\beta_{\mathsf{v}\sigma}, \hat{\beta}_{\mathsf{v}\sigma}, \beta_{\mathsf{v}\sigma'}, \hat{\beta}_{\mathsf{v}\sigma'} \subseteq (\Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}) \cup (\Lambda_{\mathsf{v}} \times \Lambda_{\mathsf{v}})$.
2. It is immediate to check that for every expression E (resp. term M), there exists an expression E' (resp. a term M') such that $E\ \beta_{\mathsf{v}\sigma}\ E'$ (resp. $M\ \hat{\beta}_{\mathsf{v}\sigma}\ M'$) iff E (resp. M) contains a $\beta_{\mathsf{v}\sigma}$(resp. $\hat{\beta}_{\mathsf{v}\sigma}$)-redex. Therefore, an expression (resp. a term) is $\beta_{\mathsf{v}\sigma}$(resp. $\hat{\beta}_{\mathsf{v}\sigma}$)-normal iff it contains no $\beta_{\mathsf{v}\sigma}$(resp. $\hat{\beta}_{\mathsf{v}\sigma}$)-redex. Analogous considerations hold for $\beta_{\mathsf{v}\sigma'}$(resp. $\hat{\beta}_{\mathsf{v}\sigma'}$)-reduction.
3. It is easy to verify that for all expressions E, E' (resp. terms M, M'), $E\ \beta_{\mathsf{v}\sigma}\ E'$ (resp. $M\ \hat{\beta}_{\mathsf{v}\sigma}\ M'$) iff E' (resp. M') is obtained from E (resp. M) by replacing exactly one $\beta_{\mathsf{v}\sigma}$(resp. $\hat{\beta}_{\mathsf{v}\sigma}$)-redex in E (resp. M) with its contractum. Analogous considerations hold for $\beta_{\mathsf{v}\sigma'}$(resp. $\hat{\beta}_{\mathsf{v}\sigma'}$)-reduction.
4. Clearly, $\hat{\beta}_{\mathsf{v}\sigma} \subseteq \beta_{\mathsf{v}\sigma} \subseteq \beta_{\mathsf{v}\sigma'}$ and $\hat{\beta}_{\mathsf{v}\sigma} \subseteq \hat{\beta}_{\mathsf{v}\sigma'} \subseteq \beta_{\mathsf{v}\sigma'}$. The converses do not hold.

We prove now a confluence property for $\beta_{\mathsf{v}\sigma}$ and $\hat{\beta}_{\mathsf{v}\sigma}$. For this purpose, we use a commutation property of β_{v}- and σ_{v}-reductions and the strong normalization of σ'_{v}.

Definition 143. *With every expression E are associated two* measures $\mathsf{size}'(E), \#_{\mathsf{w}}(E) \in \mathbb{N}$*, defined by induction on E as follows:*

- $\mathsf{size}'(x) = 2$*;*
- $\mathsf{size}'(\lambda x M) = \mathsf{size}'(M) + 1$*;*
- $\mathsf{size}'(V^!) = \mathsf{size}'(V)$*;*
- $\mathsf{size}'(MN) = \mathsf{size}'(M) + \mathsf{size}'(N)$*.*

- $\#_{\mathsf{w}}(x) = 1$*;*
- $\#_{\mathsf{w}}(\lambda x M) = \#_{\mathsf{w}}(M) + \mathsf{size}'(M)$*;*
- $\#_{\mathsf{w}}(V^!) = \#_{\mathsf{w}}(V)$*;*
- $\#_{\mathsf{w}}(MN) = \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(N) + 2\mathsf{size}'(M)\mathsf{size}'(N) - 1$*.*

$\#_{\mathsf{w}}(M)$ is a little modification of the following measure: the sum of the weights of nodes in the usual tree-like representation of M, where the weight of a node n in the tree-like representation of M is the difference between $\mathsf{size}(M)$ and the number of λ-nodes above n.

Remark 144. $\mathsf{size}'(E) \geq 2$ and $\#_{\mathsf{w}}(E) \geq 1$ for any expression E. The proof is by a straightforward induction on the expression E.

Lemma 145. *Let E and E' be expressions. If $E \;\sigma'_{\mathsf{v}}\; E'$ then $\#_{\mathsf{w}}(E) > \#_{\mathsf{w}}(E')$ and $\mathsf{size}'(E) = \mathsf{size}'(E')$.*

Proof. By induction on the length of the derivation of $E \;\sigma'_{\mathsf{v}}\; E'$. Let us consider the last rule of this derivation.

If it is the σ_1-rule then $E = (\lambda x M)^! N L$ and $E' = (\lambda x M L)^! N$ for some terms N, M and L, so

$$\begin{aligned}
\#_{\mathsf{w}}(E) &= \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(L) + \#_{\mathsf{w}}(N) + \mathsf{size}'(M) + 2\mathsf{size}'(N) + 2\mathsf{size}'(L) + \\
&\quad + 2\mathsf{size}'(M)\mathsf{size}'(N) + 2\mathsf{size}'(M)\mathsf{size}'(L) + 2\mathsf{size}'(L)\mathsf{size}'(N) - 2 \\
\#_{\mathsf{w}}(E') &= \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(L) + \#_{\mathsf{w}}(N) + \mathsf{size}'(M) + 2\mathsf{size}'(N) + \mathsf{size}'(L) + \\
&\quad + 2\mathsf{size}'(M)\mathsf{size}'(N) + 2\mathsf{size}'(M)\mathsf{size}'(L) + 2\mathsf{size}'(L)\mathsf{size}'(N) - 2 \\
&= \#_{\mathsf{w}}(E) - \mathsf{size}'(L)
\end{aligned}$$

hence $\#_{\mathsf{w}}(E) > \#_{\mathsf{w}}(E')$ by remark 144. Moreover, $\mathsf{size}'(E) = \mathsf{size}'(M) + \mathsf{size}'(L) + \mathsf{size}'(N) + 1 = \mathsf{size}'(E')$.

If it is the σ'_3-rule then $E = (M)((\lambda x L)^!)N$ and $E' = (\lambda x M L)^! N$ for some terms N, M and L, so

$$\begin{aligned}
\#_{\mathsf{w}}(E) &= \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(L) + \#_{\mathsf{w}}(N) + 2\mathsf{size}(M) + 2\mathsf{size}(N) + \mathsf{size}(L) + \\
&\quad + 2\mathsf{size}(M)\mathsf{size}(N) + 2\mathsf{size}(M)\mathsf{size}(L) + 2\mathsf{size}(L)\mathsf{size}(N) - 2 \\
\#_{\mathsf{w}}(E') &= \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(L) + \#_{\mathsf{w}}(N) + \mathsf{size}'(M) + 2\mathsf{size}'(N) + \mathsf{size}'(L) + \\
&\quad + 2\mathsf{size}'(M)\mathsf{size}'(N) + 2\mathsf{size}'(M)\mathsf{size}'(L) + 2\mathsf{size}'(L)\mathsf{size}'(N) - 2 \\
&= \#_{\mathsf{w}}(E) - \mathsf{size}'(M)
\end{aligned}$$

hence $\#_{\mathsf{w}}(E) > \#_{\mathsf{w}}(E')$ by remark 144. Moreover, $\mathsf{size}'(E) = \mathsf{size}'(M) + \mathsf{size}'(L) + \mathsf{size}'(N) + 1 = \mathsf{size}'(E')$.

If it is the λ-rule then $E = \lambda x M$ and $E' = \lambda x M'$ for some terms M and M' such that $M \;\sigma'_{\mathsf{v}}\; M'$, thus $\#_{\mathsf{w}}(E) = \#_{\mathsf{w}}(M) + \mathsf{size}'(M)$ and $\#_{\mathsf{w}}(E') = \#_{\mathsf{w}}(M') + \mathsf{size}'(M')$; by induction hypothesis, we have $\#_{\mathsf{w}}(M) > \#_{\mathsf{w}}(M')$ and $\mathsf{size}'(M) = \mathsf{size}'(M')$, therefore $\#_{\mathsf{w}}(E) > \#_{\mathsf{w}}(M') + \mathsf{size}'(M) = \#_{\mathsf{w}}(E')$ and $\mathsf{size}'(E) = \mathsf{size}'(M) + 1 = \mathsf{size}'(E')$.

If it is the !-rule then $E = V^!$ and $E' = V'^!$ for some values V and V' such that $V \;\sigma'_{\mathsf{v}}\; V'$, thus $\#_{\mathsf{w}}(E) = \#_{\mathsf{w}}(V)$ and $\#_{\mathsf{w}}(E') = \#_{\mathsf{w}}(V')$; by induction hypothesis, we have $\#_{\mathsf{w}}(V) > \#_{\mathsf{w}}(V')$ and $\mathsf{size}'(V) = \mathsf{size}'(V')$, therefore $\#_{\mathsf{w}}(E) > \#_{\mathsf{w}}(E')$ and $\mathsf{size}'(E) = \mathsf{size}'(V) = \mathsf{size}'(E')$.

If it is the $@_{\mathsf{l}}$(resp. $@_{\mathsf{r}}$)-rule then $E = MN$ and $E' = M'N$ (resp. $E = MN'$) for some terms M, N and M' (resp. N') such that $M\ \sigma'_{\mathsf{v}}\ M'$ (resp. $N\ \sigma'_{\mathsf{v}}\ N'$), thus $\#_{\mathsf{w}}(E) = \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(N) + 2\mathsf{size}'(M)\mathsf{size}'(N) - 1$ and $\#_{\mathsf{w}}(E') = \#_{\mathsf{w}}(M') + \#_{\mathsf{w}}(N) + 2\mathsf{size}'(M')\mathsf{size}'(N) - 1$ (resp. $\#_{\mathsf{w}}(E') = \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(N') + 2\mathsf{size}'(M)\mathsf{size}'(N') - 1$); by induction hypothesis, we have $\#_{\mathsf{w}}(M) > \#_{\mathsf{w}}(M')$ (resp. $\#_{\mathsf{w}}(N) > \#_{\mathsf{w}}(N')$) and $\mathsf{size}'(M) = \mathsf{size}'(M')$ (resp. $\mathsf{size}'(N) = \mathsf{size}'(N')$), therefore $\#_{\mathsf{w}}(E) > \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(N) + 2\mathsf{size}'(M')\mathsf{size}'(N) - 1 = \#_{\mathsf{w}}(E')$ (resp. $\#_{\mathsf{w}}(E) > \#_{\mathsf{w}}(M) + \#_{\mathsf{w}}(N) + 2\mathsf{size}'(M)\mathsf{size}'(N') - 1 = \#_{\mathsf{w}}(E')$) and $\mathsf{size}'(E) = \mathsf{size}'(M) + \mathsf{size}'(N) = \mathsf{size}'(E')$. □

Proposition 146. *σ'_{v} (and in particular σ_{v}) is strongly normalizing.*

PROOF. It is an immediate consequence of the previous lemma. □

Lemma 147. *σ_{v} is locally confluent.*

PROOF. By induction on the expression E such that $E\ \sigma_{\mathsf{v}}\ E_i$ for $i \in \{1, 2\}$. The only interesting case are:

- if $E = \big((\lambda x M)^!((\lambda y L)^!)N\big)L'$ with $E\ \sigma_{\mathsf{v}}\ (\lambda x M L')^!((\lambda y L)^!)N = E_1$ (by reducing the σ_1-redex E) and $E\ \sigma_{\mathsf{v}}\ (\lambda y(\lambda x M)^! L)^! N L' = E_2$ (by reducing the σ_3-redex $(\lambda x M)^!((\lambda y L)^!)N$ in E), then

  $$E_2\ \sigma_{\mathsf{v}}\ (\lambda y(\lambda x M)^! L L')^! N\ \sigma_{\mathsf{v}}\ (\lambda y(\lambda x M L')^! L)^! N = E'$$

  (by reducing twice a σ_1-redex) and $E_1\ \sigma_{\mathsf{v}}\ E'$ (by reducing the σ_3-redex E_1);

- if $E = (V^!)((\lambda x L)^!)((\lambda x' L')^!)N$ with $E\ \sigma_{\mathsf{v}}\ (V^!)(\lambda x'(\lambda x L)^! L')^! N = E_1$ (by reducing the σ_3-redex $(\lambda x L)^!((\lambda x' L')^!)N$ in E) and

  $$E\ \sigma_{\mathsf{v}}\ (\lambda x V^! L)^!((\lambda x' L')^!)N = E_2$$

  (by reducing the σ_3-redex E), then

  $$E_1\ \sigma_{\mathsf{v}}\ (\lambda x'(V)(\lambda x L)L')N\ \sigma_{\mathsf{v}}\ (\lambda x'(\lambda x V^! L)^! L')^! N = E'$$

  (by reducing twice a σ_3-redex) and $E_2\ \sigma_{\mathsf{v}}\ E'$ (by reducing the σ_3-redex E_2).

□

Remark 148. σ'_{v} and $\beta_{\mathsf{v}\sigma'}$ are not locally confluent and so neither confluent. For instance, take $N_i = (z_i^!)z_i^!$ for $i \in \{1, 2\}$ and $M = \big((\lambda x_1\, y_1^!)^! N_1\big)\big((\lambda x_2\, y_2^!)^!\big)N_2$ where $x_1, x_2, y_1, y_2, z_1, z_2$ are pairwise distinct variables: M is β_{v}-normal and it contains no σ_3-redexes but $M\ \sigma'_{\mathsf{v}}\ (\lambda x_2(\lambda x_1\, y_1^!)^! N_1 y_2^!)^! N_2$ (because of the σ'_3-rule) which contains only a σ_1-redex, and $M\ \sigma'_{\mathsf{v}}\ (\lambda x_1(y_1^!)(\lambda x_2 y_2^!)^! N_2)^! N_1$ (because of the σ_1-rule) which contains only a σ_3-redex, so M reduces to two different σ'_{v}-normal forms, $(\lambda x_2(\lambda x_1\, y_1^! y_2^!)^! N_1)^! N_2$ and $(\lambda x_1(\lambda x_2\, y_1^! y_2^!)^! N_2)^! N_1$.

We conjecture that σ' is confluent modulo the equivalence relation on Λ_{CBV} generated by the following binary relation $\sim_{\sigma_4}$ on Λ_{t} defined by:

$$(\lambda x_1(\lambda x_2 M)^! N_2)^! N_1 \sim_{\sigma_4} (\lambda x_2(\lambda x_1 M)^! N_1)^! N_2$$

where $x_2 \notin \mathsf{fv}(N_1)$ and $x_1 \notin \mathsf{fv}(N_2)$.

Proposition 149. σ_{v} *is confluent.*

Proof. By lemma 147, proposition 146 and Newman’s lemma. □

We recall a well-known result on term rewriting systems.

Lemma 150 (Hindley–Rosen)**.** *Let* $\to_1$ *and* $\to_2$ *be two binary relations on a set* X*. If they are both confluent and they commute, i.e. if* $t \to_1^* u_1$ *and* $t \to_2^* u_2$ *then there exists* s *such that* $u_1 \to_2^* s$ *and* $u_2 \to_1^* s$*, then* $\to_1 \cup \to_2$ *is confluent.*

Proof. See proposition 3.3.5 in [Bar84]. □

Lemma 151. *Let* E *and* E' *be expressions, let* V *and* V' *be values and let* x *be a variable:*

1. *if* $V \; \sigma_{\mathsf{v}} \; V'$ *then* $E[V/x] \; \sigma_{\mathsf{v}}^* \; E[V'/x]$*;*
2. *if* $E \; \sigma_{\mathsf{v}} \; E'$ *then* $E[V/x] \; \sigma_{\mathsf{v}} \; E'[V/x]$*.*

Proof.

1. By induction on the expression E.

   If $E = x$, then $E[V/x] = V$ and $E[V'/x] = V'$, so $E[V/x] \; \sigma_{\mathsf{v}}^* \; E[V'/x]$ by hypothesis.

   If $E = y \neq x$, then $E[V/x] = y = E[V'/x]$, then $E[V/x] \; \sigma_{\mathsf{v}}^* \; E[V'/x]$ by reflexivity of σ_{v}^*.

   If $E = \lambda y M$ for some term M, then we can suppose without loss of generality that $y \neq x$, hence $E[V/x] = \lambda y M[V/x]$ and $E[V'/x] = \lambda y M[V'/x]$; by induction hypothesis, $M[V/x] \; \sigma_{\mathsf{v}}^* \; M[V'/x]$ and thus $E[V/x] \; \sigma_{\mathsf{v}}^* \; E[V'/x]$ since σ_{v}^* passes to context.

   If $E = W^!$ for some value W, then $E[V/x] = (W[V/x])^!$ and $E[V'/x] = (W[V'/x])^!$; by induction hypothesis, $W[V/x] \; \sigma_{\mathsf{v}}^* \; W[V'/x]$ and thus $E[V/x] \; \sigma_{\mathsf{v}}^* \; E[V'/x]$ since σ_{v}^* passes to context.

   If $E = MN$ for some terms M, N, then $E[V/x] = M[V/x]N[V/x]$ and $E[V'/x] = M[V'/x]N[V'/x]$; $M[V/x] \; \sigma_{\mathsf{v}}^* \; M[V'/x]$ and $N[V/x] \; \sigma_{\mathsf{v}}^* \; N[V'/x]$ by induction hypothesis, therefore $E[V/x] \; \sigma_{\mathsf{v}}^*$ $M[V'/x]N[V/x] \; \sigma_{\mathsf{v}}^* \; E[V'/x]$ since σ_{v}^* passes to context.

2. By induction on the length of the derivation of $E\ \sigma_{\mathsf{v}}\ E'$. Let us consider the last rule of this derivation.

   If it is the σ_1-rule, then $E = (\lambda y M)^{!}NL$ and $E' = (\lambda y ML)^{!}N$ with $y \notin \mathsf{fv}(L)$; we can suppose without loss of generality that $y \notin \mathsf{fv}(V) \cup \{x\}$, hence $E[V/x] = (\lambda y M[V/x])^{!}N[V/x]L[V/x]$ and $E'[V/x] = (\lambda y M[V/x]L[V/x])^{!}N[V/x]$, therefore $E[V/x]\ \sigma_{\mathsf{v}}\ E'[V/x]$ by the σ_1-rule, since $y \notin (\mathsf{fv}(L) \smallsetminus \{x\}) \cup \mathsf{fv}(V) = \mathsf{fv}(L[V/x])$.

   If it is the σ_3-rule, then $E = (W^{!})((\lambda y L)^{!})N$ and $E' = (\lambda y W^{!}L)^{!}N$ with $y \notin \mathsf{fv}(W)$; we can suppose without loss of generality that $y \notin \mathsf{fv}(V) \cup \{x\}$, so $E[V/x] = (W[V/x])^{!}((\lambda y L[V/x])^{!})N[V/x]$ and $E'[V/x] = (\lambda y (W[V/x])^{!}L[V/x])^{!}N[V/x]$, therefore $E[V/x]\ \sigma_{\mathsf{v}}\ E'[V/x]$ by the σ_3-rule, since $y \notin (\mathsf{fv}(W) \smallsetminus \{x\}) \cup \mathsf{fv}(V) = \mathsf{fv}(W[V/x])$.

   If it is the λ-rule then $E = \lambda y M$ and $E' = \lambda y M'$ for some terms M and M' with $M\ \sigma_{\mathsf{v}}\ M'$; we can suppose without loss of generality that $y \notin \mathsf{fv}(V) \cup \{x\}$, hence $E[V/x] = \lambda y M[V/x]$ and $E'[V/x] = \lambda y M'[V/x]$; by induction hypothesis, $M[V/x]\ \sigma_{\mathsf{v}}\ M'[V/x]$ and thus $E[V/x]\ \sigma_{\mathsf{v}}\ E'[V/x]$ by the λ-rule.

   If it is the !-rule then $E = W^{!}$ and $E' = W'^{!}$ for some values W such that $W\ \sigma_{\mathsf{v}}\ W'$, so $E[V/x] = (W[V/x])^{!}$ and $E[V'/x] = (W'[V/x])^{!}$; by induction hypothesis, $W[V/x]\ \sigma_{\mathsf{v}}\ W'[V/x]$ and thus $E[V/x]\ \sigma_{\mathsf{v}}\ E'[V/x]$ by the !-rule.

   If it is the $@_{\mathsf{l}}$ (resp. $@_{\mathsf{r}}$) then $E = MN$ and $E = M'N$ (resp. $E' = MN'$) for some terms M, N and M' (resp. N') such that $M\ \sigma_{\mathsf{v}}\ M'$ (resp. $N\ \sigma_{\mathsf{v}}\ N'$), so $E[V/x] = M[V/x]N[V/x]$ and $E'[V/x] = M'[V/x]N[V/x]$ (resp. $E'[V/x] = M[V/x]N'[V/x]$); by induction hypothesis, $M[V/x]\ \sigma_{\mathsf{v}}\ M'[V/x]$ (resp. $N[V/x]\ \sigma_{\mathsf{v}}\ N'[V/x]$), so $E[V/x]\ \sigma_{\mathsf{v}}\ E'[V/x]$ by the $@_{\mathsf{l}}$(resp. $@_{\mathsf{r}}$)-rule.

□

Lemma 152.

1. *β_{v} (resp. $\hat{\beta}_{\mathsf{v}}$) and σ_{v} quasi-strongly commute i.e. if $M\ \sigma_{\mathsf{v}}\ N_1$ and $M\ \beta_{\mathsf{v}}\ N_2$ (resp. $M\ \hat{\beta}_{\mathsf{v}}\ N_2$) then there exists M' such that $N_2\ \sigma_{\mathsf{v}}^{*}\ M'$ and $N_1\ \beta_{\mathsf{v}}\ M'$ (resp. $N_1\ \hat{\beta}_{\mathsf{v}}\ M'$).*

2. *β_{v} (resp. $\hat{\beta}_{\mathsf{v}}$) and σ_{v} commute.*

PROOF.

1. We prove the statement about β_{v} by induction on M. The only interesting cases are:

   - if $M = (\lambda x N)^{!}V^{!}L$ with $M\ \sigma_1\ (\lambda x NL)^{!}V^{!} = N_1$ and $M\ \beta_{\mathsf{v}}\ (N[V/x])L = N_2$, then $N_1\ \beta_{\mathsf{v}}\ N_2$ since $x \notin \mathsf{fv}(L)$.

- if $M = (W^!)\big((\lambda x N)^!\big)V^!$ with $M\ \sigma_3\ (\lambda x W N)^! V^! = N_1$ and $M\ \beta_{\mathsf{v}}\ (W^!)N[V/x] = N_2$, then $N_1\ \beta_{\mathsf{v}}\ N_2$ since $x \notin \mathsf{fv}(W)$.
- if $M = \big((\lambda y P)^!((\lambda x N)^!)V^!\big)L$ with $M\ \sigma_1\ (\lambda y P L)^!\big((\lambda x N)^!\big)V^! = M_1$ and $M\ \beta_{\mathsf{v}}\ (\lambda y P)^! N[V/x]L$ $= M_2$, then $M_1\ \beta_{\mathsf{v}}\ (\lambda y P L)^! N[V/x] = M'$ and $M_2\ \sigma_1\ M'$.
- if $M = (\lambda x N)^! V^!$ with $M\ \sigma_{\mathsf{v}}\ (\lambda x N)^! V'^! = N_1$, $M\ \beta_{\mathsf{v}}\ N[V/x] = N_2$ and $V\ \sigma_{\mathsf{v}}\ V'$, then $N_1\ \beta_{\mathsf{v}}\ N[V'/x] = M'$ and so $N_2\ \sigma_{\mathsf{v}}^*\ M'$ by lemma 151.1.
- if $M = (\lambda x N)^! V^!$ with $M\ \sigma_{\mathsf{v}}\ (\lambda x N')^! V^! = N_1$, $M\ \beta_{\mathsf{v}}\ N[V/x] = N_2$ and $N\ \sigma_{\mathsf{v}}\ N'$, then $N_1\ \beta_{\mathsf{v}}\ N'[V/x] = M'$ and so $N_2\ \sigma_{\mathsf{v}}\ M'$ by lemma 151.2.

As regards the statement about $\hat{\beta}_{\mathsf{v}}$, it is not proved explicitly because it is enough to observe that in the previous proof whenever the step is $\hat{\beta}_{\mathsf{v}}$ then we can close the commutation diagram with one $\hat{\beta}_{\mathsf{v}}$-reduction step.

2. We prove the following stronger statement, in order to apply the right induction hypothesis: given $R \in \{\beta_{\mathsf{v}}, \hat{\beta}_{\mathsf{v}}\}$, if $L\ \sigma_{\mathsf{v}}^*\ N$ and $L\ R^m\ M$ then there exists L' such that $M\ \sigma_{\mathsf{v}}^*\ L'$ and $N\ R^m\ L'$. Let $L\ \sigma_{\mathsf{v}}^n\ N$: the proof is by induction on (m, n) with the lexicographical order on $\mathbb{N}^2$.

   If $m = 0$ or $n = 0$, we conclude easily.

   Let $m, n > 0$: there exist N', M' such that $L\ \sigma\ N'$, $L\ R\ M'$, $N'\ \sigma_{\mathsf{v}}^{n-1}\ N$ and $M'\ R^{m-1}\ M$. By lemma 152.1 applied to L, there exists L'' such that $N'\ R\ L''$ and $M\ \sigma_{\mathsf{v}}^*\ L''$. By induction hypothesis applied to M', there exists M'' such that $M\ \sigma_{\mathsf{v}}^*\ M''$ and $L''\ R^{m-1}\ M''$; thus $N'\ R^m\ N'$, so there exists L' such that $M''\ \sigma_{\mathsf{v}}^*\ L'$ and $N\ R^m\ L'$ by applying the induction hypothesis to N', therefore $M\ \sigma_{\mathsf{v}}^*\ L'$. □

Theorem 153. *$\beta_{\mathsf{v}\sigma}$ and $\hat{\beta}_{\mathsf{v}\sigma}$ are confluent.*

PROOF. By proposition 149 and lemmas 150 and 152, since β_{v} (see [Ehr12]) and $\hat{\beta}_{\mathsf{v}}$ (see corollary 128.1) are confluent. □

3.3 Simulation of Accattoli and Paolini's calculus and solvability

We present the Accattoli and Paolini's call-by-value λ-calculus with explicit substitutions, λ_{vsub}, introduced in [AP12]. This calculus can be seen as a merging of two already existing λ-calculi, the Herbelin and Zimmerman's one (a call-by-value λ-calculus with explicit substitutions, see [HZ09]) and the Accattoli and Kesner's one (a call-by-name λ-calculus with explicit substitutions and a very elegant notion of reduction, see [AK10, AK12]).

The following definitions 154, 155 and 158 are exactly the same as in [AP12].

Definition 154 (Syntax of λ_{vsub})**.** *We define the sets* $\lambda_{\mathsf{vsub}}^{\mathsf{terms}}$ *(of* λ_{vsub}-terms*) and* $\lambda_{\mathsf{vsub}}^{\mathsf{values}}$ *(of* λ_{vsub}-values *by mutual induction as follows:*

$$\begin{array}{llll} \lambda_{\mathsf{vsub}}^{\mathsf{term}} & s,t ::= \mathsf{v} \mid (s)t \mid s\{t/x\} & \lambda_{\mathsf{vsub}}\textit{-terms} \\ \lambda_{\mathsf{vsub}}^{\mathsf{value}} & \mathsf{u},\mathsf{v} ::= x \mid \lambda x\, s & \lambda_{\mathsf{vsub}}\textit{-values} \end{array}$$

A constructor of the form $\{t/x\}$ *is an* explicit substitution *and a term of the form* $s\{t/x\}$ *is a* term with an explicit substitution. *For any* $n \in \mathbb{N}$*, a tuple* $(\{t_1/x_1\},\dots,\{t_n/x_n\})$ *of explicit substitutions is denoted by* $\{t_1/x_1\}\dots\{t_n/x_n\}$*.*

Notice that any λ_{vsub}-value (i.e. a variable or an abstraction) is a λ_{vsub}-term.

There are two kinds of binder: $\lambda x\, t$ and $t\{u/x\}$, both binding x in t. All λ_{vsub}-terms are considered up to α-equivalence. The capture-avoiding substitution of values replacing variables is extended to λ_{vsub}-terms with explicit substitutions by setting:

$$s\{t/y\}[\mathsf{v}/x] = s[\mathsf{v}/x]\{t[\mathsf{v}/x]/y\}$$

for every λ_{vsub}-terms s and t, λ_{vsub}-value v and variable x with $y \notin \mathsf{fv}(\mathsf{v}) \cup \{x\}$.

Definition 155 ($\to_{\lambda_{\mathsf{vsub}}}$-, $\to_{\mathsf{w}}$- and $\to_{\mathsf{sw}}$-reduction)**.** *Let* R *be a binary relation on* $\lambda_{\mathsf{vsub}}^{\mathsf{term}}$*.*

The contextual closure of R *is the binary relation* R' *on* $\lambda_{\mathsf{vsub}}^{\mathsf{term}}$ *defined by applying, a finite number of times, the following rules:*

$$\dfrac{s\ R\ t}{s\ R'\ t}\,R \qquad \dfrac{s\ R'\ s'}{st\ R'\ s't}\,@_{\mathsf{l}} \qquad \dfrac{t\ R'\ t'}{st\ R'\ st'}\,@_{\mathsf{r}}$$

$$\dfrac{s\ R'\ s'}{\lambda x\, s\ R'\ \lambda x\, s'}\,\lambda \qquad \dfrac{s\ R'\ s'}{s\{t/x\}\ R'\ s'\{t/x\}}\,sub_{\mathsf{l}}$$

$$\dfrac{t\ R'\ t'}{s\{t/x\}\ R'\ s\{t'/x\}}\,sub_{\mathsf{r}}$$

The applicative closure of R *is the binary relation* R' *on* $\lambda_{\mathsf{vsub}}^{\mathsf{term}}$ *defined by applying, a finite number of times, the following rules:*

$$\dfrac{s\ R\ t}{s\ R'\ t}\,R \qquad \dfrac{s\ R'\ s'}{st\ R'\ s't}\,@_{\mathsf{l}} \qquad \dfrac{t\ R'\ t'}{st\ R'\ st'}\,@_{\mathsf{r}}$$

$$\dfrac{s\ R'\ s'}{s\{t/x\}\ R'\ s'\{t/x\}}\,sub_{\mathsf{l}} \qquad \dfrac{t\ R'\ t'}{s\{t/x\}\ R'\ s\{t'/x\}}\,sub_{\mathsf{r}}$$

$\to_{\lambda_{\mathsf{vsub}}}$ *(resp.* $\to_{\mathsf{w}}$*), called the* $\to_{\lambda_{\mathsf{vsub}}}$-reduction *(resp.* weak $\to_{\lambda_{\mathsf{vsub}}}$*-reduction) is the contextual (resp. applicative) closure of the binary relation* $\mapsto_{\lambda_{\mathsf{vsub}}}$ *(resp.* $\mapsto_{\mathsf{w}}$*) on* $\lambda_{\mathsf{vsub}}^{\mathsf{term}}$ *defined by the following rules:*

$$\frac{}{(\lambda x\, s)\mathsf{L}t \mapsto_{\lambda_{\mathsf{vsub}}} s\{t/x\}\mathsf{L}}\ \mathsf{d}\beta \qquad \frac{}{s\{\mathsf{vL}/x\} \mapsto_{\lambda_{\mathsf{vsub}}} s[\mathsf{v}/x]\mathsf{L}}\ \mathsf{sv}$$

where $\mathsf{L} = \{t_1/x_1\}\dots\{t_n/x_n\}$ *for some* $n \in \mathbb{N}$ *and* $\mathsf{v} \in \lambda_{\mathsf{vsub}}^{\mathsf{value}}$*, moreover* $x_i \notin \mathsf{fv}(t)$ *(resp.* $x_i \notin \mathsf{fv}(s)$*) for every* $1 \le i \le n$ *in the* $\mathsf{d}\beta$*(resp.* sv*)-rule.*

The stratified-weak λ_{vsub}-reduction *is the binary relation* $\to_{\mathsf{sw}}$ *on* $\lambda_{\mathsf{vsub}}^{\mathsf{term}}$ *defined by applying, a finite number of times, the following rules:*

$$\frac{s \to_{\mathsf{w}} t}{s \to_{\mathsf{sw}} t}\ \mathsf{w} \qquad \frac{s \to_{\mathsf{sw}} s'}{st \to_{\mathsf{sw}} s't}\ @_{\mathsf{l}}$$

$$\frac{s \to_{\mathsf{sw}} s'}{\lambda x\, s \to_{\mathsf{sw}} \lambda x\, s'}\ \lambda \qquad \frac{s \to_{\mathsf{sw}} s'}{s\{t/x\} \to_{\mathsf{sw}} s'\{t/x\}}\ sub_{\mathsf{l}}$$

The $\mathsf{d}\beta$-rule (coming from the call-by-name λ-calculus with explicit substitutions introduced in [AK12]) extend the notion of β-redex: indeed, given some λ_{vsub}-terms s, t and u, $(\lambda x\, s)\{t/y\}u \to_{\lambda_{\mathsf{vsub}}} s\{u/x\}\{t/y\}$ by the $\mathsf{d}\beta$-rule. This means that the $\mathsf{d}\beta$-rule acts a distance. In the proof-nets representation of λ_{vsub}-terms this apparent distance is avoided, the $\mathsf{d}\beta$-rule is perfectly local from the proof-nets point of view.

The sv-rule impose the "call-by-value" constraint in λ_{vsub}, because only an explicit substitution $\{\mathsf{vL}/x\}$ (where v is a λ_{vsub}-value and L is a finite sequence of explicit substitutions) can perform an effective substitution of the occurrences of x for v. The fact that $s\{\mathsf{vL}/x\} \to_{\lambda_{\mathsf{vsub}}} s[\mathsf{v}/x]\mathsf{L}$ by the sv-rule means that also the sv-rule acts at a distance.

Remark 156. Clearly, $\to_{\mathsf{w}} \subseteq \to_{\mathsf{sw}} \subseteq \to_{\lambda_{\mathsf{vsub}}}$.

Stratified-weak λ_{vsub}-reduction extends weak λ_{vsub}-reduction allowing reduction under top-level abstractions, which have the important property that cannot be duplicated nor erased.

Proposition 157. $\to_{\lambda_{\mathsf{vsub}}}$*,* $\to_{\mathsf{sw}}$ *and* $\to_{\mathsf{w}}$ *are confluent.*

PROOF. See corollary 1 and lemma 11 in [AP12]. ☐

In λ_{vsub} two terms can have the same behavior and differ only for the position of explicit substitutions, which is not relevant because they do not block $\to_{\lambda_{\mathsf{vsub}}}$-redexes. This is formalized in a precise way by o-equivalence on λ_{vsub}-terms.

Definition 158 (o-equivalence)**.** *For every* $i \in \{1,2,3,4\}$*, let* $\sim_{\mathsf{o}_i}$ *be the contextual closure of the relation* o_i *defined by the* o_i*-rule:*

$$\frac{}{t\{s/x\}\{u/y\}\ \mathsf{o}_1\ t\{u/y\}\{s/x\}}\ \mathsf{o}_1 \quad \textit{where } x \notin \mathsf{fv}(u) \textit{ and } y \notin \mathsf{fv}(s)$$

$$\frac{}{t\,u\{s/x\}\ \mathsf{o}_2\ (tu)\{s/x\}}\ \mathsf{o}_2 \quad \textit{where } x \notin \mathsf{fv}(t)$$

$$\frac{}{t\{s/x\}\,u\ \mathsf{o}_3\ (tu)\{s/x\}}\ \mathsf{o}_3 \quad \textit{where } x \notin \mathsf{fv}(u)$$

$$\frac{}{t\{s\{u/y\}/x\}\ \mathsf{o}_4\ t\{s/x\}\{u/y\}}\ \mathsf{o}_4 \quad \textit{where } y \notin \mathsf{fv}(t)$$

We set $\sim_{\mathsf{o}} = \bigcup_{i=1}^{4} \sim_{\mathsf{o}_i}$. *The* o-equivalence *is the symmetric and reflexive-transitive closure of* $\sim_{\mathsf{o}}$, *i.e.* $\equiv_{\mathsf{o}} = (\sim_{\mathsf{o}}^{T})^{*}$.

Remark that $\equiv_{\mathsf{o}}$ is an equivalence relation on λ_{vsub} which allows the commutation of explicit substitutions with every constructor of λ_{vsub} except abstractions.

We remind a standard notion of rewriting theory and some well-known results about it.

Definition 159 (Strong bisimulation). *Let* X *be a set and let* $\to_X$ *be a binary relation on* X.

A strong bisimulaton for $(X, \to_X)$ *is a binary symmetric relation* $\equiv$ *on* X *such that, for every* $s, s', t \in X$, *if* $s \equiv t$ *and* $s \to_X s'$ *then there exists* $t' \in X$ *such that* $t \to_X t'$ *and* $s' \equiv t'$.

Given an equivalence relation $\equiv$ *on* X*:*

- *we denote by* $\to_{X/\equiv}$ *the binary relation on* X *defined by:* $s \to_{X/\equiv} s'$ *iff there exists* $t, u \in X$ *such that* $s \equiv t \to_X u \equiv s'$;
- *we set* $\leftrightarrow_{X/\equiv} = (\to_X^{T} \cup \equiv)^{*}$;
- $\to_X$ is Church-Rosser modulo $\equiv$ *if for every* $s, s' \in X$ *such that* $s \leftrightarrow_{X/\equiv} s'$, *there exist* $t, t' \in X$ *such that* $s \to_X^{*} t$, $t \equiv t'$ *and* $s' \to_X^{*} t'$.

Remark 160. Let X be a set, let $\to_X$ be a binary relation on X and let $\equiv$ be a strong bisimulation for $(X, \to_X)$. If $\to_X$ is Church-Rosser modulo $\equiv$ then $\to_X$ is confluent modulo $\equiv$, i.e. for every $s, s', u, u' \in X$ such that $s \equiv u$, $s \to_X^{*} s'$ and $u \to_X^{*} u'$, there exist $t, t' \in X$ such that $s \to_X^{*} t$, $t \equiv t'$ and $s' \to_X^{*} t'$.

Lemma 161. *Let* X *be a set, let* $\to_X$ *be a binary relation on* X *and let* $\equiv$ *be a strong bisimulation for* $(X, \to_X)$ *which is an equivalence relation on* X.

1. $\equiv$ *can be postponed to* $\to_X$, *i.e. for every* $s, s' \in X$, *if* $s \to_{X/\equiv}^{*} s'$ *then there exists* $t \in X$ *such that* $s \to_X^{*} t \equiv s'$.
2. *If* $\to_X$ *is confluent then* $\to_{X/\equiv}$ *is confluent and* $\to_X$ *is Church-Rosser modulo* $\equiv$.

Proof. See for example [Acc11], pp. 86-87. ☐

Accattoli and Paolini showed that:

Lemma 162. $\equiv_{\mathsf{o}}$ *is a strong bisimulation for both* $(\lambda_{\mathsf{vsub}}^{\mathsf{term}}, \to_{\lambda_{\mathsf{vsub}}})$ *and* $(\lambda_{\mathsf{vsub}}^{\mathsf{term}}, \to_{\mathsf{sw}}$ *).*

Proof. See lemma 12 in [AP12]. ☐

Therefore, according to proposition 157 ad lemmas 161 and 162, $\equiv_{\mathsf{o}}$ can be postponed to $\to_{\lambda_{\mathsf{vsub}}}$ and $\to_{\mathsf{sw}}$, $\to_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}}$ and $\to_{\mathsf{sw}/\equiv_{\mathsf{o}}}$ are confluent and $\to_{\lambda_{\mathsf{vsub}}}$ and $\to_{\mathsf{sw}}$ are Church-Rosser modulo $\equiv_{\mathsf{o}}$.

There is a natural way to simulate the λ_{vsub}-calculus into our Λ_{CBV}: it is based on the following translation which transforms a λ_{vsub}-term with an explicit substitution in a β_{v}-redex in Λ_{CBV}.

Definition 163. *With every λ_{vsub}-term t there is associated a term $(t)^{\diamondsuit} \in \Lambda_{\mathsf{CBV}}$ (also denoted by $t^{\diamondsuit}$) as follows (the definition is by induction on $t \in \lambda_{\mathsf{vsub}}^{\mathsf{term}}$):*

- $(x)^{\diamondsuit} = x^{!}$;
- $(st)^{\diamondsuit} = s^{\diamondsuit}t^{\diamondsuit}$;
- $(\lambda x\, t)^{\diamondsuit} = (\lambda x\, t^{\diamondsuit})^{!}$;
- $(s\{t/x\})^{\diamondsuit} = (\lambda x\, s^{\diamondsuit})^{!}t^{\diamondsuit}$

With every λ_{vsub}-value v there is associated a value $(\mathsf{v})^{\blacklozenge} \in \Lambda_{\mathsf{CBV}}$ (also denoted by $\mathsf{v}^{\blacklozenge}$) as follows:

- $(x)^{\blacklozenge} = x$;
- $(\lambda x\, s)^{\blacklozenge} = \lambda x\, s^{\diamondsuit}$.

Remark 164. It is immediate to check that:

1. for every $\mathsf{v} \in \lambda_{\mathsf{vsub}}^{\mathsf{value}}$, one has $\mathsf{v}^{\diamondsuit} = (\mathsf{v}^{\blacklozenge})^{!}$;
2. for every $t \in \lambda_{\mathsf{vsub}}^{\mathsf{term}}$, one has $\mathsf{fv}(t) = \mathsf{fv}(t^{\diamondsuit}) = \mathsf{fv}(t^{\blacklozenge})$ (the proof is by straightforward induction on $t \in \lambda_{\mathsf{vsub}}^{\mathsf{term}}$).

Lemma 165. *For every λ_{vsub}-term t, λ_{vsub}-value v and variable x, one has $(t[\mathsf{v}/x])^{\diamondsuit} = t^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$.*

Proof. By induction on the λ_{vsub}-term t.

If $t = x$ then $(t[\mathsf{v}/x])^{\diamondsuit} = \mathsf{v}^{\diamondsuit} = (\mathsf{v}^{\blacklozenge})^{!} = t^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$, by remark 164.

If $t = y$ for some variable $y \neq x$, then $(t[\mathsf{v}/x])^{\diamondsuit} = y^{!} = t^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$.

If $t = \lambda y\, s$ for some λ_{vsub}-term s, then $(s[\mathsf{v}/x])^{\diamondsuit} = s^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$ by induction hypothesis, moreover we can suppose without loss of generality that $y \notin \mathsf{fv}(\mathsf{v}) \cup \{x\}$, thus $(t[\mathsf{v}/x])^{\diamondsuit} = (\lambda y\, (s[\mathsf{v}/x])^{\diamondsuit})^{!} = (\lambda y\, s^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x])^{!} = t^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$.

If $t = s\{u/y\}$ for some λ_{vsub}-terms s and u, then $(s[\mathsf{v}/x])^{\diamondsuit} = s^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$ and $(u[\mathsf{v}/x])^{\diamondsuit} = u^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$ by induction hypothesis, moreover we can suppose without loss of generality that $y \notin \mathsf{fv}(\mathsf{v}) \cup \{x\}$, hence by lemma 165

$$\begin{aligned}(t[\mathsf{v}/x])^{\diamondsuit} &= (s[\mathsf{v}/x]\{u[\mathsf{v}/x]/y\})^{\diamondsuit} = (\lambda y\, (s[\mathsf{v}/x])^{\diamondsuit})^{!}(u[\mathsf{v}/x])^{\diamondsuit} \\ &= (\lambda y\, s^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x])^{!}u^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x] = ((\lambda y\, s^{\diamondsuit})^{!}u^{\diamondsuit})[\mathsf{v}^{\blacklozenge}/x] = t^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]\end{aligned}$$

If $t = su$ for some λ_{vsub}-terms s and u, then $(s[\mathsf{v}/x])^{\diamondsuit} = s^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$ and $(u[\mathsf{v}/x])^{\diamondsuit} = u^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$ by induction hypothesis, so $(t[\mathsf{v}/x])^{\diamondsuit} = s[\mathsf{v}/x]^{\diamondsuit}u[\mathsf{v}/x]^{\diamondsuit} = s^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]u^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x] = t^{\diamondsuit}[\mathsf{v}^{\blacklozenge}/x]$. □

In order to simulate in Λ_{CBV} all the reductions seen in definition 155, we introduce the following notions.

Definition 166. *The* weak $\beta_{\mathsf{v}\sigma}$-reduction *is a binary relation* β_{w} *on* Λ_{CBV} *defined by applying a finite number of times the following rules:*

$$\frac{M\ \hat{\beta}_{\mathsf{v}\sigma}\ M'}{M\ \beta_{\mathsf{w}}\ N}\,\hat{\beta}_{\mathsf{v}\sigma} \qquad \frac{M\ \beta_{\mathsf{w}}\ M'}{(\lambda xM)^{!}N\ \beta_{\mathsf{w}}\ (\lambda xM')^{!}N}\,\mathit{red}_{\mathsf{l}}$$

$$\frac{M\ \beta_{\mathsf{w}}\ M'}{MN\ \beta_{\mathsf{w}}\ M'N}\,@_{\mathsf{l}} \qquad \frac{N\ \beta_{\mathsf{w}}\ N'}{MN\ \beta_{\mathsf{w}}\ MN'}\,@_{\mathsf{r}}$$

The stratified-weak $\beta_{\mathsf{v}\sigma}$-reduction *is a binary relation* β_{sw} *on* Λ_{CBV} *defined by applying a finite number of times the following rules:*

$$\frac{M\ \beta_{\mathsf{w}}\ M'}{M\ \beta_{\mathsf{sw}}\ N}\,\mathsf{w} \qquad \frac{M\ \beta_{\mathsf{sw}}\ M'}{MN\ \beta_{\mathsf{sw}}\ M'N}\,@_{\mathsf{l}} \qquad \frac{M\ \beta_{\mathsf{sw}}\ M'}{\lambda xM\ \beta_{\mathsf{sw}}\ \lambda xM'}\,\lambda$$

$$\frac{V\ \beta_{\mathsf{sw}}\ V'}{V^{!}\ \beta_{\mathsf{sw}}\ V'^{!}}\,!$$

Remark 167. By remark 118, one has $\beta_{\mathsf{w}}, \beta_{\mathsf{sw}} \subseteq (\Lambda_{\mathsf{t}} \times \Lambda_{\mathsf{t}}) \cup (\Lambda_{\mathsf{v}} \times \Lambda_{\mathsf{v}})$. Furthermore, $\hat{\beta}_{\mathsf{v}\sigma} \subseteq \beta_{\mathsf{w}} \subseteq \beta_{\mathsf{sw}} \subseteq \beta_{\mathsf{v}\sigma}$. The converses do not hold.

Proposition 168 (Simulation of λ_{vsub} in Λ_{CBV})**.** *Let* $s, t \in \lambda_{\mathsf{vsub}}^{\mathsf{term}}$.

1. *If* $s \to_{\lambda_{\mathsf{vsub}}} t$ *then* $s^{\Diamond}\ \beta_{\mathsf{v}\sigma}^{*}\ t^{\Diamond}$.
2. *If* $s \to_{\mathsf{w}} t$ *then* $s^{\Diamond}\ \beta_{\mathsf{w}}^{*}\ t^{\Diamond}$.
3. *If* $s \to_{\mathsf{sw}} t$ *then* $s^{\Diamond}\ \beta_{\mathsf{sw}}^{*}\ t^{\Diamond}$.

Proof.

1. By induction on the length of the derivation of $s \to_{\lambda_{\mathsf{vsub}}} t$. Let us consider the last rule of this derivation.

   If it is the $\mathsf{d}\beta$-rule, then $s = (\lambda x\, u)\mathsf{L}w$ and $t = u\{w/x\}\mathsf{L}$ for some λ_{vsub}-terms u, w and tuple of explicit substitutions $\mathsf{L} = \{t_1/x_1\}\ldots\{t_n/x_n\}$ with $n \in \mathbb{N}$. We can suppose without loss of generality that $x_i \notin \mathsf{fv}(w)$ for every $1 \leq i \leq n$, so

   $$s^{\Diamond} = (\lambda x_n \ldots (\lambda x_1 (\lambda x\, u^{\Diamond})^{!})^{!} t_1^{\Diamond} \ldots)^{!} t_n^{\Diamond} w^{\Diamond}\ \sigma_1^{n}\ (\lambda x_n \ldots (\lambda x_1 (\lambda x\, u^{\Diamond})^{!} w^{\Diamond})^{!} t_1^{\Diamond} \ldots)^{!} t_n^{\Diamond} = t^{\Diamond}$$

   (in particular $s^{\Diamond} = t^{\Diamond}$ if $n = 0$), therefore $s^{\Diamond}\ \hat{\beta}_{\mathsf{v}\sigma}^{*}\ t^{\Diamond}$ and thus $s^{\Diamond}\ \beta_{\mathsf{w}}^{*}\ t^{\Diamond}$ (by the $\hat{\beta}_{\mathsf{v}\sigma}$-rule for β_{w}) and $s^{\Diamond}\ \beta_{\mathsf{v}\sigma}^{*}\ t^{\Diamond}$.

   If it is the sv-rule, then $s = u\{\mathsf{vL}/x\}$ and $t = u[\mathsf{v}/x]\mathsf{L}$ for some λ_{vsub}-term u, λ_{vsub}-value v and tuple of explicit substitutions $\mathsf{L} = \{t_1/x_1\}\ldots\{t_n/x_n\}$ with $n \in \mathbb{N}$. We can suppose without loss of generality that $x_i \notin \mathsf{fv}(u) \cup \{x\}$ for every $1 \leq i \leq n$, hence by lemma 165 and remark 164

   $$\begin{aligned} s^{\Diamond} &= (\lambda x\, u^{\Diamond})^{!}\big((\lambda x_n \ldots (\lambda x_1 \mathsf{v}^{\Diamond})^{!} t_1^{\Diamond} \ldots)^{!}\big) t_n^{\Diamond}\ \sigma_3^{n}\ (\lambda x_n \ldots (\lambda x_1 (\lambda x\, u^{\Diamond})^{!} (\mathsf{v}^{\blacklozenge})^{!})^{!} t_1^{\Diamond} \ldots)^{!} t_n^{\Diamond} \\ &\quad \beta_{\mathsf{v}}\ (\lambda x_n \ldots (\lambda x_1 u^{\Diamond}[\mathsf{v}^{\blacklozenge}/x])^{!} t_1^{\Diamond} \ldots)^{!} t_n^{\Diamond} = (\lambda x_n \ldots (\lambda x_1 (u\{\mathsf{v}/x\})^{\Diamond})^{!} t_1^{\Diamond} \ldots)^{!} t_n^{\Diamond} = t^{\Diamond}. \end{aligned}$$

therefore $s^{\Diamond}\ \beta^*_{\mathsf{w}}\ t^{\Diamond}$ and thus $s^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ t^{\Diamond}$.

If it is the $@_{\mathsf{l}}$-rule (resp. $@_{\mathsf{r}}$-rule) then $s = uw$ and $t = u'w$ (resp. $t = uw'$) for some λ_{vsub}-terms u, u' and w (resp. w') such that $u \to_{\lambda_{\mathsf{vsub}}} u'$ (resp. $w \to_{\lambda_{\mathsf{vsub}}} w'$). By induction hypothesis, $u^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ u'^{\Diamond}$ (resp. $w^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ w'^{\Diamond}$), so $s^{\Diamond} = u^{\Diamond}w^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ u'^{\Diamond}w^{\Diamond} = t^{\Diamond}$ (resp. $s^{\Diamond} = u^{\Diamond}w^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ u^{\Diamond}w'^{\Diamond} = t^{\Diamond}$) since $\beta^*_{\mathsf{v}\sigma}$ passes to context.

If it is the λ-rule then $s = \lambda x\, u$ and $t = \lambda x\, u'$ for some λ_{vsub}-terms u and u' such that $u \to_{\lambda_{\mathsf{vsub}}} u'$. By induction hypothesis, $u^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ u'^{\Diamond}$, thus $s^{\Diamond} = (\lambda x\, u^{\Diamond})^{!}\ \beta^*_{\mathsf{v}\sigma}\ (\lambda x\, u'^{\Diamond})^{!} = t^{\Diamond}$ since $\beta^*_{\mathsf{v}\sigma}$ passes to context.

If it is the $\mathit{sub}_{\mathsf{l}}$-rule (resp. $\mathit{sub}_{\mathsf{r}}$-rule) then $s = u\{w/x\}$ and $t = u'\{w/x\}$ (resp. $t = u\{w'/x\}$) for some λ_{vsub}-terms u, u' and w (resp. w') such that $u \to_{\lambda_{\mathsf{vsub}}} u'$ (resp. $w \to_{\lambda_{\mathsf{vsub}}} w'$). By induction hypothesis, $u^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ u'^{\Diamond}$ (resp. $w^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ w'^{\Diamond}$), so $s^{\Diamond} = (\lambda x\, u^{\Diamond})^{!}w^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ (\lambda x\, u'^{\Diamond})^{!}w^{\Diamond} = t^{\Diamond}$ (resp. $s^{\Diamond} = (\lambda x\, u^{\Diamond})^{!}w^{\Diamond}\ \beta^*_{\mathsf{v}\sigma}\ (\lambda x\, u^{\Diamond})^{!}w'^{\Diamond} = t^{\Diamond}$) since $\beta^*_{\mathsf{v}\sigma}$ passes to context.

2. By induction on the length of the derivation of $s \to_{\mathsf{w}} t$. Let us consider the last rule of this derivation.

   If it is the $\mathsf{d}\beta$- or sv-rule, then we have seen in the proof of proposition 168.1 that $s^{\Diamond}\ \beta^*_{\mathsf{w}}\ t^{\Diamond}$.

   If it is the $@_{\mathsf{l}}$-rule (resp. $@_{\mathsf{r}}$-rule) then $s = uw$ and $t = u'w$ (resp. $t = uw'$) for some λ_{vsub}-terms u, u' and w (resp. w') such that $u \to_{\mathsf{w}} u'$ (resp. $w \to_{\mathsf{w}} w'$). By induction hypothesis, $u^{\Diamond}\ \beta^*_{\mathsf{w}}\ u'^{\Diamond}$ (resp. $w^{\Diamond}\ \beta^*_{\mathsf{w}}\ w'^{\Diamond}$), so $s^{\Diamond} = u^{\Diamond}w^{\Diamond}\ \beta^*_{\mathsf{w}}\ u'^{\Diamond}w^{\Diamond} = t^{\Diamond}$ (resp. $s^{\Diamond} = u^{\Diamond}w^{\Diamond}\ \beta^*_{\mathsf{w}}\ u^{\Diamond}w'^{\Diamond} = t^{\Diamond}$) by the $@_{\mathsf{l}}$-rule (resp. $@_{\mathsf{r}}$-rule) for β_{w}.

   If it is the $\mathit{sub}_{\mathsf{l}}$-rule (resp. $\mathit{sub}_{\mathsf{r}}$-rule) then $s = u\{w/x\}$ and $t = u'\{w/x\}$ (resp. $t = u\{w'/x\}$) for some λ_{vsub}-terms u, u' and w (resp. w') such that $u \to_{\mathsf{w}} u'$ (resp. $w \to_{\mathsf{w}} w'$). By induction hypothesis, $u^{\Diamond}\ \beta^*_{\mathsf{w}}\ u'^{\Diamond}$ (resp. $w^{\Diamond}\ \beta^*_{\mathsf{w}}\ w'^{\Diamond}$), so $s^{\Diamond} = (\lambda x\, u^{\Diamond})^{!}w^{\Diamond}\ \beta^*_{\mathsf{w}}\ (\lambda x\, u'^{\Diamond})^{!}w^{\Diamond} = t^{\Diamond}$ (resp. $s^{\Diamond} = (\lambda x\, u^{\Diamond})^{!}w^{\Diamond}\ \beta^*_{\mathsf{w}}\ (\lambda x\, u^{\Diamond})^{!}w'^{\Diamond} = t^{\Diamond}$) by the $\mathit{red}_{\mathsf{l}}$-rule (resp. $@_{\mathsf{r}}$-rule) for β_{w}.

3. By induction on the length of the derivation of $s \to_{\mathsf{sw}} t$. Let us consider the last rule of this derivation.

   If it is the w-rule, then $s^{\Diamond}\ \beta^*_{\mathsf{w}}\ t^{\Diamond}$ by 168.2, thus $s^{\Diamond}\ \beta^*_{\mathsf{sw}}\ t^{\Diamond}$ by the w-rule for β_{sw}.

   If it is the $@_{\mathsf{l}}$-rule then $s = uw$ and $t = u'w$ for some λ_{vsub}-terms u, u' and w such that $u \to_{\mathsf{sw}} u'$. By induction hypothesis, $u^{\Diamond}\ \beta^*_{\mathsf{sw}}\ u'^{\Diamond}$, so $s^{\Diamond} = u^{\Diamond}w^{\Diamond}\ \beta^*_{\mathsf{sw}}\ u'^{\Diamond}w^{\Diamond} = t^{\Diamond}$ by the $@_{\mathsf{l}}$-rule for β_{sw}.

   If it is the $\mathit{sub}_{\mathsf{l}}$-rule then $s = u\{w/x\}$ and $t = u'\{w/x\}$ for some λ_{vsub}-terms u, u' and w such that $u \to_{\mathsf{sw}} u'$. By induction hypothesis, $u^{\Diamond}\ \beta^*_{\mathsf{sw}}\ u'^{\Diamond}$, so $s^{\Diamond} = (\lambda x\, u^{\Diamond})^{!}w^{\Diamond}\ \beta^*_{\mathsf{sw}}\ (\lambda x\, u'^{\Diamond})^{!}w^{\Diamond} = t^{\Diamond}$ by the λ-rule, $!$-rule and $@_{\mathsf{l}}$-rule for β_{sw}.

If it is the λ-rule then $s = \lambda x\, u$ and $t = \lambda x\, u'$ for some λ_{vsub}-terms u and u' such that $u \to_{\mathsf{sw}} u'$. By induction hypothesis, $u^{\Diamond}\ \beta^{*}_{\mathsf{sw}}\ u'^{\Diamond}$, so $s^{\Diamond} = \lambda x\, u^{\Diamond}\ \beta^{*}_{\mathsf{sw}}\ \lambda x\, u'^{\Diamond} = t^{\Diamond}$ by the λ-rule for β_{sw}.

□

Remark 169. $s \to_{\mathsf{w}} t$ does not implies that $s^{\Diamond}\ \hat{\beta}^{*}_{\mathsf{v}\sigma}\ t^{\Diamond}$. For instance, take $u = (z_1)z_2 \in \lambda^{\mathsf{term}}_{\mathsf{vsub}}$ and $s = x_1\{y_1\{u/y_2\}/x_2\} \in \lambda^{\mathsf{term}}_{\mathsf{vsub}}$ where $x_1, x_2 y_1, y_2, z_1, z_2$ are pairwise distinct variables: then $u^{\Diamond} = (z_1^{!})z_2^{!}$, $s \to_{\mathsf{w}} x_1\{u/y_2\} = t$ and

$$s^{\Diamond} = (\lambda x_2\, x_1^{!})^{!}\big((\lambda y_2 y_1^{!})^{!}\big)u^{\Diamond}\ \sigma_3\ (\lambda y_2(\lambda x_2\, x_1^{!})^{!}y_1^{!})^{!}u^{\Diamond} = M$$

where M is $\hat{\beta}_{\mathsf{v}\sigma}$-normal but $t^{\Diamond} = (\lambda y_2 x_1^{!})^{!}u \neq M$. On the contrary, $M\ \beta_{\mathsf{w}}\ (\lambda y_2\, x_1^{!})^{!}u^{\Diamond} = t^{\Diamond}$ thanks to $\mathit{red}_{\mathsf{l}}$-rule.

By means of "forgetful functor" $(\,)^{\mathrm{F}}$ (see p. 73) and o-equivalence, $\to_{\lambda_{\mathsf{vsub}}}$-reduction can simulate the $\beta_{\mathsf{v}\sigma}$-reduction.

Remark 170. For every $E \in \Lambda_{\mathsf{CBV}}$ one has $\mathsf{fv}(E) = \mathsf{fv}(E^{\mathrm{F}})$ (the proof is by straightforward induction on $E \in \Lambda_{\mathsf{CBV}}$).

Lemma 171. *For every term M, value V and variable x, one has $(M[V/x])^{\mathrm{F}} = M^{\mathrm{F}}[V^{\mathrm{F}}/x]$.*

Proof. By induction on $M \in \Lambda_{\mathsf{t}}$.

If $M = x$ then $M^{\mathrm{F}} = x$, thus $(M[V/x])^{\mathrm{F}} = V^{\mathrm{F}} = M^{\mathrm{F}}[V^{\mathrm{F}}/x]$.

If $M = y$ for some variable $x \neq y$, $M^{\mathrm{F}} = y$, thus $(M[V/x])^{\mathrm{F}} = y = M^{\mathrm{F}}[V^{\mathrm{F}}/x]$.

If $M = \lambda y N$ for some term N, then we can suppose without loss of generality that $y \notin \mathsf{fv}(V) \cup \{x\}$; by induction hypothesis, $(N[V/x])^{\mathrm{F}} = N^{\mathrm{F}}[V^{\mathrm{F}}/x]$, thus $(M[V/x])^{\mathrm{F}} = (\lambda y N[V/x])^{\mathrm{F}} = \lambda y (N[V/x])^{\mathrm{F}} = \lambda y N^{\mathrm{F}}[V^{\mathrm{F}}/x] = M^{\mathrm{F}}[V^{\mathrm{F}}/x]$.

If $M = W^{!}$ for some value W, then $(W[V/x])^{\mathrm{F}} = W^{\mathrm{F}}[V^{\mathrm{F}}/x]$ by induction hypothesis, so $(M[V/x])^{\mathrm{F}} = ((W[V/x])^{!})^{\mathrm{F}} = (W[V/x])^{\mathrm{F}} = W^{\mathrm{F}}[V^{\mathrm{F}}/x] = M^{\mathrm{F}}[V^{\mathrm{F}}/x]$.

If $M = NL$ for some terms N and L, then $(N[V/x])^{\mathrm{F}} = N^{\mathrm{F}}[V^{\mathrm{F}}/x]$ and $(L[V/x])^{\mathrm{F}} = L^{\mathrm{F}}[V^{\mathrm{F}}/x]$, hence $(M[V/x])^{\mathrm{F}} = (N[V/x]L[V/x])^{\mathrm{F}} = (N[V/x])^{\mathrm{F}}(L[V/x])^{\mathrm{F}} = N^{\mathrm{F}}[V^{\mathrm{F}}/x]L^{\mathrm{F}}[V^{\mathrm{F}}/x] = M^{\mathrm{F}}[V^{\mathrm{F}}/x]$. □

Lemma 172. *Let $E, E' \in \Lambda_{\mathsf{CBV}}$.*

1. *If $E\ \beta_{\mathsf{v}}\ E'$ then $E^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} E'^{\mathrm{F}}$.*
2. *If $E\ \sigma_{\mathsf{v}}\ E'$ then $E^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} E'^{\mathrm{F}}$.*
3. *If $E\ \beta^{*}_{\mathsf{v}\sigma}\ E'$ then $E^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} E'^{\mathrm{F}}$.*
4. *For every $M \in \Lambda_{\mathsf{CBV}}$, if $M\ \beta^{*}_{\mathsf{v}\sigma}\ (\lambda x\, x^{!})^{!}$ then $M^{\mathrm{F}} \to^{*}_{\lambda_{\mathsf{vsub}}} \lambda x\, x$.*

Proof.

1. By induction on the derivation of $E \ \beta_{\mathsf{v}}\ E'$. Let us consider th last rule of this derivation.

   If it is the β-rule, then $E = (\lambda x N)^! V^!$ and $E' = N[V/x]$, hence $E^{\mathrm{F}} = (\lambda x N^{\mathrm{F}}) V^{\mathrm{F}} \to_{\lambda_{\mathsf{vsub}}} N^{\mathrm{F}}\{V^{\mathrm{F}}/x\} \to_{\lambda_{\mathsf{vsub}}} N^{\mathrm{F}}[V^{\mathrm{F}}/x] = (N[V/x])^{\mathrm{F}} = E'^{\mathrm{F}}$ by $\mathsf{d}\beta$- and sv-rule and lemma 171.

   If it is the $@_{\mathsf{l}}$(resp. $@_{\mathsf{r}}$)-rule, then $E = NL$ and $E' = N'L$ (resp. $E' = NL'$) for some terms N, L and N' (resp. L') such that $N \ \beta_{\mathsf{v}}\ N'$ (resp. $L \ \beta_{\mathsf{v}}\ L'$). By induction hypothesis $N^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} N'^{\mathrm{F}}$ (resp. $L^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} L'^{\mathrm{F}}$), so $E^{\mathrm{F}} = N^{\mathrm{F}} L^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} N'^{\mathrm{F}} L^{\mathrm{F}} = E'^{\mathrm{F}}$ (resp. $E^{\mathrm{F}} = N^{\mathrm{F}} L^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} N^{\mathrm{F}} L'^{\mathrm{F}} = E'^{\mathrm{F}}$).

   If it is the λ-rule, then $E = \lambda x N$ and $E' = \lambda x N'$ for some terms N and N' such that $N \ \beta_{\mathsf{v}}\ N'$. By induction hypothesis $N^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} N'^{\mathrm{F}}$, thus $E^{\mathrm{F}} = \lambda x N^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} \lambda x N'^{\mathrm{F}} = E'^{\mathrm{F}}$.

   If it is the !-rule then $E = V^!$ and $E' = V'^!$ for some values V and V' such that $V \ \beta_{\mathsf{v}}\ V'$. By induction hypothesis $V^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} V'^{\mathrm{F}}$, hence $E^{\mathrm{F}} = V^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} V'^{\mathrm{F}} = E'^{\mathrm{F}}$.

2. By induction on the derivation of $E \ \sigma_{\mathsf{v}}\ E'$. Let us consider th last rule of this derivation.

   If it is the σ_1-rule, then $E = (\lambda x M)^! N L$ and $E' = (\lambda x M L)^! N$ for some terms M, N and L with $x \notin \mathsf{fv}(L) = \mathsf{fv}(L^{\mathrm{F}})$, thus $E^{\mathrm{F}} = (\lambda x M^{\mathrm{F}}) N^{\mathrm{F}} L^{\mathrm{F}} \to_{\lambda_{\mathsf{vsub}}} M^{\mathrm{F}}\{N^{\mathrm{F}}/x\} L^{\mathrm{F}} \equiv_{\mathsf{o}} (M^{\mathrm{F}} L^{\mathrm{F}})\{N^{\mathrm{F}}/x\} \ {}_{\lambda_{\mathsf{vsub}}}\!\leftarrow (\lambda x M^{\mathrm{F}} L^{\mathrm{F}}) N^{\mathrm{F}} = E'^{\mathrm{F}}$. Therefore $E^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} E'^{\mathrm{F}}$.

   If it is the σ_3-rule, then $E = (V^!)\big((\lambda x L)^!\big) N$ and $E' = (\lambda x V^! L)^! N$ for some terms N and L and value V with $x \notin \mathsf{fv}(V) = \mathsf{fv}(V^{\mathrm{F}})$, hence $E^{\mathrm{F}} = (V^{\mathrm{F}})(\lambda x L^{\mathrm{F}}) N^{\mathrm{F}} \to_{\lambda_{\mathsf{vsub}}} V^{\mathrm{F}} L^{\mathrm{F}}\{N^{\mathrm{F}}/x\} \equiv_{\mathsf{o}} (V^{\mathrm{F}} L^{\mathrm{F}})\{N^{\mathrm{F}}/x\} \ {}_{\lambda_{\mathsf{vsub}}}\!\leftarrow (\lambda x V^{\mathrm{F}} L^{\mathrm{F}}) N^{\mathrm{F}} = E'^{\mathrm{F}}$. Therefore $E^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} E'^{\mathrm{F}}$.

   If it is the $@_{\mathsf{l}}$(resp. $@_{\mathsf{r}}$)-rule, then $E = NL$ and $M' = N'L$ (resp. $E' = NL'$) for some terms N, L and N' (resp. L') such that $N \ \sigma_{\mathsf{v}}\ N'$ (resp. $L \ \sigma_{\mathsf{v}}\ L'$). By induction hypothesis $N^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} N'^{\mathrm{F}}$ (resp. $L^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} L'^{\mathrm{F}}$), so $M^{\mathrm{F}} = N^{\mathrm{F}} L^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} N'^{\mathrm{F}} L^{\mathrm{F}} = M'^{\mathrm{F}}$ (resp. $E^{\mathrm{F}} = N^{\mathrm{F}} L^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} N^{\mathrm{F}} L'^{\mathrm{F}} = E'^{\mathrm{F}}$).

   If it is the λ-rule, then $E = \lambda x N$ and $E' = \lambda x N'$ for some terms N and N' such that $N \ \sigma_{\mathsf{v}}\ N'$. By induction hypothesis $N^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} N'^{\mathrm{F}}$, thus $E^{\mathrm{F}} = \lambda x N^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} \lambda x N'^{\mathrm{F}} = E'^{\mathrm{F}}$.

   If it is the !-rule then $E = V^!$ and $E = V'^!$ for some values V and V' such that $V \ \beta_{\mathsf{v}}\ V'$. By induction hypothesis $V^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} V'^{\mathrm{F}}$, hence $E^{\mathrm{F}} = V^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} V'^{\mathrm{F}} = E'^{\mathrm{F}}$.

3. By induction on the number $n \in \mathbb{N}$ of steps of the $\beta_{\mathsf{v}\sigma}$-reduction from E to E'. If $n = 0$ then $E = E'$ and so $E^{\mathrm{F}} = E'^{\mathrm{F}}$, thus $E^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} E'^{\mathrm{F}}$. If $n > 0$ then there exists $E'' \in \Lambda_{\mathsf{CBV}}$ such that $E \;\beta^{n}_{\mathsf{v}\sigma}\; E'' \;\beta_{\mathsf{v}\sigma}\; E'$; by induction hypothesis $E^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} E''^{\mathrm{F}}$; if $E'' \;\beta_{\mathsf{v}}\; E'$ then $E'^{\mathrm{F}} \to^{+}_{\lambda_{\mathsf{vsub}}} E'^{\mathrm{F}}$ by lemma 172.1, otherwise $E''^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} E'^{\mathrm{F}}$ by lemma 172.2; in any case $E^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} E'^{\mathrm{F}}$.

4. By lemma 172.3 $M^{\mathrm{F}} \leftrightarrow_{\lambda_{\mathsf{vsub}}/\equiv_{\mathsf{o}}} (\lambda x\, x^{!})^{!\mathrm{F}} = \lambda x\, x$. By lemmas 161.2 and 162, there exist $t, t' \in \lambda^{\mathsf{term}}_{\mathsf{vsub}}$ such that $M^{\mathrm{F}} \to^{*}_{\lambda_{\mathsf{vsub}}} t$, $t \equiv_{\mathsf{o}} t'$ and $\lambda x\, x \to^{*}_{\lambda_{\mathsf{vsub}}} t'$. As $\lambda x\, x$ is $\to_{\lambda_{\mathsf{vsub}}}$-normal, one has $t' = \lambda x\, x$, thus $t \equiv_{\mathsf{o}} \lambda x\, x$ that implies $t = \lambda x\, x$. Therefore $M^{\mathrm{F}} \to^{*}_{\lambda_{\mathsf{vsub}}} \lambda x\, x$. □

Lemma 172.4 says that if a term $M \in \Lambda_{\mathsf{CBV}}$ is solvable then $M^{\mathrm{F}} \in \lambda^{\mathsf{term}}_{\mathsf{vsub}}$ is solvable. This is not yet enough to characterize solvability in Λ_{CBV}, but we are trusting that a further and finer investigation on Λ_{CBV} can attain this result.

3.4 From terms to trees

It is more natural to study $\hat{\beta}_{\mathsf{vl}}$-, $\hat{\beta}_{\mathsf{vr}}$-, $\hat{\beta}_{\mathsf{v}}$- and $\hat{\beta}_{\mathsf{vt}}$-reductions and the decompositions seen in remark 113.2, theorem 131, corollaries 132.2 and 132.5 and proposition 134 by means of labeled full binary trees. This analysis, perhaps implicit in some publications on call-by-value λ-calculus, has never been made explicitly to our knowledge and reveals deep symmetries of this calculus which can be seen from a more "geometrical" point of view.

3.4.1 Syntax of applicative trees

Definition 173 (Quasi-leaf, applicative tree). *We denote by $\mathcal{T}_{@}$ the set of full binary trees whose internal nodes are labeled by @ and whose leaves are labeled by terms of the shape $V^{!}$ with $V \in \Lambda_{\mathsf{v}}$.*

A quasi-leaf *is an element of $\mathcal{T}_{@}$ whose root is a @-node having two leaves of T as children. For every term $T \in \mathcal{T}_{@}$, a* quasi-leaf of T *is a subtree of T which is a quasi-leaf.*

With every term M is associated $\mathsf{app}(M) \in \mathcal{T}_{@}$*, called* applicative tree of M*, defined by induction on M as follows:*

- $\mathsf{app}(V^{!})$ *consists of a leaf labeled by $V^{!}$;*
- $\mathsf{app}(MN)$ *consists of a node labeled by @ whose left (resp. right) child is* $\mathsf{app}(M)$ *(resp.* $\mathsf{app}(N)$*).*

With every $T \in \mathcal{T}_{@}$ is associated $\mathsf{term}(T) \in \Lambda_{\mathsf{v}}$*, called* term of T*, defined by induction on T as follows:*

- *if T consist simply of a leaf labeled by $V^!$ for some value V then* $\mathsf{term}(T) = V^!$*;*
- *if T consists of a node labeled by @ whose left (resp. right) child is* $\mathsf{term}(T_1)$ *(resp.* $\mathsf{term}(T_2)$*), then* $\mathsf{term}(T) = (\mathsf{term}(T_1))\mathsf{term}(T_2)$.

Remark 174.

1. Every leaf of every $T \in \mathcal{T}_@$ is labeled by a term of the shape $V^!$ for some value V.
2. For every terms M and N, $\mathsf{app}(MN)$ has at least one quasi-leaf: this is a well-known result on finite full binary trees having more than one node. As a consequence, for every term M, $\mathsf{app}(M)$ is either a leaf and in this case $M = V^!$ for some value V, or such that its root has two children and each sub-trees of $\mathsf{app}(M)$ contains a quasi-leaf.
3. It is immediate to check that the two functions app and term are inverses of each other: for every $T \in \mathcal{T}_@$ and term M, $\mathsf{app}(\mathsf{term}(T)) = T$ and $\mathsf{term}(\mathsf{app}(M)) = M$. So each element of $\mathcal{T}_@$ is the applicative tree of some term and each term is uniquely determined by its applicative tree.

Definition 175. *Let $T \in \mathcal{T}_@$. With every node n of T is associated a finite sequence* $\mathsf{pos}_T(n)$ *of elements of* $\{\mathsf{l}, \mathsf{r}\}$ *as follows (the definition is by induction on T):*

- *if the root of T is a leaf, then* $\mathsf{pos}_T(n) = \emptyset$*;*
- *if the root of T is not a leaf and if T_l (resp. n_r) is the left (resp. right) child of n, then for every node m in T_l (resp. T_r)* $\mathsf{pos}_T(m) = \mathsf{l} \cdot \mathsf{pos}_{T_\mathsf{l}}(m)$ *(resp.* $\mathsf{pos}_T(m) = \mathsf{r} \cdot \mathsf{pos}_{T_\mathsf{r}}(m)$*).*

For every subtree T' of T we set $\mathsf{pos}_T(T') = \mathsf{pos}_T(n)$ *where n is the root of T'.*

Remark 176. Given $T \in \mathcal{T}_@$, pos_T is a injection: if $\mathsf{pos}_T(n) = \mathsf{pos}_T(m)$ (resp. $\mathsf{pos}_T(T_1) = \mathsf{pos}_T(T_2)$) then $n = m$ (resp. $T_1 = T_2$), for any nodes m, n (resp. subtrees T_1, T_2) of T; this is a consequence of acyclicity of trees;

Given a term M, we can localize uniquely all its subterms thanks to $\mathsf{pos}^{-1}_{\mathsf{app}(M)}$.

We recall a well-known results on trees.

Proposition 177. *Let T be a tree.*

1. *Given two subtrees T_1 and T_2 of T, either T_1 is a subtree of T_2, or T_2 is a subtree of T_1, or T_1 and T_2 are disjoint.*

2. *Let* $<_T$ *be the binary relation on the subtrees of* T *defined by:* $R <_T S$ *(*R is on the left of S in T*) iff* R *and* S *are subtrees of* T *such that* $\mathsf{pos}_T(R) = (r_1, \ldots, r_{n_R})$, $\mathsf{pos}_T(S) = (s_1, \ldots, s_{n_S})$ *with* $n_R, n_S \in \mathbb{N}$, $r_i, s_j \in \{\mathsf{l}, \mathsf{r}\}$ *for any* $1 \leq i \leq n_R$ *and* $1 \leq j \leq n_S$*, and there exists* $m \leq n_R, n_S$ *such that* $r_m = \mathsf{l}$, $s_m = \mathsf{r}$ *and* $r_i = s_i$ *for every* $1 \leq i < m$*. Then* $<_T$ *is an order relation on the disjoint subtrees of* T*.*

Proof. T is a acyclic and connected graph, so we can conclude. □

Proposition 177.2 says that a natural order relation "from left to right" is definable on the disjoint subtrees of T.

Remark 178. Proposition 177 has as one of its consequences that for every $T \in \mathcal{T}_{@}$, if T_1 and T_2 are two distinct quasi-leaves of T then T_1 are T_2 are disjoint and either T_1 is on the left of T_2 or T_2 is on the left of T_1. This order relation is on the quasi-leaves of T is well-founded.

This notion of order is useful for the following definition.

Definition 179 **(**$\hat{\beta}_{\mathsf{v}}$**-redex on** $\mathcal{T}_{@}$**).** *A* $\hat{\beta}_{\mathsf{v}}$-redex *on* $\mathcal{T}_{@}$ *is an element of* $\mathcal{T}_{@}$ *which is a quasi-leaf whose left child is labeled by* $(\lambda x M)^!$ *for some* $M \in \Lambda_{\mathsf{t}}$*. Let* T *be a* $\hat{\beta}_{\mathsf{v}}$*-redex on* $\mathcal{T}_{@}$ *whose left (resp. right) child is labeled by* $(\lambda x M)^!$ *(resp.* $V^!$*) for some term* M *(resp. value* V*): the* contractum of T *is* $\mathsf{app}(M[V/x])$*.*

Let $T \in \mathcal{T}_{@}$*. A* $\hat{\beta}_{\mathsf{v}}$-redex in T *is an occurrence of subtree of* T *which is a* $\hat{\beta}_{\mathsf{v}}$*-redex on* $\mathcal{T}_{@}$*. If* T' *is both the leftmost (resp. rightmost) quasi-leave of* T *and a* $\hat{\beta}_{\mathsf{v}}$*-redex on* $\mathcal{T}_{@}$*,* T' *is the* $\hat{\beta}_{\mathsf{vl}}$-*(resp.* $\hat{\beta}_{\mathsf{vr}}$*-)*redex of T*.*

We say that T contains a $\hat{\beta}_{\mathsf{v}}$-*(resp.* $\hat{\beta}_{\mathsf{vl}}$*-;* $\hat{\beta}_{\mathsf{vr}}$*-)*redex *if there is a* $\hat{\beta}_{\mathsf{v}}$*-(resp.* $\hat{\beta}_{\mathsf{vl}}$*-;* $\hat{\beta}_{\mathsf{vr}}$*-)redex in* T*.*

Some elements of $\mathcal{T}_{@}$ might contains a $\hat{\beta}_{\mathsf{v}}$-redex without having neither the $\hat{\beta}_{\mathsf{vl}}$-redex nor the $\hat{\beta}_{\mathsf{vr}}$-redex, for example $\mathsf{app}((((x_1^!)x_2^!)(M)z^!)(y_1^1)y_2^!)$ where $M = (\lambda x N)^!$ for some term N.

Formally, the notions of $\hat{\beta}_{\mathsf{v}}$-, $\hat{\beta}_{\mathsf{vl}}$- and $\hat{\beta}_{\mathsf{vr}}$-redex on $\mathcal{T}_{@}$ are distinct from the notions of $\hat{\beta}_{\mathsf{v}}$-, $\hat{\beta}_{\mathsf{vl}}$- and $\hat{\beta}_{\mathsf{vr}}$-redex on terms, but actually they are strictly correlated. For example, if $M = ((\lambda x_1 x_1^!)^! y^!)((\lambda x_2 x_2^!)^!)z^!$, $\mathsf{app}(M)$ contains exactly two $\hat{\beta}_{\mathsf{v}}$-redexes, $\mathsf{app}((\lambda x_1 x_1^!)^! y^!)$ (the $\hat{\beta}_{\mathsf{vl}}$-redex of $\mathsf{app}(M)$) and $\mathsf{app}((\lambda x_2 x_2^!)^! z^!)$ (the $\hat{\beta}_{\mathsf{vr}}$-redex of $\mathsf{app}(M)$); but M also contains exactly two $\hat{\beta}_{\mathsf{v}}$-redexes, $(\lambda x_1 x_1^!)^! y^!$ (the $\hat{\beta}_{\mathsf{vl}}$-redex of M) and $(\lambda x_2 x_2^!)^! z^!$ (the $\hat{\beta}_{\mathsf{vr}}$-redex of M).

Lemma 180. *Let* M, N *be terms:* $\mathsf{app}(N)$ *is a* $\hat{\beta}_{\mathsf{v}}$*-redex in* $\mathsf{app}(M)$ *iff* N *is a* $\hat{\beta}_{\mathsf{v}}$*-redex in* M*.*

Proof. By induction on the term M.

If $M = V^!$ for some value V, then there is no $\hat{\beta}_{\mathsf{v}}$-redex in M and $\mathsf{app}(M)$ consists only of a leaf labeled by $V^!$, so there is no $\hat{\beta}_{\mathsf{vt}}$-redex in $\mathsf{app}(M)$.

If $M = M_1 M_2$ for some terms M_1, M_2, then there are three cases.

- N (resp. $\mathsf{app}(N)$) is a $\hat{\beta}_{\mathsf{v}}$-(resp. $\hat{\beta}_{\mathsf{vt}}$-)redex in M_1 (resp. $\mathsf{app}(M_1)$): by induction hypothesis, $\mathsf{app}(N)$ (resp. N) is a $\hat{\beta}_{\mathsf{v}}$-redex in $\mathsf{app}(M_1)$ (resp. M_1); as $\mathsf{app}(M)$ consists of a node whose left child is $\mathsf{app}(M_1)$ (resp. as $M = M_1M_2$), then $\mathsf{app}(N)$ (resp. N) is a $\hat{\beta}_{\mathsf{v}}$-redex of $\mathsf{app}(M)$ (resp. M).

- N (resp. $\mathsf{app}(N)$) is a $\hat{\beta}_{\mathsf{v}}$-(resp. $\hat{\beta}_{\mathsf{vt}}$-)redex in M_2 (resp. $\mathsf{app}(M_2)$): by induction hypothesis, $\mathsf{app}(N)$ (resp. N) is a $\hat{\beta}_{\mathsf{v}}$-redex in $\mathsf{app}(M_2)$ (resp. M_2); as $\mathsf{app}(M)$ consists of a node whose left child is $\mathsf{app}(M_2)$ (resp. as $M = M_1M_2$), then $\mathsf{app}(N)$ (resp. N) is a $\hat{\beta}_{\mathsf{v}}$-redex of $\mathsf{app}(M)$ (resp. M).

- $M = N = (\lambda x N')^! V^!$ for some term N' and value V: $\mathsf{app}(M) = \mathsf{app}(N)$ consists of a quasi-leaf whose left leaf is labeled by $(\lambda x N')^!$ and whose right leaf is labeled by $V^!$, hence $\mathsf{app}(M)$ is a $\hat{\beta}_{\mathsf{vt}}$-redex in $\mathsf{app}(M)$. □

3.4.2 Some reductions on applicative trees

Definition 181 ($\hat{\beta}_{\mathsf{v}}$-, $\hat{\beta}_{\mathsf{vl}}$-, $\hat{\beta}_{\mathsf{vr}}$- and $\hat{\beta}_{\mathsf{vt}}$-reduction on $\mathcal{T}_{@}$)**.** *We define a relation* $\hat{\beta}_{\mathsf{v}} \subseteq \mathcal{T}_{@} \times \mathcal{T}_{@}$*, called* $\hat{\beta}_{\mathsf{v}}$-reduction *or* weak β_{v}-reduction *on* $\mathcal{T}_{@}$*, as follows:* $T \ \hat{\beta}_{\mathsf{v}} \ T'$ *if* $T, T' \in \mathcal{T}_{@}$ *and* T' *is obrained from* T *by replacing a* $\hat{\beta}_{\mathsf{v}}$*-redex in* T *with its contractum.*

We define a relation $\hat{\beta}_{\mathsf{vl}}$ *(resp.* $\hat{\beta}_{\mathsf{vr}}$*)* $\subseteq \mathcal{T}_{@} \times \mathcal{T}_{@}$*, called* $\hat{\beta}_{\mathsf{vl}}$*(resp.* $\hat{\beta}_{\mathsf{vr}}$*)-*reduction *or* leftmost *(resp.* rightmost*)* weak β_{v}-reduction *on* $\mathcal{T}_{@}$*, as follows:* $T \ \hat{\beta}_{\mathsf{vl}} \, \hat{\beta}_{\mathsf{vr}} \ T'$ *if* $T, T' \in \mathcal{T}_{@}$ *and* T' *is obtained from* T *by replacing the* $\hat{\beta}_{\mathsf{vl}}$*(resp.* $\hat{\beta}_{\mathsf{vr}}$*)-redex in* T *(if any) with its contractum.*

We define a relation $\hat{\beta}_{\mathsf{vt}} \subseteq \mathcal{T}_{@} \times \mathcal{T}_{@}$*, called* $\hat{\beta}_{\mathsf{vt}}$-reduction *or* turbo weak β_{v}-reduction on $\mathcal{T}_{@}$*, as follows:* $T \ \hat{\beta}_{\mathsf{vt}} \ T'$ *if* $T, T' \in \mathcal{T}_{@}$ *and* T' *is obtained from* T *by replacing each* $\hat{\beta}_{\mathsf{v}}$*-redex in* T *with its contractum.*

Theorem 182. *Let* $T, T' \in \mathcal{T}_{@}$*.*

- *If* $T \ \hat{\beta}_{\mathsf{v}} \ T'$ *then* $\mathsf{term}(T) \ \hat{\beta}_{\mathsf{v}} \ \mathsf{term}(T')$*.*
- *If* $T \ \hat{\beta}_{\mathsf{vl}}$ *(resp.* $\hat{\beta}_{\mathsf{vr}}$*)* T' *then* $\mathsf{term}(T) \ \hat{\beta}_{\mathsf{vl}}$ *(resp.* $\hat{\beta}_{\mathsf{vr}}$*)* $\mathsf{term}(T')$*.*
- *If* $T \ \hat{\beta}_{\mathsf{vt}} \ T'$ *then* $\mathsf{term}(T) \ \hat{\beta}_{\mathsf{vt}} \ \mathsf{term}(T')$*.*

Proof. Immediate consequence of lemma 180. □

This theorem shows that it is very natural to consider terms as finite full binary trees when we study $\hat{\beta}_{\mathsf{v}}$-, $\hat{\beta}_{\mathsf{vl}}$-, $\hat{\beta}_{\mathsf{vr}}$- and $\hat{\beta}_{\mathsf{vt}}$-reductions: indeed this result gives a geometrical insight that should be kept in mind for the next chapter.

Chapter 4

Two symmetrical call-by-value Krivine abstract machines

Abstract machines play an important role in the implementation of programming languages. The reason abstract machines are so useful is because, on the one hand, they are sufficiently "abstract" to relate easily to other kinds of mathematical semantics, such as equational semantics; on the other hand, they are sufficiently "machine-like" to be easily implementable on real machines.

For the ordinary (call-by-name) λ-calculus, the most remarkable example of abstract machine is the Krivine's machine (KAM) [Kri85, Kri07, DR04]. For the call-by-value λ-calculus, the first abstract machine was the Landin's SECD [Lan65], another more recent example is the Leroy's ZINC [Ler90].

We introduce two versions of the KAM for our call-by-value λ-calculus Λ_{CBV}, that one without environments (closer to $\hat{\beta}_{\mathsf{v}}$-reduction) and that one with environments (closer to what happens in implementations of functional programming languages). Both versions have two subversions: the left-hand one and the right-hand one, which are perfectly symmetric. Our approach is more theoretical and "λ-calculus-like" (as in [Kri85, Kri07, DR04, dC09]) than the abstract machines defined in [Lan65, Ler90].

4.1 The versions without environments

Definition 183 (Stack, process). *A* stack *is a finite sequence of expressions.*

A process *is a pair (M, π), denoted by $M * \pi$, where M is a term and π is a stack.*

In other words, a process is a non-empty stack whose first component is a term.

Intuitively, a process can be seen as a program in execution.

Notation. Let $\pi = (E_1, \ldots, E_n)$ be a stack with $n \in \mathbb{N}$: if $n = 0$ we denote π by $\emptyset$; for every expression E, we denote by $E \cdot \pi$ (resp. $\pi \cdot E$) the stack $(E, E_1, \ldots, E_n)$ (resp. $(E_1, \ldots, E_n, E)$); moreover, we denote $E \cdot \emptyset$ and $\emptyset \cdot E$ by E.

Definition 184. *We define two call-by-value Krivine abstract machines without environments* K^{l} *(the* left CBV-KAM*) and* K^{r} *(the* right CBV-KAM*) by the following reduction rules:*

- *this reduction rule is common to* K^{l} *and* K^{r}

$$\mathsf{swap} \quad V^! * N \cdot \pi \quad \rightarrow \quad N * V \cdot \pi$$

- *these reduction rules are specific for* K^{l}

$$\begin{array}{rrcl} \mathsf{push_l} & (M)N * \pi & \rightarrow & M * N \cdot \pi \\ \mathsf{pop_l} & V^! * \lambda x M \cdot \pi & \rightarrow & M[V/x] * \pi \end{array}$$

- *these reduction rules are specific for* K^{r}

$$\begin{array}{rrcl} \mathsf{push_r} & (M)N * \pi & \rightarrow & N * M \cdot \pi \\ \mathsf{pop_r} & (\lambda x M)^! * V \cdot \pi & \rightarrow & M[V/x] * \pi \end{array}$$

Remark 185. The reduction rules for K^{l} (resp. K^{r}) are "strongly deterministic" (i.e. they form a partial map from the set of processes to the set of processes): for every process $M * \pi$ there exists at most one process $M' * \pi'$ such that $M * \pi \ \rightarrow \ M' * \pi'$ according to a reduction rule of K^{l} (resp. K^{r}).

Note the different role played by values and terms in K^{l} and K^{r}'s stacks respectively: in K^{l}(resp. K^{r})'s stack, values have to be seen as functions (resp. arguments) and terms have to be seen as arguments (resp. functions).

The fact that only the $\mathsf{pop_c}$ rule (with $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$) performs a substitution corresponds to the call-by-value constraint for reduction: the argument in a β_{v}-redex has to be a value. The $\mathsf{push_l}$ (resp. $\mathsf{push_r}$) and swap rules impose the call-by-value strategy reducing the "leftmost-(resp. rightmost-)outermost" β_{v}-redex.

Remark 186. Let M, M' be terms, E be an expression and π, π' be stacks: by definition, if $M * \pi \rightarrow_{\mathsf{x}} M' * \pi'$ with $\mathsf{x} \in \{\mathsf{pop_l}, \mathsf{pop_r}, \mathsf{push_l}, \mathsf{push_r}, \mathsf{swap}\}$, then $M * \pi \cdot E \rightarrow_{\mathsf{x}} M' * \pi' \cdot E$ for every expression E.

Definition 187. *With every process* $M * \pi$ *is associated a term* $\overline{M * \pi}^{\mathsf{l}}$ *and a term* $\overline{M * \pi}^{\mathsf{r}}$ *defined by induction on the length of* π *as follows:*

$$\begin{array}{rcl} \overline{M * \emptyset}^{\mathsf{l}} & := & M \\ \overline{M * V \cdot \pi}^{\mathsf{l}} & := & \overline{(V^!)M * \pi}^{\mathsf{l}} \\ \overline{M * N \cdot \pi}^{\mathsf{l}} & := & \overline{(M)N * \pi}^{\mathsf{l}} \end{array} \qquad \begin{array}{rcl} \overline{M * \emptyset}^{\mathsf{r}} & := & M \\ \overline{M * V \cdot \pi}^{\mathsf{r}} & := & \overline{(M)V^! * \pi}^{\mathsf{r}} \\ \overline{M * N \cdot \pi}^{\mathsf{r}} & := & \overline{(N)M * \pi}^{\mathsf{r}} \end{array}$$

Roughly speaking, the stack π in a process $M * \pi$ can be seen as the "applicative closure" of the term M, and the function $\overline{(\,)}^{\mathsf{l}}$ (resp. $\overline{(\,)}^{\mathsf{r}}$) allows to rebuild the term corresponding to a given process of K^{l} (resp. K^{r}), taking into account the swapped application in the stack i.e. the different role played by terms and values in the K^{l}(resp. K^{r})'s stack.

Lemma 188. *Let M, N be terms, V be a value and π be a stack.*

1. $\overline{M * \pi \cdot V}^{\mathsf{l}} = (V^{!})\overline{M * \pi}^{\mathsf{l}}$.
2. $\overline{M * \pi \cdot N}^{\mathsf{l}} = \big(\overline{M * \pi}^{\mathsf{l}}\big)N$.
3. $\overline{M * \pi \cdot V}^{\mathsf{r}} = \big(\overline{M * \pi}^{\mathsf{r}}\big)V^{!}$.
4. $\overline{M * \pi \cdot N}^{\mathsf{r}} = (N)\overline{M * \pi}^{\mathsf{r}}$.

Proof. All the proofs are by induction on the length of π.

1. If $\pi = \emptyset$, then $\overline{M * \pi \cdot V}^{\mathsf{l}} = \overline{M * V}^{\mathsf{l}} = \overline{(V^{!})M * \emptyset}^{\mathsf{l}} = (V^{!})M = (V^{!})\overline{M * \emptyset}^{\mathsf{l}} = (V^{!})\overline{M * \pi}^{\mathsf{l}}$.
   If $\pi = W \cdot \pi'$ where W is a value, then $\overline{M * \pi \cdot V}^{\mathsf{l}} = \overline{(W^{!})M * \pi' \cdot V}^{\mathsf{l}} = (V^{!})\overline{(W^{!})M * \pi'}^{\mathsf{l}} = (V^{!})\overline{M * \pi}^{\mathsf{l}}$ (the central identity holds by induction hypothesis).
   If $\pi = L \cdot \pi'$ where L is a term, then $\overline{M * \pi \cdot V}^{\mathsf{l}} = \overline{(M)L * \pi' \cdot V}^{\mathsf{l}} = (V^{!})\overline{(M)L * \pi'}^{\mathsf{l}} = (V^{!})\overline{M * \pi}^{\mathsf{l}}$ (the central identity holds by induction hypothesis).

2. If $\pi = \emptyset$, then $\overline{M * \pi \cdot N}^{\mathsf{l}} = \overline{M * N}^{\mathsf{l}} = \overline{(M)N * \emptyset}^{\mathsf{l}} = (M)N = (\overline{M * \emptyset}^{\mathsf{l}})N = (\overline{M * \pi}^{\mathsf{l}})N$.
   If $\pi = W \cdot \pi'$ where W is a value, then $\overline{M * \pi \cdot N}^{\mathsf{l}} = \overline{(W^{!})M * \pi' \cdot N}^{\mathsf{l}} = (\overline{(W^{!})M * \pi'}^{\mathsf{l}})N = (\overline{M * \pi}^{\mathsf{l}})N$ (the central identity holds by induction hypothesis).
   If $\pi = L \cdot \pi'$ where L is a term, then $\overline{M * \pi \cdot N}^{\mathsf{l}} = \overline{(M)L * \pi' \cdot N}^{\mathsf{l}} = (\overline{(M)L * \pi'}^{\mathsf{l}})N = (\overline{M * \pi}^{\mathsf{l}})N$ (the central identity holds by induction hypothesis).

3. If $\pi = \emptyset$, then $\overline{M * \pi \cdot V}^{\mathsf{r}} = \overline{M * V}^{\mathsf{r}} = \overline{(M)V^{!} * \emptyset}^{\mathsf{r}} = (M)V^{!} = (\overline{M * \emptyset}^{\mathsf{r}})V^{!} = (\overline{M * \pi}^{\mathsf{r}})V^{!}$.
   If $\pi = W \cdot \pi'$ where W is a value, then $\overline{M * \pi \cdot V}^{\mathsf{r}} = \overline{(M)W^{!} * \pi' \cdot V}^{\mathsf{r}} = (\overline{(M)W^{!} * \pi'}^{\mathsf{r}})V^{!} = (\overline{M * \pi}^{\mathsf{r}})V^{!}$ (the central identity holds by induction hypothesis).
   If $\pi = L \cdot \pi'$ where L is a term, then $\overline{M * \pi \cdot V}^{\mathsf{r}} = \overline{(L)M * \pi' \cdot V}^{\mathsf{r}} = (\overline{(L)M * \pi'}^{\mathsf{r}})V^{!} = (\overline{M * \pi}^{\mathsf{r}})V^{!}$ (the central identity holds by induction hypothesis).

4. If $\pi = \emptyset$, then $\overline{M * \pi \cdot N}^{\mathsf{r}} = \overline{M * N}^{\mathsf{r}} = \overline{(N)M * \emptyset}^{\mathsf{r}} = (N)M = (N)\overline{M * \emptyset}^{\mathsf{r}} = (N)\overline{M * \pi}^{\mathsf{r}}$.

If $\pi = W \cdot \pi'$ where W is a value, then $\overline{M * \pi \cdot N}^{\mathsf{r}} = \overline{(M)W^! * \pi' \cdot N}^{\mathsf{r}} = (N)\overline{(M)W^! * \pi'}^{\mathsf{r}} = (N)\overline{M * \pi}^{\mathsf{r}}$ (the central identity holds by induction hypothesis).
If $\pi = L \cdot \pi'$ where L is a term, then $\overline{M * \pi \cdot N}^{\mathsf{r}} = \overline{(L)M * \pi' \cdot N}^{\mathsf{r}} = (N)\overline{(L)M * \pi'}^{\mathsf{r}} = (N)\overline{M * \pi}^{\mathsf{r}}$ (the central identity holds by induction hypothesis).

□

Now we compare the two CBV-KAMs with β_{v}-reduction, more precisely we compare K^{l}(resp. K^{r})'s reduction rules with $\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)reduction.

Lemma 189. *Let M be a termand π be a stack.*

1. *If $M \; \hat{\beta}_{\mathsf{v}} \; M'$ (resp. $M \; \hat{\beta}_{\mathsf{vl}} \; M'$) then $\overline{M * \pi}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{v}} \; \overline{M' * \pi}^{\mathsf{l}}$ (resp. $\overline{M * \pi}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{vl}} \; \overline{M' * \pi}^{\mathsf{l}}$).*
2. *If $M \; \hat{\beta}_{\mathsf{v}} \; M'$ (resp. $M \; \hat{\beta}_{\mathsf{vr}} \; M'$) then $\overline{M * \pi}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{v}} \; \overline{M' * \pi}^{\mathsf{r}}$ (resp. $\overline{M * \pi}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{vr}} \; \overline{M' * \pi}^{\mathsf{r}}$).*

PROOF. Both proofs are by induction on the length of π.

1. If $\pi = \emptyset$, then $\overline{M * \pi}^{\mathsf{l}} = \overline{M * \emptyset}^{\mathsf{l}} = M \; \hat{\beta}_{\mathsf{v}} \; M' = \overline{M' * \emptyset}^{\mathsf{l}} = \overline{M' * \pi}^{\mathsf{l}}$.
   If $\pi = V\pi'$ where V is a value, then $\overline{M * \pi}^{\mathsf{l}} = \overline{(V^!)M * \pi'}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{v}} \; \overline{(V^!)M' * \pi'}^{\mathsf{l}} = \overline{M' * \pi}^{\mathsf{l}}$ (the central relation holds by induction hypothesis, since $(V^!)M \; \hat{\beta}_{\mathsf{v}} \; (V^!)M'$).
   If $\pi = N\pi'$ where N is a term, then $\overline{M * \pi}^{\mathsf{l}} = \overline{(M)N * \pi'}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{v}} \; \overline{(M')N * \pi'}^{\mathsf{l}} = \overline{M' * \pi}^{\mathsf{l}}$ (the central relation holds by induction hypothesis, since $(M)N \; \hat{\beta}_{\mathsf{v}} \; (M')N$).

   The proof that $M \; \hat{\beta}_{\mathsf{vl}} \; M'$ implies $\overline{M * \pi} \; \hat{\beta}_{\mathsf{vl}} \; \overline{M' * \pi}$ is analogous, it suffices to replace $\hat{\beta}_{\mathsf{v}}$ by $\hat{\beta}_{\mathsf{vl}}$.

2. If $\pi = \emptyset$, then $\overline{M * \pi}^{\mathsf{r}} = \overline{M * \emptyset}^{\mathsf{r}} = M \; \hat{\beta}_{\mathsf{v}} \; M' = \overline{M' * \emptyset}^{\mathsf{r}} = \overline{M' * \pi}^{\mathsf{r}}$.
   If $\pi = V\pi'$ where V is a value, then $\overline{M * \pi}^{\mathsf{r}} = \overline{(M)V^! * \pi'}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{v}} \; \overline{(M')V^! * \pi'}^{\mathsf{r}} = \overline{M' * \pi}^{\mathsf{r}}$ (the central relation holds by induction hypothesis, since $(M)V^! \; \hat{\beta}_{\mathsf{v}} \; (M')V^!$).
   If $\pi = N\pi'$ where N is a term, then $\overline{M * \pi}^{\mathsf{r}} = \overline{(N)M * \pi'}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{v}} \; \overline{(N)M' * \pi'}^{\mathsf{r}} = \overline{M' * \pi}^{\mathsf{r}}$ (the central relation holds by induction hypothesis, since $(N)M \; \hat{\beta}_{\mathsf{v}} \; (N)M'$).

   The proof that $M \; \hat{\beta}_{\mathsf{vr}} \; M'$ implies $\overline{M * \pi} \; \hat{\beta}_{\mathsf{vr}} \; \overline{M' * \pi}$ is analogous, it suffices to replace $\hat{\beta}_{\mathsf{v}}$ by $\hat{\beta}_{\mathsf{vr}}$.

□

Notice that $M \; \hat{\beta}_{\mathsf{v}} \; M'$ does not entail either $\overline{M * \pi}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{vl}} \; \overline{M' * \pi}^{\mathsf{l}}$ or $\overline{M * \pi}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{vr}} \; \overline{M' * \pi}^{\mathsf{r}}$; for example, take $M = ((z^!)z^!)((\lambda x x^!)^!)y^!$ (resp. $M = ((\lambda x x^!)^! y^!)(z^!)z^!$), $M' = ((z^!)z^!)y^!$ (resp. $M' = (y^!)(z^!)z^!$) and $\pi = \emptyset$: then $M \; \hat{\beta}_{\mathsf{v}} \; M'$ but $\overline{M * \emptyset}^{\mathsf{l}} = \overline{M * \emptyset}^{\mathsf{r}} = M$ which is a $\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)normal form.

Proposition 190. *Let M and M' be terms, let π and π' be stacks:*

- *if $M * \pi \rightarrow_{\mathsf{pop}_{\mathsf{l}}} M' * \pi'$ (resp. $M * \pi \rightarrow_{\mathsf{pop}_{\mathsf{r}}} M' * \pi'$) then $\overline{M * \pi}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{vl}} \; \overline{M' * \pi'}^{\mathsf{l}}$ (resp. $\overline{M * \pi}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{vr}} \; \overline{M' * \pi'}^{\mathsf{r}}$);*

- *if* $M * \pi \to_{\mathsf{x}} M' * \pi'$ *where* $\mathsf{x} \in \{\mathsf{push}_{\mathsf{l}}, \mathsf{swap}\}$ *(resp.* $\mathsf{x} \in \{\mathsf{push}_{\mathsf{r}}, \mathsf{swap}\}$*) then* $\overline{M * \pi}^{\mathsf{l}} = \overline{M' * \pi'}^{\mathsf{l}}$ *(resp.* $\overline{M * \pi}^{\mathsf{r}} = \overline{M' * \pi'}^{\mathsf{r}}$*).*

Proof. If $M * \pi \to_{\mathsf{pop}_{\mathsf{l}}} M' * \pi'$ (resp. $M * \pi \to_{\mathsf{pop}_{\mathsf{r}}} M' * \pi'$) then $M = V^!$, $\pi = \lambda x N \cdot \pi'$ (resp. $M = (\lambda x N)^!$, $\pi = V \cdot \pi'$) and $M' = N[V/x]$ for some term N and value V; as $(\lambda x N)^! V^! \; \hat{\beta}_{\mathsf{vl}}$ (resp. $\hat{\beta}_{\mathsf{vr}}$) $N[V/x]$, by lemma 189.1 (resp. 189.2) $\overline{M * \pi}^{\mathsf{l}} = \overline{(\lambda x N)^! V^! * \pi'}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{vl}} \; \overline{M' * \pi'}^{\mathsf{l}}$ (resp. $\overline{M * \pi}^{\mathsf{r}} = \overline{(\lambda x N)^! V^! * \pi'}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{vr}} \; \overline{M' * \pi'}^{\mathsf{r}}$).

If $M * \pi \to_{\mathsf{push}_{\mathsf{l}}} M' * \pi'$ (resp. $M * \pi \to_{\mathsf{push}_{\mathsf{r}}} M' * \pi'$) then $M = (L)N$ (resp. $M = (N)L$), $M' = L$ and $\pi' = N \cdot \pi$ for some terms N and L, so $\overline{M * \pi}^{\mathsf{c}} = \overline{L * N \cdot \pi}^{\mathsf{c}} = \overline{M' * \pi'}^{\mathsf{c}}$ with $\mathsf{c} = \mathsf{l}$ (resp. $\mathsf{c} = \mathsf{r}$).

If $M * \pi \to_{\mathsf{swap}} M' * \pi'$ then $M = V^!$, $M' = N$, $\pi = N \cdot \pi_0$ and $\pi' = V \cdot \pi_0$ for some term N and value V, so $\overline{M * \pi}^{\mathsf{l}} = \overline{(V^!)N * \pi_0}^{\mathsf{l}} = \overline{M' * \pi'}^{\mathsf{l}}$ (resp. $\overline{M * \pi}^{\mathsf{r}} = \overline{(N)V^! * \pi_0}^{\mathsf{r}} = \overline{M' * \pi'}^{\mathsf{r}}$). ☐

Proposition 190 states the soundness of the CBV-KAM K^{l}(resp. K^{r})'s reduction rules with respect to $\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)reduction (and $\hat{\beta}_{\mathsf{v}}$-reduction, by remark 130). Indeed the following is an immediate corollary of proposition 190:

Corollary 191. *Let* M, M' *be terms and* π, π' *be stacks. If* $M * \pi \to_{\mathsf{l}} M' * \pi'$ *(resp.* $M * \pi \to_{\mathsf{r}} M' * \pi'$*) then* $\overline{M * \pi}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{vl}}^{=} \; \overline{M' * \pi'}^{\mathsf{l}}$ *and* $\overline{M * \pi}^{\mathsf{l}} \; \hat{\beta}_{\mathsf{v}}^{=} \; \overline{M' * \pi'}^{\mathsf{l}}$ *(resp.* $\overline{M * \pi}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{vr}}^{=} \; \overline{M' * \pi'}^{\mathsf{r}}$ *and* $\overline{M * \pi}^{\mathsf{r}} \; \hat{\beta}_{\mathsf{v}}^{=} \; \overline{M' * \pi'}^{\mathsf{r}}$*). In particular, if* $M * \emptyset \to_{\mathsf{l}} M' * \pi'$ *(resp.* $M * \emptyset \to_{\mathsf{r}} M' * \pi'$*) then* $M \; \hat{\beta}_{\mathsf{vl}}^{=} \; \overline{M' * \pi'}^{\mathsf{l}}$ *and* $M \; \hat{\beta}_{\mathsf{v}}^{=} \; \overline{M' * \pi'}^{\mathsf{l}}$ *(resp.* $M \; \hat{\beta}_{\mathsf{vr}}^{=} \; \overline{M' * \pi'}^{\mathsf{r}}$ *and* $M \; \hat{\beta}_{\mathsf{v}}^{=} \; \overline{M' * \pi'}^{\mathsf{r}}$*).*

Remark 192. Let M, M', N be terms: $M \; \hat{\beta}_{\mathsf{vl}}$ (resp. $\hat{\beta}_{\mathsf{vr}}$) M' does not entail $N * M \to_{\mathsf{l}}^{*}$ (resp. $\to_{\mathsf{r}}^{*}$) $N * M'$. For example, take $M = (\lambda y y^!)^! z^!$, $M' = z^!$ and $N = x^!$: $M \; \hat{\beta}_{\mathsf{vl}} \; M'$ and $M \; \hat{\beta}_{\mathsf{vr}} \; M'$ but

$$N*M = x^!*(\lambda y y^!)^! z^! \to_{\mathsf{swap}} (\lambda y y^!)^! z^! * x \left\{ \begin{array}{c} \to_{\mathsf{push}_{\mathsf{l}}} (\lambda y y^!)^! * z^! \cdot x \to_{\mathsf{swap}} z^! * \lambda y y^! \cdot x \to_{\mathsf{pop}_{\mathsf{l}}} \\ \to_{\mathsf{push}_{\mathsf{r}}} z^! * (\lambda y y^!)^! \cdot x \to_{\mathsf{swap}} (\lambda y y^!)^! * z \cdot x \to_{\mathsf{pop}_{\mathsf{r}}} \end{array} \right\} z^! * x \not\to$$

and every process in the K^{l}-(resp. K^{r}-)reduction started with $N * M$ is different from $N * M'$.

As a consequence, $M \; \hat{\beta}_{\mathsf{v}} \; M'$ does not imply that there exist a term N and a stack π such that $M * \emptyset \to_{\mathsf{l}}^{*}$ (resp. $\to_{\mathsf{r}}^{*}$) $N * \pi$ and $M' = \overline{N * \pi}^{\mathsf{l}}$ (resp. $M' = \overline{N * \pi}^{\mathsf{r}}$). For instance, take $M = ((\lambda x_1 x_1^!)^! y^!)((\lambda x_2 x_2^!)^!) z^!$ and $M' = ((\lambda x_1 x_1^!)^! y^!) z^!$ (resp. $M' = (y^!)((\lambda x_2 x_2^!)^!) z^!$): $M \; \hat{\beta}_{\mathsf{v}} \; M'$ but

$$M * \emptyset \to_{\mathsf{push}_{\mathsf{l}}} (\lambda x_1 x_1^!)^! y^! * (\lambda x_2 x_2^!)^! z^! \to_{\mathsf{push}_{\mathsf{l}}} (\lambda x_1 x_1^!)^! * y^! \cdot (\lambda x_2 x_2^!)^! z^! \to_{\mathsf{swap}} y^! * \lambda x_1 x_1^! \cdot (\lambda x_2 x_2^!)^! z^!$$

$$\to_{\mathsf{pop}_{\mathsf{l}}} y^! * (\lambda x_2 x_2^!)^! z^! \to_{\mathsf{swap}} (\lambda x_2 x_2^!)^! z^! * y \to_{\mathsf{push}_{\mathsf{l}}} (\lambda x_2 x_2^!)^! * z^! \cdot y \to_{\mathsf{swap}} z^! * \lambda x_2 x_2^! \cdot y \to_{\mathsf{pop}_{\mathsf{l}}} z^! * y \not\to$$

$$(\text{resp.}\; M * \emptyset \to_{\mathsf{push}_{\mathsf{r}}} (\lambda x_2 x_2^!)^! z^! * (\lambda x_1 x_1^!)^! y^! \to_{\mathsf{push}_{\mathsf{r}}} z^! * (\lambda x_2 x_2^!)^! \cdot (\lambda x_1 x_1^!)^! y^! \to_{\mathsf{swap}} (\lambda x_2 x_2^!)^! * z \cdot (\lambda x_1 x_1^!)^! y^!$$

$$\to_{\mathsf{pop}_{\mathsf{r}}} z^! * (\lambda x_1 x_1^!)^! y^! \to_{\mathsf{swap}} (\lambda x_1 x_1^!)^! y^! * z \to_{\mathsf{push}_{\mathsf{r}}} y^! * (\lambda x_1 x_1^!)^! \cdot z \to_{\mathsf{swap}} (\lambda x_1 x_1^!)^! * y \cdot z \to_{\mathsf{pop}_{\mathsf{r}}} y^! * z \not\to\;)$$

and no process $N * \pi$ in the K^{l}-(resp. K^{r}-)reduction started with $M * \emptyset$ is such that $M' = \overline{N * \pi}^{\mathsf{l}}$ (resp. $M' = \overline{N * \pi}^{\mathsf{r}}$).

What makes both implications of remark 192 fail is that CBV-KAM K^{l}(resp. K^{r})'s reduction rules correspond to the call-by-value strategy reducing the "leftmost-(resp. rightmost-)outermost" β_{v}-redex, i.e. the $\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)reduction.

Proposition 193. *Let M and M' be terms. If $M \ \hat{\beta}_{\mathsf{vl}}$ (resp. $\hat{\beta}_{\mathsf{vr}}$) M' then there exist a term N and a stack π such that $M * \emptyset \rightarrow_{\mathsf{l}}^{+}$ (resp. $\rightarrow_{\mathsf{r}}^{+}$) $N * \pi$ and $M' = \overline{N * \pi}^{\mathsf{l}}$ (resp. $M' = \overline{N * \pi}^{\mathsf{r}}$).*

PROOF. By induction on $M \in \Lambda_{\mathsf{t}}$. Let us consider the last rule of the derivation of $M \ \hat{\beta}_{\mathsf{vl}}$ (resp. $\hat{\beta}_{\mathsf{vr}}$) M'.

If it is the β-rule, then $M = (\lambda x N)^! V^!$ and $M' = N[V/x]$ for some term N and value V, hence $M * \emptyset \rightarrow_{\mathsf{push_l}} (\lambda x N)^! * V^! \rightarrow_{\mathsf{swap}} V^! * \lambda x N \rightarrow_{\mathsf{pop_l}} N[V/x] * \emptyset$ (resp. $M * \emptyset \rightarrow_{\mathsf{push_r}} V^! * (\lambda x N)^! \rightarrow_{\mathsf{swap}} (\lambda x N)^! * V \rightarrow_{\mathsf{pop_r}} N[V/x] * \emptyset$), where $M' = \overline{N[V/x] * \emptyset}^{\mathsf{l}}$ (resp. $M' = \overline{N[V/x] * \emptyset}^{\mathsf{r}}$).

If it is the $@_{\mathsf{rv}}$-(resp. $@_{\mathsf{lv}}$-)rule, then $M = (V^!)L$ and $M' = (V^!)L'$ (resp. $M = (L)V^!$ and $M' = (L')V^!$) for some terms L, L' and value V with $L \ \hat{\beta}_{\mathsf{vl}}$ (resp. $\hat{\beta}_{\mathsf{vr}}$) L', thus there exist a term N and a stack π such that $L * \emptyset \rightarrow_{\mathsf{l}}^{+}$ (resp. $\rightarrow_{\mathsf{r}}^{+}$) $N * \pi$ and $L' = \overline{N * \pi}^{\mathsf{l}}$ (resp. $L' = \overline{N * \pi}^{\mathsf{r}}$) by induction hypothesis; hence $M * \emptyset \rightarrow_{\mathsf{push_l}}$(resp. $\rightarrow_{\mathsf{push_r}}$) $V^! * L \rightarrow_{\mathsf{swap}} L * V \rightarrow_{\mathsf{l}}^{+}$ (resp. $\rightarrow_{\mathsf{r}}^{+}$) $N * \pi \cdot V$ by remark 186, where $M' = (V^!)\overline{N * \pi}^{\mathsf{l}} = \overline{N * \pi \cdot V}^{\mathsf{l}}$ (resp. $M' = (\overline{N * \pi}^{\mathsf{r}})V^! = \overline{N * \pi \cdot V}^{\mathsf{r}}$) by lemma 188.1 (resp. 188.3).

If it is the $@_{\mathsf{l}}$-(resp. $@_{\mathsf{r}}$-)rule, then $M = (L_2)L_1$ and $M' = (L_2')L_1$ (resp. $M = (L_1)L_2$ and $M' = (L_1)L_2'$) for some terms L_1, L_2, L_2' with $L_2 \ \hat{\beta}_{\mathsf{vl}}$ (resp. $\hat{\beta}_{\mathsf{vr}}$) L_2', thus there exist an expression N and a stack π such that $L_2 * \emptyset \rightarrow_{\mathsf{l}}^{+}$ (resp. $\rightarrow_{\mathsf{r}}^{+}$) $N * \pi$ and $L_2' = \overline{N * \pi}^{\mathsf{l}}$ (resp. $L_2' = \overline{N * \pi}^{\mathsf{r}}$) by induction hypothesis; hence $M * \emptyset \rightarrow_{\mathsf{push_l}}$ (resp. $\rightarrow_{\mathsf{push_r}}$) $L_2 * L_1 \rightarrow_{\mathsf{l}}^{+}$ (resp. $\rightarrow_{\mathsf{r}}^{+}$) $N * \pi \cdot L_1$ by remark 186, where $M' = (\overline{N * \pi}^{\mathsf{l}})L_1 = \overline{N * \pi \cdot L_1}^{\mathsf{l}}$ (resp. $M' = (L_1)\overline{N * \pi}^{\mathsf{r}} = \overline{N * \pi \cdot L_1}^{\mathsf{r}}$) by lemma 188.2 (resp. 188.4). ☐

Intuitively, proposition 193 is a sort of converse to proposition 190, i.e. it states the "completeness" of the CBV-KAM K^{l}(resp. K^{r})'s reduction rules with respect to $\hat{\beta}_{\mathsf{vl}}$-(resp. $\hat{\beta}_{\mathsf{vr}}$-)reduction.

4.2 The versions with environments

We recall that the set of variables (resp. values; terms) of Λ_{CBV} is denoted by $\mathcal{V}$ (resp. Λ_{v}; Λ_{t}).

Definition 194 (Environment)**.** *For every $p \in \mathbb{N}$, we define a set $\mathcal{E}_p$, by induction on p, as follows:*

- $\mathcal{E}_0 = \mathcal{V} \rightharpoonup_{\text{fin}} \emptyset$ *(i.e. the set containing only the empty function* $\perp$*);*
- *if* $p > 0$ *then* $\mathcal{E}_p = \mathcal{V} \rightharpoonup_{\text{fin}} (\Lambda_{\mathsf{v}} \times \mathcal{E}_{p-1})$.

We set $\mathcal{E} = \bigcup_{p\in\mathbb{N}} \mathcal{E}_p$*, whose elements are called* environments*. For every* $e \in \mathcal{E}$*, we denote by* $\mathsf{d}(e)$ *the least* $p \in \mathbb{N}$ *such that* $e \in \mathcal{E}_p$*.*

Intuitively, an environment can be seen as a kind of heap memory used for dynamic memory allocation.

Remark 195. For every $p \in \mathbb{N}$, $\mathcal{E}_p \subseteq \mathcal{E}_{p+1}$. The proof is a straightforward induction on $p \in \mathbb{N}$. The empty function is in $\mathcal{E}_1$, so $\mathcal{E}_0 \subseteq \mathcal{E}_1$. Let $p > 0$ and $e \in \mathcal{E}_p$: if $\mathsf{dom}(e) = \{x_1, \ldots, x_n\}$ for some $n \in \mathbb{N}$, then for every $1 \leq i \leq n$ there exist a value V_i and $e_i \in \mathcal{E}_{p-1}$ such that $e(x_i) = (V_i, e_i)$; by induction hypothesis, $\mathcal{E}_{p-1} \subseteq \mathcal{E}_p$ and thus $e_i \in \mathcal{E}_p$, hence $e(x_i) \in \Lambda_{\mathsf{v}} \times \mathcal{E}_p$; therefore $e \in \mathcal{V} \rightharpoonup_{\text{fin}} (\Lambda_{\mathsf{v}} \times \mathcal{E}_p) = \mathcal{E}_{p+1}$, whence $\mathcal{E}_p \subseteq \mathcal{E}_{p+1}$.

Definition 196 (Closure)**.** *The set* $\mathcal{C}_{\mathsf{v}}$ *of* value *(resp.* term*)* closures *is defined by* $\mathcal{C}_{\mathsf{v}} = \Lambda_{\mathsf{v}} \times \mathcal{E}$ *(resp.* $\mathcal{C}_{\mathsf{t}} = \Lambda_{\mathsf{t}} \times \mathcal{E}$*). The set* $\mathcal{C}$ *of* closures *is defined by* $\mathcal{C} = \mathcal{C}_{\mathsf{v}} \cup \mathcal{C}_{\mathsf{t}}$*.*

Given $v = (V, e) \in \mathcal{C}_{\mathsf{v}}$*, we define* $\overline{v} = V[e] \in \Lambda_{\mathsf{v}}$ *by induction on* $\mathsf{d}(e) \in \mathbb{N}$*:*

- *if* $\mathsf{d}(e) = 0$ *then* $V[e] = V$*;*
- *if* $\mathsf{d}(e) > 0$ *then* $V[e] = V[\overline{e(x_1)}/x_1, \ldots, \overline{e(x_n)}/x_n]$ *where* $\mathsf{dom}(e) = \{x_1, \ldots, x_n\}$ *for some* $n \in \mathbb{N}$*.*

Given $t = (M, e) \in \mathcal{C}_{\mathsf{t}}$*, we define* $\overline{t} = M[e] \in \Lambda_{\mathsf{t}}$ *by induction on* $M \in \Lambda_{\mathsf{t}}$*:*

- *if* $M = V^!$ *for some value* V*, then* $M[e] = (V[e])^!$*;*
- *if* $M = NL$ *for some terms* N *and* L*, then* $M[e] = (N[e])L[e]$*.*

Remark 197.

1. With reference to notations used in definition 196, note that $\overline{v}$ is well-defined for $v = (V, e) \in \mathcal{C}_{\mathsf{v}}$: indeed, in the case $\mathsf{d}(e) > 0$, for every $1 \leq i \leq n$ there exists a value V_i and $e_i \in \mathcal{E}_{\mathsf{d}(e)-1}$ such that $e(x_i) = (V_i, e_i)$, hence $\mathsf{d}(e_i) \leq \mathsf{d}(e) - 1$ and so $\overline{e(x_i)}$ is defined by induction hypothesis. Furthermore if $c \in \mathcal{C}$ is a value (resp. term) closure, then $\overline{c}$ is a value (resp. term).
2. By definition, $\mathcal{E} = \mathcal{V} \rightharpoonup_{\text{fin}} \mathcal{C}_v$, i.e. environments are the partial functions with finite domains from the set of variables to the set of value closure.
3. $\mathcal{C} = \mathcal{C}_{\mathsf{v}} \uplus \mathcal{C}_{\mathsf{t}}$, since $\Lambda_{\mathsf{v}} \cap \Lambda_{\mathsf{t}} = \emptyset$.

Let E be an expression and e be an environment: each pair (x, v) (where x is a variable and $v = (V, e')$ is a value closure) in the graph of e can be seen as a sort of "recursive" explicit substitution in the expression $E[e]$, associating $V[e']$ with the free occurrences of x in E.

Definition 198 (Stack, state)**.** *A* stack *is a finite sequence of closures.*

A state *is a pair* (t, π)*, denoted by* $t * \pi$*, where* t *is a term closure and* π *is a stack. If* $s = t * (c_1, \ldots, c_n)$ *for some* $n \in \mathbb{N}$ *is a state, then* $\overline{s}$ *denotes the term* $(\overline{t})\overline{c_1} \cdots \overline{c_n}$*.*

In other words, a state is a non empty stack whose first component is a term closure.

Intuitively, a state is a program in execution, taking into account also the environment of this execution.

Definition 199 (Variable convention)**.** *For every value closure* $v = (V, e)$*, we define, by induction on* $\mathsf{d}(e)$*, what means that* the value closure v respects the variable convention*;* v *respects the variable convention if the following conditions are fulfilled:*

- *every bound variable in* E *is bound in* E *at most once;*
- *for every bound variable* x *in* E*,* $x \notin \mathsf{dom}(e)$*;*
- *for every* $v \in \mathsf{im}(e)$*,* v *respects the variable convention.*

For every term closure $t = (M, e)$*, we define, by induction on* M*, what means that* the term closure t respects the variable convention*:*

- *if* $M = V^!$ *for some value* V *then* t *respects the variable convention if* (V, e) *respects the variable convention;*
- *if* $M = NL$ *for some terms* N *and* L *then* t *respects the variable convention if* (N, e) *and* (L, e) *respect the variable convention.*

We say that state $t * (c_1, \ldots, c_n)$ *(where* $n \in \mathbb{N}$*,* t *is a term closure and* c_i *is a closure for any* $1 \leq i \leq n$*)* respects the variable convention *if the closures* $t, c_1, \ldots, c_n$ *respect the variable convention.*

For instance, $(\lambda y(\lambda x(x^!)x^!)^!, \perp)$ respects the variable convention, whereas $(\lambda x(\lambda x\, x^!)^!, \perp)$ does not.

Definition 200. *We define two call-by-value Krivine abstract machines with environments* $\mathsf{K}^{\mathsf{l}}_{\mathsf{env}}$ *(the* left CBV-KAM$_{\mathsf{env}}$*) and* $\mathsf{K}^{\mathsf{r}}_{\mathsf{env}}$ *(the* right CBV-KAM$_{\mathsf{env}}$*) by the following reduction rules:*

- *these reductions rule are common to* $\mathsf{K}^{\mathsf{l}}_{\mathsf{env}}$ *and* $\mathsf{K}^{\mathsf{r}}_{\mathsf{env}}$

$$\begin{array}{rrcll} \mathsf{swap} & (V^!, e) * (M, e')\cdot\pi & \to & (M, e') * (V, e)\cdot\pi & \textit{if either } V \notin \mathcal{V}, \textit{ or } V \in \mathcal{V} \textit{ and } V \notin \mathsf{dom} \\ \mathsf{sub} & (x^!, e) * \pi & \to & (V^!, e') * \pi & \textit{if } x \in \mathsf{dom}(e) \textit{ and } e(x) = (V, e') \end{array}$$

- *these reduction rules are specific for* $\mathsf{K}^{\mathsf{l}}_{\mathsf{env}}$

$$\begin{array}{rrcl} \mathsf{push_l} & (MN, e) * \pi & \to & (M, e) * (N, e)\cdot\pi \\ \mathsf{pop_l} & (V^!, e') * (\lambda x M, e)\cdot\pi & \to & (M, e \cup \{x \mapsto (V, e')\}) * \pi \end{array}$$

- *these reduction rules are specific for* $\mathsf{K}^{\mathsf{r}}_{\mathsf{env}}$

$$\begin{array}{rrcl} \mathsf{push_r} & (MN, e) * \pi & \to & (N, e) * (M, e) \cdot \pi \\ \mathsf{pop_r} & ((\lambda x M)^!, e) * (V, e') \cdot \pi & \to & (M, e \cup \{x \mapsto (V, e')\}) * \pi \end{array}$$

Remark 201. The reduction rules for $\mathsf{K}^{\mathsf{l}}_{\mathsf{env}}$ (resp. $\mathsf{K}^{\mathsf{r}}_{\mathsf{env}}$) are "strongly deterministic" (i.e. they form a partial map from the set of states to the set of states): for every state $t * \pi$ there exists at most one state $t' * \pi'$ such that $t * \pi \ \to \ t' * \pi'$ according to a reduction rule of $\mathsf{K}^{\mathsf{l}}_{\mathsf{env}}$ (resp. $\mathsf{K}^{\mathsf{r}}_{\mathsf{env}}$).

In a further work we aim to investigate the relationship between the two CBV-KAM versions with environments and the $\hat{\beta}_{\mathsf{vl}}$- and $\hat{\beta}_{\mathsf{vr}}$-reductions (or their modifications), in particular this study can hopefully allow to define a call-by-value version of the head linear reduction (see [DR04] for the call-by-name case). Finally, it should be interesting to look over the possibility of defining a resource-sensitive CBV-KAM version with environments (similar to the KAM version defined in [ER06a] for the call-by-name λ-calculus) in order to prove that also in the call-by-value case normalization and Taylor expansion commute, where normalizing a term in Λ_{CBV} stands for the computation performed by this resource-sensitive CBV-KAM, analogously to the result proved by Ehrhard and Regnier in [ER06a] and [ER08] for the ordinary λ-calculus.

Chapter 5

Translations

5.1 The typed Λ_{CBV} and boring translations in Linear Logic

A type system is a class of formulas in some language, the purpose of which is to express some properties of λ-terms. By introducing such formulas, as comments in the terms, we construct what we call typed terms, which correspond to programs in a high level programming language. The main connective in these formulas is "$\rightarrow$", the type $A \rightarrow B$ being that of the "functions" from A to B, that is to say from the set of terms of type A to the set of terms of type B.

By a variable declaration, we mean an ordered pair (x, A), where x is a variable of the λ-calculus, and A is a type. It will be denoted by $x : A$ instead of (x, A). A context Γ is a mapping from a finite set of variables to the set of all types. Thus it is a finite set $\{x_1 : A_1, \ldots, x_k : A_k\}$ of variable declarations, where $x_1, \ldots, x_k$ are distinct variables ; we will denote it by $x_1 : A_1, \ldots, x_k : A_k$ (without the braces). So, in such an expression, the order does not matter. We will say that x_i is declared of type A_i in the context Γ. The integer k may be 0; in that case, we have the empty context.

We will write $\Gamma, x : A$ in order to denote the context obtained by adding the declaration $x : A$ to the context Γ, provided that x is not already declared in Γ.

Given a λ-term t, a type A, and a context Γ, we define, by means of the following rules, the notion: t is of type A in the context Γ (we will also say : "t may be given type A in the context Γ") ; this will be denoted by $\Gamma \vdash_{\mathcal{L}} t : A$ (or $\Gamma \vdash t : A$ if there is no ambiguity) :

$$\frac{}{\Gamma, x : A \vdash x : A}\ ax \qquad\qquad \frac{\Gamma \vdash V : A}{\Gamma \vdash V^{!} : A}\ !$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x M : A \rightarrow B} \rightarrow_i \qquad\qquad \frac{\Gamma \vdash M : A \rightarrow B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \rightarrow_e$$

There are two ways to traduce the intuitionistic arrow $A \to B$ in Linear Logic with a “call-by-value” style (see [Gir87]).

$$\begin{array}{rcl}
X^\circ & := & !X \\
(A \to B)^\circ & := & !(A^\circ \multimap B^\circ) \\
(\Gamma \vdash V : A)^\circ & := & !\Gamma^\circ \vdash A^\circ \\
(\Gamma \vdash M : A)^\circ & := & !\Gamma^\circ \vdash A^\circ
\end{array}
\qquad
\begin{array}{rcl}
X^\bullet & := & X \\
(A \to B)^\bullet & := & (!A^\bullet \multimap !B^\bullet) \\
(\Gamma \vdash V : A)^\bullet & := & !\Gamma^\bullet \vdash V : A^\bullet \\
(\Gamma \vdash M : A)^\bullet & := & !\Gamma^\bullet \vdash M : !A^\bullet
\end{array}$$

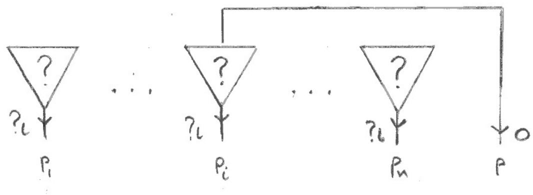
In some sense, the following proposition means that these two translations are equivalent.

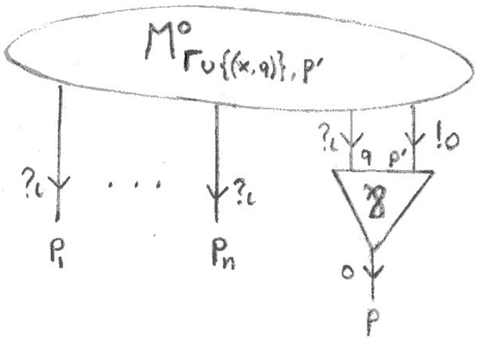
Proposition 202. *For every formula A of the implicative fragment, $A^\circ = !A^\bullet$.*

Proof. By induction on the formula A.

- If $A = X$, then $A^\circ = !X = !A^\bullet$.
- If $A = B \to C$, then $B^\circ = !B^\bullet$ and $C^\circ = !C^\bullet$ by induction hypothesis, so $A^\circ = !(B^\circ \multimap C^\circ) = !(!B^\bullet \multimap !C^\bullet) = !A^\bullet$.

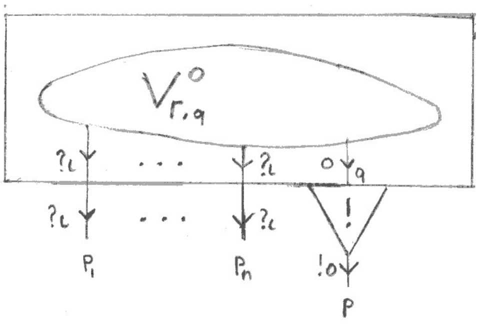
□

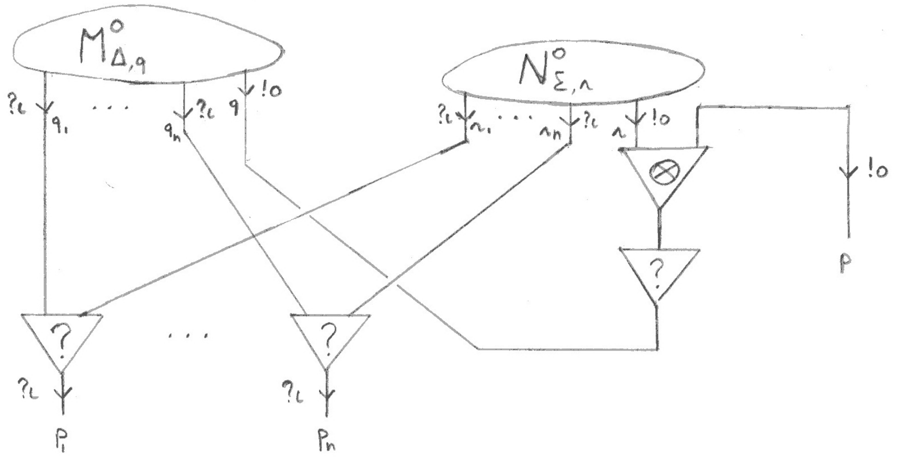




5.2 σ_{v}-equivalence

In the ordinary (call-by-name) λ-calculus the σ-equivalence (introduced by Regnier in [Reg92, Reg94]) identifies terms that differ only in their sequential





structure (e.g. $(\lambda x_1 \lambda x_2 u) v_1 v_2$ and $(\lambda x_2 \lambda x_1 u) v_2 v_1$): λ-terms contain pieces of information, which are unnecessary from the operational view-point.

The same phenomenon may be found in the call-by-value λ-calculus. So two questions naturally arise for Λ_{CBV}: find the σ_{V}-equivalence for Λ_{CBV}; find some parallel syntax which identifies σ_{V}-equivalent terms. In the ordinary λ-calculus, these two questions are answered by means of the Girard's translation of intuitionistic logic into Linear Logic proof-nets: $(A \to B) \leadsto (!A \multimap B)$. We give an analogous answer for Λ_{CBV} by means of the "boring" translation of intuitionistic logic into Linear Logic proof-nets: $(A \to B) \leadsto (!A \multimap !B)$.

Interestingly, this new σ_{V}-equivalence relation is not included in the β_{V}-equivalence, i.e. the σ_{V}-equivalence identifies distinct β_{V}-normal terms. We eventually show that two terms are equivalent iff they are translated as the same Linear Logic proof-net.

The σ_{V}-equivalence is generated by the following rules:

σ_1: $(\lambda x M)^! N L \simeq (\lambda x M L)^! N$ with $x \notin \mathsf{fv}(L)$;

σ_2: $(\lambda x (\lambda y L)^!)^! M N \simeq (\lambda y (\lambda x L)^!)^! N M$;

σ_3: $(M)((\lambda x L)^!) N \simeq (\lambda x M L)^! N$ with $x \notin \mathsf{fv}(M)$

None of this rules are included in the β_{v}-equivalence differently from the standard (call-by-name) λ-calculus, where the σ-equivalence is included in the β-equivalence. In some sense, the β_{v}-equivalence is incomplete, and the σ_{v}-equivalence is its completion.

Theorem 203. *For every expression E and F, $E \simeq F$ iff $E^\bullet = F^\bullet$.*

5.3 CPS

A more significant way than the forgetful functor to embedding the call-by-value λ-calculus Λ_{CBV} into the ordinary (call-by-name) λ-calculus Λ is the continuation-passing style (CPS) translation. We present two CPS translations, the left one $(\,)^{\mathsf{l}}$ (already used in [Plo75, Sel01]) and the right one $(\,)^{\mathsf{r}}$.

Definition 204. *Let E be an expression.*

We define, by induction on E, the left CPS translation of E*, denoted by $E^{\mathsf{l}} \in \Lambda$, as follows:*

- $x^{\mathsf{l}} = x$;
- $(\lambda x M)^{\mathsf{l}} = \lambda x M^{\mathsf{l}}$;
- $(V^!)^{\mathsf{l}} = \lambda k (k) V^{\mathsf{l}}$ *with* $k \notin \mathsf{fv}(V)$;
- $(MN)^{\mathsf{l}} = \lambda k (M^{\mathsf{l}}) \lambda m (N^{\mathsf{l}}) \lambda n (m) n k$ *with* $k, m, n \notin \mathsf{fv}(MN)$.

We define, by induction on E, the right CPS translation of E*, denoted by $E^{\mathsf{r}} \in \Lambda$, as follows:*

- $x^{\mathsf{r}} = x$;
- $(\lambda x M)^{\mathsf{r}} = \lambda x M^{\mathsf{r}}$;
- $(V^!)^{\mathsf{r}} = \lambda k (k) V^{\mathsf{r}}$ *with* $k \notin \mathsf{fv}(V)$;
- $(MN)^{\mathsf{r}} = \lambda k (N^{\mathsf{r}}) \lambda n (M^{\mathsf{r}}) \lambda m (m) n k$ *with* $k, m, n \notin \mathsf{fv}(MN)$.

Note that the only difference between left and right CPS translations is in the applicative case.

Remark 205. For every expression E, $\mathsf{fv}(E^{\mathsf{c}}) = \mathsf{fv}(E)$ with $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$ (the proof is a straightforward induction on $E \in \Lambda_{\mathsf{CBV}}$).

Lemma 206 (Substitution)**.** *For every expression E, value V and variable x, $(E[V/x])^{\mathsf{c}} = E^{\mathsf{c}}[V^{\mathsf{c}}/x]$ with $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$.*

Proof. By induction on $E \in \Lambda_{\mathsf{CBV}}$. Let $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$.

If $E = x$, then $E[V/x] = V$ and $E^{\mathsf{c}} = x$, so $(E[V/x])^{\mathsf{c}} = V^{\mathsf{c}} = E^{\mathsf{c}}[V^{\mathsf{c}}/x]$.

If $E = y$ for some variable $y \neq x$, then $E[V/x] = y$ and $E^{\mathsf{c}} = y$, so $(E[V/x])^{\mathsf{c}} = y = E^{\mathsf{c}}[V^{\mathsf{c}}/x]$.

If $E = \lambda y M$ for some term M, then we can suppose without loss of generality $y \notin \mathsf{fv}(V) \cup \{x\}$ (by α-equivalence), thus $E[V/x] = \lambda y M[V/x]$ and $E^{\mathsf{c}} = \lambda y M^{\mathsf{c}}$ with $(M[V/x])^{\mathsf{c}} = M^{\mathsf{c}}[V^{\mathsf{c}}/x]$ by induction hypothesis, so $(E[V/x])^{\mathsf{c}} = (\lambda y M[V/x])^{\mathsf{c}} = \lambda y (M[V/x])^{\mathsf{c}} = \lambda y M^{\mathsf{c}}[V^{\mathsf{c}}/x] = E^{\mathsf{c}}[V^{\mathsf{c}}/x]$ since $y \notin \mathsf{fv}(V^{\mathsf{c}}) \cup \{x\}$ by remark 205.

If $E = W^!$ for some value W, then $E[V/x] = (W[V/x])^!$ and $E^{\mathsf{c}} = \lambda k(k)W^{\mathsf{c}}$ with $k \notin \mathsf{fv}(W) \cup \{x\}$ (by α-equivalence), thus $k \notin \{x\} \cup \mathsf{fv}(W^{\mathsf{c}})$ by remark 205, moreover $(W[V/x])^{\mathsf{c}} = W^{\mathsf{c}}[V^{\mathsf{c}}/x]$ by induction hypothesis, so $(E[V/x])^{\mathsf{c}} = \lambda k(k)(W[V/x])^{\mathsf{c}} = \lambda k(k)W^{\mathsf{c}}[V^{\mathsf{c}}/x] = E^{\mathsf{c}}[W^{\mathsf{c}}/x]$.

If $E = MN$ for some terms M and N, then $E[V/x] = M[V/x]N[V/x]$ and

$$\begin{aligned} E^{\mathsf{l}} &= \lambda k(M^{\mathsf{l}})\lambda m(N^{\mathsf{l}})\lambda n(m)nk \\ E^{\mathsf{r}} &= \lambda k(N^{\mathsf{r}})\lambda n(M^{\mathsf{r}})\lambda m(m)nk \end{aligned}$$

with $k, m, n \notin \mathsf{fv}(M) \cup \mathsf{fv}(N) \cup \{x\} = \mathsf{fv}(M^{\mathsf{c}}) \cup \mathsf{fv}(N^{\mathsf{c}}) \cup \{x\}$ by remark 205 and α-equivalence, moreover $(M[V/x])^{\mathsf{c}} = M^{\mathsf{c}}[V^{\mathsf{c}}/x]$ and $(N[V/x])^{\mathsf{c}} = N^{\mathsf{c}}[V^{\mathsf{c}}/x]$ by induction hypothesis, hence

$$\begin{aligned} (E[V/x])^{\mathsf{l}} &= \lambda k(M[V/x])^{\mathsf{l}}\lambda m(N[V/x])^{\mathsf{l}}\lambda n(m)nk \\ &= \lambda k(M^{\mathsf{l}}[V^{\mathsf{l}}/x])\lambda m(N^{\mathsf{l}}[V^{\mathsf{l}}/x])\lambda n(m)nk = E^{\mathsf{l}}[V^{\mathsf{l}}/x] \\ (E[V/x])^{\mathsf{r}} &= \lambda k(N[V/x])^{\mathsf{r}}\lambda n(M[V/x])^{\mathsf{r}}\lambda m(m)nk \\ &= \lambda k(N^{\mathsf{r}}[V^{\mathsf{r}}/x])\lambda n(M^{\mathsf{r}}[V^{\mathsf{r}}/x])\lambda m(m)nk = E^{\mathsf{r}}[V^{\mathsf{r}}/x] \end{aligned}$$

□

Remark 207. For every values V_1 and V_2, if $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$ then $(V_1^! V_2^!)^{\mathsf{c}} \ \beta^+ \ \lambda k(V_1^{\mathsf{c}})V_2^{\mathsf{c}}k$ with $k \notin \mathsf{fv}(V_1^{\mathsf{c}}V_2^{\mathsf{c}})$. Indeed, let $k, m, n \notin \mathsf{fv}(V_1^{\mathsf{c}}) \cup \mathsf{fv}(V_2^{\mathsf{c}}) = \mathsf{fv}(V_1^{\mathsf{c}}V_2^{\mathsf{c}})$ with $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$:

$$\begin{aligned} (V_1^! V_2^!)^{\mathsf{l}} &= \lambda k(\lambda k_1(k_1)V_1^{\mathsf{l}})\lambda m(\lambda k_2(k_2)V_2^{\mathsf{l}})\lambda n(m)nk \\ &\quad \beta \ \lambda k(\lambda k_1(k_1)V_1^{\mathsf{l}})\lambda m(\lambda n(m)nk)V_2^{\mathsf{l}} \\ &\quad \beta \ \lambda k(\lambda m(\lambda n(m)nk)V_2^{\mathsf{l}})V_1^{\mathsf{l}} \ \beta \ \lambda k(\lambda n(V_1^{\mathsf{l}})nk)V_2^{\mathsf{l}} \ \beta \ \lambda k V_1^{\mathsf{l}}V_2^{\mathsf{l}}k\,. \\ (V_1^! V_2^!)^{\mathsf{r}} &= \lambda k(\lambda k_2(k_2)V_2^{\mathsf{r}})\lambda n(\lambda k_1(k_1)V_1^{\mathsf{r}})\lambda m(m)nk \\ &\quad \beta \ \lambda k(\lambda k_2(k_2)V_2^{\mathsf{r}})\lambda n(\lambda m(m)nk)V_1^{\mathsf{r}} \\ &\quad \beta \ \lambda k(\lambda n(\lambda m(m)nk)V_1^{\mathsf{r}})V_2^{\mathsf{r}} \ \beta \ \lambda k(\lambda n(V_1^{\mathsf{r}})nk)V_2^{\mathsf{r}} \ \beta \ \lambda k V_1^{\mathsf{r}}V_2^{\mathsf{r}}k\,. \end{aligned}$$

The following proposition claims that one step of β_{v}-(and so $\hat{\beta}_{\mathsf{v}}$-)reduction is simulated by at least one step of $\beta\eta$-reduction in ordinary λ-calculus, modulo left or right CPS translation.

Proposition 208. *Let* $E, E' \in \Lambda_{\mathsf{CBV}}$*: if* $E \ \beta_{\mathsf{v}}\ E'$ *then* $E^{\mathsf{c}}\ \beta\eta^{+}\ E'^{\mathsf{c}}$ *with* $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$.

Proof. By induction on $E \in \Lambda_{\mathsf{CBV}}$. Let us consider the last rule of the derivation of $E\ \beta_{\mathsf{v}}\ E'$.

If it is the β-rule, then $E = (\lambda x M)^{!}V^{!}$ and $E' = M[V/x]$ for some term M and value V, hence $E^{\mathsf{c}}\ \beta^{+}\ \lambda k(\lambda x M^{\mathsf{c}})V^{\mathsf{c}}k$ by remark 207 and $E'^{\mathsf{c}} = (M[V/x])^{\mathsf{c}} = M^{\mathsf{c}}[V^{\mathsf{c}}/x]$ by lemma 206; thus

$$E^{\mathsf{c}}\ \beta^{+}\ \lambda k(\lambda x M^{\mathsf{c}})V^{\mathsf{c}}k\ \beta\ \lambda k(M^{\mathsf{c}}[V^{\mathsf{c}}/x])k = \lambda k(E'^{\mathsf{c}})k\ \eta\ E'^{\mathsf{c}}$$

since $k \notin \mathsf{fv}(M^{\mathsf{c}}) \cup \mathsf{fv}(V^{\mathsf{c}}) \cup \{x\}$.

If it is the $@_{\mathsf{l}}$-rule, then $E = MN$ and $E' = M'N$ where M, M', N are terms with $M\ \beta_{\mathsf{v}}\ M'$, hence $M^{\mathsf{c}}\ \beta\eta^{+}\ M'^{\mathsf{c}}$ with $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$ by induction hypothesis, so

$$\begin{aligned} E^{\mathsf{l}} &= \lambda k(M^{\mathsf{l}})\lambda m(N^{\mathsf{l}})\lambda n(m)nk \quad \beta\eta^{+}\ \lambda k(M'^{\mathsf{l}})\lambda m(N^{\mathsf{l}})\lambda n(m)nk = E'^{\mathsf{l}} \\ E^{\mathsf{r}} &= \lambda k(N^{\mathsf{r}})\lambda n(M^{\mathsf{r}})\lambda m(m)nk \quad \beta\eta^{+}\ \lambda k(N^{\mathsf{r}})\lambda n(M'^{\mathsf{r}})\lambda m(m)nk = E'^{\mathsf{r}} \end{aligned}$$

since $\beta\eta$-reduction passes to context.

If it is the $@_{\mathsf{r}}$-rule, then $E = MN$ and $E' = MN'$ where M, N, N' are terms with $N\ \beta_{\mathsf{v}}\ N'$, hence $N^{\mathsf{c}}\ \beta\eta^{+}\ N'^{\mathsf{c}}$ with $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$ by induction hypothesis, so

$$\begin{aligned} E^{\mathsf{l}} &= \lambda k(M^{\mathsf{l}})\lambda m(N^{\mathsf{l}})\lambda n(m)nk \quad \beta\eta^{+}\ \lambda k(M^{\mathsf{l}})\lambda m(N'^{\mathsf{l}})\lambda n(m)nk = E'^{\mathsf{l}} \\ E^{\mathsf{r}} &= \lambda k(N^{\mathsf{r}})\lambda n(M^{\mathsf{r}})\lambda m(m)nk \quad \beta\eta^{+}\ \lambda k(N'^{\mathsf{r}})\lambda n(M^{\mathsf{r}})\lambda m(m)nk = E'^{\mathsf{r}} \end{aligned}$$

since $\beta\eta$-reduction passes to context.

If it is the λ-rule, then $E = \lambda x M$ and $E' = \lambda x M'$ where M is a term with $M\ \beta_{\mathsf{v}}\ M'$, hence for $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$, $M^{\mathsf{c}}\ \beta\eta^{+}\ M'^{\mathsf{c}}$ by induction hypothesis, so $E^{\mathsf{c}} = \lambda x M^{\mathsf{c}}\ \beta\eta^{+}\ \lambda x M'^{\mathsf{c}} = E'^{\mathsf{c}}$ since $\beta\eta$-reduction passes to context.

If it is the !-rule, then $E = V^{!}$ and $E' = V'^{!}$ where V, V' are values with $V\ \beta_{\mathsf{v}}\ V'$, hence for $\mathsf{c} \in \{\mathsf{l}, \mathsf{r}\}$, $V^{\mathsf{c}}\ \beta\eta^{+}\ V'^{\mathsf{c}}$ by induction hypothesis, so $E^{\mathsf{c}} = \lambda k(k)V^{\mathsf{c}}\ \beta\eta^{+}\ \lambda k(k)V'^{\mathsf{c}} = E'^{\mathsf{c}}$ since $\beta\eta$-reduction passes to context. ☐

We set about to show the following result: the call-by-value Krivine's machine K^{l} (resp. K^{r}) without environments is simulated by the call-by-name Krivine's machine modulo the CPS translation $(\,)^{\mathsf{l}}$ (resp. $(\,)^{\mathsf{r}}$).

We remind that Λ denotes the set of ordinary (i.e. call-by-name) terms and that a stack is a finite sequence of expressions (in Λ_{CBV}).

Notation. Let $s, t \in \Lambda$ and m, n be variables with $m, n \notin \mathsf{fv}(s) \cup \mathsf{fv}(t)$. We set:

- $\mathsf{arg}_\mathsf{l}(s,t) = \lambda m(s)\lambda n(m)nt$;
- $\mathsf{fun}_\mathsf{l}(s,t) = \lambda n(s)nt$;
- $\mathsf{arg}_\mathsf{r}(s,t) = \lambda m(m)st$;
- $\mathsf{fun}_\mathsf{r}(s,t) = \lambda n(s)\lambda m(m)nt$.

Definition 209. *Let* $t \in \Lambda$. *With every stack* π *there is associated* $\pi^\mathsf{l}_t \in \Lambda$ *and* $\pi^\mathsf{r}_t \in \Lambda$ *defined as follows by induction on the length of* π*:*

- $\emptyset^\mathsf{l}_t = t$*;*
- $(M\cdot\pi)^\mathsf{l}_t = \mathsf{arg}_\mathsf{l}(M^\mathsf{l}, \pi^\mathsf{l}_t)$*;*
- $(V\cdot\pi)^\mathsf{l}_t = \mathsf{fun}_\mathsf{l}(V^\mathsf{l}, \pi^\mathsf{l}_t)$*;*
- $\emptyset^\mathsf{r}_t = t$*;*
- $(M\cdot\pi)^\mathsf{r}_t = \mathsf{fun}_\mathsf{r}(M^\mathsf{l}, \pi^\mathsf{r}_t)$*;*
- $(V\cdot\pi)^\mathsf{r}_t = \mathsf{arg}_\mathsf{r}(V^\mathsf{l}, \pi^\mathsf{r}_t)$*;*

We remind the definition of the call-by-name Krivine abstract machine without environments.

Definition 210. *An* ordinary stack *is a finite sequence of ordinary terms* Λ.

An ordinary process *is a pair* (t,π)*, denoted by* $t*\pi$*, where* t *is an ordinary term (in* Λ*) and* π *is an ordinary stack.*

The call-by-name Krivine abstract machines without environments K *is defined by the following reduction rules:*

$$\begin{array}{rl} \mathsf{push} & (s)t * \pi \to_{\text{name}} s * t\cdot\pi \\ \mathsf{pop} & \lambda x\, s * t\cdot\pi \to_{\text{name}} s[t/x] * \pi \end{array}$$

Proposition 211. *Let* $t \in \Lambda$*, let* M, M' *be terms (in* Λ_CBV*) and* π, π' *be stacks (in* Λ_CBV*). If* $M * \pi \to_\mathsf{x} M' * \pi'$ *with* $\mathsf{x} \in \{\mathsf{pop}_\mathsf{l}, \mathsf{push}_\mathsf{l}, \mathsf{swap}\}$ *(resp.* $\mathsf{x} \in \{\mathsf{pop}_\mathsf{r}, \mathsf{push}_\mathsf{r}, \mathsf{swap}\}$*), then* $M^\mathsf{l} * \pi^\mathsf{l}_t \to^+_{\text{name}} M'^\mathsf{l} * \pi'^\mathsf{l}_t$ *(resp.* $M^\mathsf{r} * \pi^\mathsf{r}_t \to^+_{\text{name}} M'^\mathsf{r} * \pi'^\mathsf{r}_t$*).*

Proof. If $M * \pi \to_\mathsf{swap} M' * \pi'$ then $M = V^!$, $\pi = N\cdot\rho$, $M' = N$ and $\pi' = V\cdot\rho$ for some value V, term N and stack ρ. Hence $M^\mathsf{l} = \lambda k(k)V^\mathsf{l}$ and $\pi^\mathsf{l}_t = \mathsf{arg}_\mathsf{l}(N^\mathsf{l}, \rho^\mathsf{l}_t)$ (resp. $M^\mathsf{r} = \lambda k(k)V^\mathsf{r}$ and $\pi^\mathsf{r}_t = \mathsf{fun}_\mathsf{r}(N^\mathsf{r}, \rho^\mathsf{r}_t)$), so

$$\begin{array}{rl} M^\mathsf{l} * \pi^\mathsf{l}_t = & \lambda k(k)V^\mathsf{l} * \mathsf{arg}_\mathsf{l}(N^\mathsf{l}, \rho^\mathsf{l}_t) \to_{\text{name}} (\mathsf{arg}_\mathsf{l}(N^\mathsf{l}, \rho^\mathsf{l}_t))V^\mathsf{l} * \emptyset \\ & \to_{\text{name}} \lambda m(N^\mathsf{l})\lambda n(m)n\rho^\mathsf{l}_t * V^\mathsf{l} \to_{\text{name}} (N^\mathsf{l})\lambda n(V^\mathsf{l})n\rho^\mathsf{l}_t * \emptyset \\ & \to_{\text{name}} N^\mathsf{l} * \mathsf{fun}_\mathsf{l}(V^\mathsf{l}, \rho^\mathsf{l}_t) = M'^\mathsf{l} * \pi'^\mathsf{l}_t \\ \text{(resp. } M^\mathsf{r} * \pi^\mathsf{r}_t = & \lambda k(k)V^\mathsf{r} * \mathsf{fun}_\mathsf{r}(N^\mathsf{r}, \rho^\mathsf{r}_t) \to_{\text{name}} (\mathsf{fun}_\mathsf{r}(N^\mathsf{r}, \rho^\mathsf{r}_t))V^\mathsf{r} * \emptyset \\ & \to_{\text{name}} \lambda n(N^\mathsf{r})\lambda m(m)n\rho^\mathsf{r}_t * V^\mathsf{r} \to_{\text{name}} (N^\mathsf{r})\lambda m(m)V^\mathsf{r}\rho^\mathsf{r}_t * \emptyset \\ & \to_{\text{name}} N^\mathsf{r} * \mathsf{arg}_\mathsf{r}(V^\mathsf{r}, \rho^\mathsf{r}_t) = M'^\mathsf{r} * \pi'^\mathsf{r}_t \text{)} \end{array}$$

If $M * \pi \to_{\mathsf{push}_\mathsf{l}} M' * \pi'$ (resp. $M * \pi \to_{\mathsf{push}_\mathsf{r}} M' * \pi'$) then $M = NL$, $M' = N$ (resp. $M' = L$) and $\pi' = N\cdot\pi$ for some terms N, L. Hence

$M^{\mathsf{l}} = \lambda k(N^{\mathsf{l}})\mathsf{arg}_{\mathsf{l}}(L^{\mathsf{l}}, k)$ and $\pi'^{\mathsf{l}}_t = \mathsf{arg}_{\mathsf{l}}(L^{\mathsf{l}}, \pi^{\mathsf{l}}_t)$ (resp. $M^{\mathsf{r}} = \lambda k(L^{\mathsf{r}})\mathsf{fun}_{\mathsf{r}}(N^{\mathsf{r}}, k)$ and $\pi'^{\mathsf{r}}_t = \mathsf{fun}_{\mathsf{l}}(N^{\mathsf{r}}, \pi^{\mathsf{r}}_t)$), so

$$\begin{aligned}
M^{\mathsf{l}} * \pi^{\mathsf{l}}_t &= \lambda k(N^{\mathsf{l}})\mathsf{arg}_{\mathsf{l}}(L^{\mathsf{l}}, k) * \pi^{\mathsf{l}}_t \to_{\text{name}} (N^{\mathsf{l}})\mathsf{arg}_{\mathsf{l}}(L^{\mathsf{l}}, \pi^{\mathsf{l}}_t) * \emptyset \\
&\to_{\text{name}} N^{\mathsf{l}} * \mathsf{arg}_{\mathsf{l}}(L^{\mathsf{l}}, \pi^{\mathsf{l}}_t) = M'^{\mathsf{l}} * \pi'^{\mathsf{l}}_t \\
\text{(resp. } M^{\mathsf{r}} * \pi^{\mathsf{r}}_t &= \lambda k(L^{\mathsf{r}})\mathsf{fun}_{\mathsf{r}}(N^{\mathsf{r}}, k) * \pi^{\mathsf{r}}_t \to_{\text{name}} (L^{\mathsf{r}})\mathsf{fun}_{\mathsf{r}}(N^{\mathsf{r}}, \pi^{\mathsf{r}}_t) * \emptyset \\
&\to_{\text{name}} L^{\mathsf{r}} * \mathsf{fun}_{\mathsf{r}}(N^{\mathsf{r}}, \pi^{\mathsf{r}}_t) = M'^{\mathsf{r}} * \pi'^{\mathsf{r}}_t \text{)}
\end{aligned}$$

If $M * \pi \to_{\mathsf{pop}_{\mathsf{l}}} M' * \pi'$ (resp. $M * \pi \to_{\mathsf{pop}_{\mathsf{r}}} M' * \pi'$) then $M = V^!$ (resp. $M = (\lambda x N)^!$), $\pi = \lambda x N \cdot \pi'$ (resp. $\pi = V \cdot \pi'$) and $M' = N[V/x]$ for some terms N and value V. Hence $M^{\mathsf{l}} = \lambda k(k)V^{\mathsf{l}}$ and $\pi^{\mathsf{l}}_t = \mathsf{fun}_{\mathsf{l}}(\lambda x N^{\mathsf{l}}, \pi'^{\mathsf{l}}_t)$ (resp. $M^{\mathsf{r}} = \lambda k(k)\lambda x N^{\mathsf{r}}$ and $\pi^{\mathsf{r}}_t = \mathsf{arg}_{\mathsf{r}}(V^{\mathsf{r}}, \pi'^{\mathsf{r}}_t)$), so

$$\begin{aligned}
M^{\mathsf{l}} * \pi^{\mathsf{l}}_t &= \lambda k(k)V^{\mathsf{l}} * \mathsf{fun}_{\mathsf{l}}(\lambda x N^{\mathsf{l}}, \pi'^{\mathsf{l}}_t) \to_{\text{name}} (\mathsf{fun}_{\mathsf{l}}(\lambda x N^{\mathsf{l}}, \pi'^{\mathsf{l}}_t))V^{\mathsf{l}} * \emptyset \\
&\to_{\text{name}} \lambda n(\lambda x N^{\mathsf{l}})n\pi'^{\mathsf{l}}_t * V^{\mathsf{l}} \to_{\text{name}} (\lambda x N^{\mathsf{l}})V^{\mathsf{l}}\pi'^{\mathsf{l}}_t * \emptyset \\
&\to_{\text{name}} (\lambda x N^{\mathsf{l}})V^{\mathsf{l}} * \pi'^{\mathsf{l}}_t \to_{\text{name}} (\lambda x N^{\mathsf{l}}) * V^{\mathsf{l}} \cdot \pi'^{\mathsf{l}}_t \\
&\to_{\text{name}} N^{\mathsf{l}}[V^{\mathsf{l}}/x] * \pi'^{\mathsf{l}}_t = M'^{\mathsf{l}} * \pi'^{\mathsf{l}}_t \\
\text{(resp. } M^{\mathsf{r}} * \pi^{\mathsf{r}}_t &= \lambda k(k)\lambda x N^{\mathsf{r}} * \mathsf{arg}_{\mathsf{r}}(V^{\mathsf{r}}, \pi'^{\mathsf{r}}_t) \to_{\text{name}} (\mathsf{arg}_{\mathsf{r}}(V^{\mathsf{r}}, \pi'^{\mathsf{r}}_t))\lambda x N^{\mathsf{r}} * \emptyset \\
&\to_{\text{name}} \lambda m(m)V^{\mathsf{r}}\pi'^{\mathsf{r}}_t * \lambda x N^{\mathsf{r}} \to_{\text{name}} (\lambda x N^{\mathsf{r}})V^{\mathsf{r}}\pi'^{\mathsf{r}}_t * \emptyset \\
&\to_{\text{name}} (\lambda x N^{\mathsf{r}})V^{\mathsf{r}} * \pi'^{\mathsf{r}}_t \to_{\text{name}} (\lambda x N^{\mathsf{r}}) * V^{\mathsf{r}} \cdot \pi'^{\mathsf{r}}_t \\
&\to_{\text{name}} N^{\mathsf{r}}[V^{\mathsf{r}}/x] * \pi'^{\mathsf{r}}_t = M'^{\mathsf{r}} * \pi'^{\mathsf{r}}_t \text{)}
\end{aligned}$$

where the last identity holds thanks to lemma 206. ☐

Bibliography

[Acc11] B. Accattoli. *Jumping around the box: graphical and operational studies on Lambda Calculus and Linear Logic*. PhD thesis, Università "La Sapienza", Roma, January 2011.

[AK10] B. Accattoli and D. Kesner. The structural lambda-calculus. In *Proceedings of the 19th EACSL Annual Conference on Computer Science and Logic*, number 6247 in Lecture Notes in Computer Science, pages 381–395. Springer-Verlag, August 2010.

[AK12] B. Accattoli and D. Kesner. Preservation of Strong Normalisation modulo permutations for the structural lambda-calculus. *Logical Methods in Computer Science*, 8(1), 2012.

[AP12] B. Accattoli and L. Paolini. Call-by-Value Solvability, Revisited. In T. Schrijvers and P. Thiemann, editors, *Functional and Logic Programming*, volume 7294 of *Lecture Notes in Computer Science*, pages 4–16. Springer Berlin Heidelberg, 2012.

[Bar84] H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1984.

[Böh68] C. Böhm. Alcune proprietà delle forme β-η-normali nel λ-K-calcolo. *Pubblicazioni dell'Istituto per le Applicazioni del Calcolo*, 696, 1968.

[dC07] D. de Carvalho. *Sémantiques de la logique linéaire et temps de calcul*. PhD thesis, Université Aix-Marseille II, 2007.

[dC09] D. de Carvalho. Execution time of lambda-terms via denotational semantics and intersection types. To appear in *Mathematical Structures in Computer Science*, 2009. Available at `http://arxiv.org/abs/0905.4251`.

[dCPT11] D. de Carvalho, M. Pagani, and L. Tortora de Falco. A Semantic Measure of the Execution Time in Linear Logic. *Theoretical Computer Science, Special issue Girard's Festschrift*, 412(20):1884–1902, 2011.

[dCT12] D. de Carvalho and L. Tortora de Falco. The relational model is injective for Multiplicative Exponential Linear Logic (without weakenings). *Annals of Pure and Applied Logic*, 163(9):1210–1236, September 2012.

[DL07] R. Dyckhoff and S. Lengrand. Call-by-Value λ-calculus and LJQ. *Journal of Logic and Computation*, 17(6):1109–1134, 2007.

[DR89] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical logic*, 28(3):181–203, 1989.

[DR95] V. Danos and L. Regnier. Proof-nets and the Hilbert space. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *Lonon Mathematical Society Lecture Notes Series*, pages 307–328. Cambridge University Press, 1995.

[DR04] V. Danos and L. Regnier. Head Linear Reduction. Submitted for publication, 2004. Available at `http://iml.univ-mrs.fr/~regnier/articles/pam.ps.gz`.

[Ehr05] T. Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(04):615–646, 2005.

[Ehr12] T. Ehrhard. Collapsing non-idempotent intersection types. In P. Cégielski and A. Durand, editors, *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL*, volume 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 259–273, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[ER03] T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1-3):1–41, 2003.

[ER06a] T. Ehrhard and L. Regnier. Böhm trees, Krivine machine and the Taylor expansion of ordinary lambda-terms. *Lecture Notes in Computer Science*, 3988:186–197, 2006.

[ER06b] T. Ehrhard and L. Regnier. Differential interaction nets. *Theoretical Computer Science*, 364(2):166–195, 2006.

[ER08] T. Ehrhard and L. Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theoretical Computer Science*, 403(2-3):347–372, 2008.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

[Hof95] M. Hofmann. Sound and complete axiomatisations of call-by-value control operators. *Mathematical Structures in Computer Science*, 5(4):461–482, 1995.

[HZ09] H. Herbelin and S. Zimmermann. An operational account of call-by-value minimal and classical λ-calculus in "natural deduction" form. *Typed Lambda Calculi and Applications*, pages 142–156, 2009.

[Kri85] J.-L. Krivine. Un intepréteur du λ-calcul. Unpublished notes, Université Paris 7, 1985. Available at `https://www.pps.univ-paris-diderot.fr/~krivine/articles/interprt.pdf`.

[Kri93] J.-L. Krivine. *Lambda-calculus, types and models.* Ellis Horwood, 1993.

[Kri07] J.-L. Krivine. A call-by-name lambda-calculus machine. *Higher-Order and Symbolic Computation*, 20(3):199–207, September 2007.

[Laf95] Y. Lafont. From Proof-Nets to Interaction Nets. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes Series*, pages 225–247. Cambridge University Press, 1995.

[Lan65] P. J. Landin. A correspondence between ALGOL 60 and Church's lambda notation. *Communications of the ACM*, 8:89–101; 158–165, 1965.

[Lau03] O. Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Theoretical Computer Science*, 290(1):161–188, January 2003.

[Ler90] X. Leroy. The ZINC experiment: an economical implementation of the ML language. Technical report 117, INRIA, February 1990.

[LT06] O. Laurent and L. Tortora de Falco. Obsessional cliques: a semantic characterization of bounded time complexity. In *Proceedings of the twenty-first annual IEEE symposium on Logic In Computer Science*, pages 179–188. IEEE Computer Society Press, August 2006.

[Mog89] E. Moggi. Computational lambda-calculus and monads. In *Logic in Computer Science, 1989. LICS'89, Proceedings., Fourth Annual Symposium on*, pages 14–23. IEEE, 1989.

[MOTW95] J. Maraist, M. Odersky, D. N. Turner, and P. Wadler. Call-by-name, Call-by-value, Call-by-need, and the Linear Lambda

Calculus. *Electronic Notes in Theoretical Computer Science*, 1:370–392, 1995.

[MP07] D. Mazza and M. Pagani. The Separation Theorem for Differential Interaction Nets. In N. Dershowitz, editor, *Proceedings of the Ten-Foruth International Conference on Logic for Programming Artificial Intelligence and Reasoning Conference (LPAR 2007)*, volume 4790 of *Lecture Notes in Artificial Intelligence*. Springer, 2007.

[NGP12] P. Nogueira and Á. García-Pérez. Towards Böhm trees for lambda-value: the operational and proof-theoretical machinery. Submitted to *Mathematical Structures in Computer Science*, 2012.

[Pag09] M. Pagani. The Cut-Elimination Thereom for Differential Nets with Boxes. In P.-L. Curien, editor, *Proceedings of the Ninth International Conference on Typed Lambda Calculi and Applications (TLCA 2009)*, Lecture Notes in Computer Science, pages 219–233. Springer, 2009.

[Plo75] G. D. Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theoretical Computer Science*, 1(2):125–159, 1975.

[PR99] L. Paolini and S. Ronchi Della Rocca. Call-by-value Solvability. *Theoretical Informatics and Applications*, 33(6):507–534, November 1999. RAIRO Series, EDP-Sciences.

[PR04] L. Paolini and S. Ronchi Della Rocca. *The Parametric λ-Calculus: a Metamodel for Computation.* Texts in Theoretical Computer Science: An EATCS Series. Springer-Verlag, Berlin, 2004.

[PT10] M. Pagani and L. Tortora de Falco. Strong Normalization Property for Second Order Linear Logic. *Theoretical Computer Science*, 411(2):410–444, 2010.

[Reg92] L. Regnier. *Lambda calcul et réseaux.* PhD thesis, Université Paris 7, 1992.

[Reg94] L. Regnier. Une équivalence sur les lambda-termes. *Theoretical Computer Science*, 126(2):281–292, April 1994.

[Sel01] P. Selinger. Control categories and duality: on the categorical semantics of the lambda-mu calculus. *Mathematical Structures in Computer Science*, 11(02):207–260, 2001.

[Tor00] L. Tortora de Falco. *Réseaux, cohérence et expériences obsessionnelles.* PhD thesis, Université Paris 7, 2000.

[Tor03] L. Tortora de Falco. Obsessional Experiments For Linear Logic Proof-Nets. *Mathematical Structures in Computer Science*, 13(6):799–855, December 2003.

[Tra11] P. Tranquilli. Intuitionistic differential nets and lambda-calculus. *Theoretical Computer Science*, 412(20):1979–1997, 2011.