



HAL
open science

Glueability of resource proof-structures: inverting the Taylor expansion

Giulio Guerrieri, Luc Pellissier, Lorenzo Tortora de Falco

► **To cite this version:**

Giulio Guerrieri, Luc Pellissier, Lorenzo Tortora de Falco. Glueability of resource proof-structures: inverting the Taylor expansion. 28th EACSL Annual Conference on Computer Science Logic, Jan 2020, Barcelone, Spain. 10.4230/LIPIcs.CSL.2020.24 . hal-02957862

HAL Id: hal-02957862

<https://hal.science/hal-02957862v1>

Submitted on 5 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Glueability of resource proof-structures: inverting the Taylor expansion

Giulio Guerrieri

University of Bath, Department of Computer Science, Bath, UK

g.guerrieri@bath.ac.uk

Luc Pellissier

Université de Paris, IRIF, CNRS, F-75013 Paris, France

pellissier@irif.fr

Lorenzo Tortora de Falco

Università Roma Tre, Dipartimento di Matematica e Fisica, Rome, Italy

tortora@uniroma3.it

Abstract

A Multiplicative-Exponential Linear Logic (MELL) proof-structure can be expanded into a set of resource proof-structures: its Taylor expansion. We introduce a new criterion characterizing those sets of resource proof-structures that are part of the Taylor expansion of some MELL proof-structure, through a rewriting system acting both on resource and MELL proof-structures.

2012 ACM Subject Classification Logic → Linear Logic

Keywords and phrases linear logic, Taylor expansion, proof-net, proof-structure

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

Resource λ -calculus and the Taylor expansion Girard’s linear logic (LL, [15]) is a refinement of intuitionistic and classical logic that isolates the infinitary parts of reasoning in two (dual) modalities: the *exponentials* ! and ?. They give a logical status to the operations of memory management such as *copying* and *erasing*: a linear proof corresponds—via Curry–Howard isomorphism—to a program that uses its argument *linearly*, *i.e.* exactly once, while an exponential proof corresponds to a program that can use its argument at will.

The intuition that linear programs are analogous to linear functions (as studied in linear algebra) while exponential programs mirror a more general class of analytic functions got a technical incarnation in Ehrhard’s work [9, 10] on LL-based denotational semantics for the λ -calculus. This investigation has been then internalized in the syntax, yielding the *resource λ -calculus* [5, 11, 14]: there, copying and erasing are forbidden and replaced by the possibility to apply a function to a *bag* of resource λ -terms which specifies how many times an argument can be linearly passed to the function, so as to represent only bounded computations.

The *Taylor expansion* associates with an ordinary λ -term a (generally infinite) set of resource λ -terms, recursively approximating the usual application: the Taylor expansion of the λ -term MN is made of resource λ -terms of the form $t[u_1, \dots, u_n]$, where t is a resource λ -term in the Taylor expansions of M , and $[u_1, \dots, u_n]$ is a bag of arbitrarily finitely many (possibly 0) resource λ -terms in the Taylor expansion of N . Roughly, the idea is to decompose a program into a set of purely “resource-sensitive programs”, all of them containing only bounded (although possibly non-linear) calls to inputs. The notion of Taylor expansion has many applications in the theory of the λ -calculus, *e.g.* in the study of linear head reduction [12], normalization [23, 26], Böhm trees [4, 18], λ -theories [19], intersection types [21]. More generally, understanding the relation between a program and its Taylor expansion renews the logical approach to the quantitative analysis of computation started with the inception of LL.



© Giulio Guerrieri and Luc Pellissier and Lorenzo Tortora de Falco; licensed under Creative Commons License CC-BY

42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

45 A natural question is the *inverse Taylor expansion problem*: how to characterize which
 46 sets of resource λ -terms are contained in the Taylor expansion of a same λ -term? Ehrhard and
 47 Regnier [14] defined a simple *coherence* relation such that a finite set of resource λ -terms is
 48 included in the Taylor expansion of a λ -term if and only if the elements of this set are pairwise
 49 coherent. Coherence is crucial in many structural properties of the resource λ -calculus, such
 50 as in the proof that in the λ -calculus normalization and Taylor expansion commute [12, 14].

51 We aim to solve the inverse Taylor expansion problem in the more general context of LL,
 52 being aware that for LL no coherence relation can solve the problem (see below).

53 **Proof-nets, proof-structures and their Taylor expansion: seeing trees behind graphs** In
 54 the multiplicative-exponential fragment of LL (MELL), linearity and the sharp analysis of
 55 computations naturally lead to represent proofs in a more general *graph*-like syntax instead
 56 of a term-like or tree-like one.¹ Indeed, linear negation is involutive and classical duality
 57 can be interpreted as the possibility of juggling between different conclusions, without a
 58 distinguished output. Graphs representing proofs in MELL are called *proof-nets*: their syntax
 59 is richer and more expressive than the λ -calculus. Contrary to λ -terms, proof-nets are special
 60 inhabitants of the wider land of *proof-structures*: they can be characterized, among proof-
 61 structures, by abstract (geometric) conditions called correctness criteria [15]. The procedure
 62 of cut-elimination can be applied to proof-structures, and proof-nets can also be seen as the
 63 proof-structures with a good behavior with respect to cut-elimination [1]. Proof-structures
 64 can be interpreted in denotational models and proof-nets can be characterized among them
 65 by semantic means [24]. It is then natural to attack problems in the general framework of
 66 proof-structures. In this work, correctness plays no role at all, hence we will only consider
 67 proof-structures and not only proof-nets. Proof-structures are a particular kind of graphs,
 68 whose edges are labeled by MELL formulæ and vertices by MELL connectives, and for which
 69 special subgraphs are highlighted, the *boxes*, representing the parts of the proof-structure that
 70 can be copied and discarded (*i.e.* called an unbounded number of times). A box is delimited
 71 from the rest of a proof-structure by exponential modalities: its border is made of one !-cell,
 72 its principal door, and arbitrarily many ?-cells, its auxiliary doors. Boxes are nested or disjoint
 73 so as to add a tree-like structure to proof-structures *aside* from their graph-like nature.

74 As in λ -calculus, one can define [13] box-free *resource proof-structures*² where !-cells make
 75 resources available boundedly, and the *Taylor expansion* of MELL proof-structures into these
 76 resource proof-structures, that recursively copies the content of the boxes an arbitrary number
 77 of times. In fact, as somehow anticipated by Boudes [3], such a Taylor expansion operation can
 78 be carried on any tree-like structure. This primitive, abstract, notion of Taylor expansion can
 79 then be pulled back to the structure of interest, as shown in [17] and put forth again here.

80 **The question of coherence for proof-structures** The inverse Taylor expansion problem
 81 has a natural counterpart in the world of MELL proof-structures: given a set of resource
 82 proof-structures, is there a MELL proof-structure the expansion of which contains the set?
 83 Pagani and Tasson [22] give the following answer: it is possible to decide whether a finite set of
 84 resource proof-structures is a subset of the Taylor expansion of a same MELL proof-structure
 85 (and even possible to do it in non-deterministic polynomial time); but unlike the λ -calculus,
 86 the structure of the relation “being part of the Taylor expansion of a same proof-structure”
 87 is *much more* complicated than a binary (or even n -ary) coherence. Indeed, for any $n > 1$, it

¹ A term-like object is essentially a tree, with one output (its root) and many inputs (its other leaves).

² Also known as differential proof-structures [6] or differential nets [13, 20, 7] or simple nets [22].

88 is possible to find $n + 1$ resource proof-structures such that any n of them are in the Taylor
 89 expansion of some MELL proof-structure, but there is no MELL proof-structure whose Taylor
 90 expansion has all the $n + 1$ as elements (see our Example 21 and [25, pp. 244-246]).

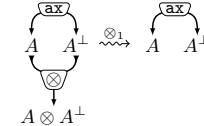
91 In this work, we introduce a new combinatorial criterion, *glueability*, for deciding whether
 92 a set of resource proof-structures is a subset of the Taylor expansion of some MELL proof
 93 structure, based on a rewriting system on sequences of MELL formulæ. Our criterion is more
 94 general (and, we believe, simpler) than the one of [22], which is limited to the *cut-free* case with
 95 *atomic axioms* and characterizes only *finite* sets: we do not have these limitations. We believe
 96 that our criterion is a useful tool for studying proof-structures. We conjecture that it can be
 97 used to show that, for a suitable geometric restriction, a binary coherence relation does exist
 98 for resource proof-structures. It might also shed light on correctness and sequentialization.

99 As the proof-structures we consider are typed, an unrelated difficulty arises: a resource
 100 proof-structure might not be in the Taylor expansion of any MELL proof-structure, not
 101 because it does not respect the structure imposed by the Taylor expansion, but because
 102 its type is impossible.³ To this means, we enrich the MELL proof-structures' syntax with a
 103 "universal" proof-structure: a special \boxtimes -cell (*daimon*) which can have any number of outputs
 104 and any type, and we allow it to appear inside a box, representing information plainly missing
 105 (see Section 8 for more details and the way this matter is handled by Pagani and Tasson [22]).

106 2 Outline and technical issues

107 **The rewritings** The essence of our rewriting system is not located on proof-structures but
 108 on lists of MELL formulæ (Definition 9). In a very down-to-earth way, this rewriting system is
 109 generated by elementary steps akin to rules of sequent calculus read from the *bottom up*: it acts
 110 on a list of conclusions, analogous to a monolaterous right-handed sequent. These steps are
 111 actually more sequentialized than sequent calculus rules, as they do not allow for commutation.
 112 For instance, the rule corresponding to the introduction of a \otimes on the i -th formula, is defined
 113 as $\otimes_i : (\gamma_1, \dots, \gamma_{i-1}, A \otimes B, \gamma_{i+1}, \dots, \gamma_n) \rightarrow (\gamma_1, \dots, \gamma_{i-1}, A, B, \gamma_{i+1}, \dots, \gamma_n)$.

114 These rewrite steps then act on MELL proof-structures, coherently
 115 with their type, by modifying (most of the times, erasing) the cells
 116 directly connected to the conclusion of the proof-structure. Formally,
 117 this means that there is a functor $\mathbf{qMELL}^{\boxtimes}$ from the rewriting steps
 118 into the category **Rel** of sets and relations, associating with a list of formulæ the set of
 119 MELL proof-structures with these conclusions, and to a rewriting a relation implementing it
 120 (Definition 12). The rules *deconstruct* the proof-structure, starting from its conclusions. The
 121 rule \otimes_1 acts by removing a \otimes -cell on the first conclusion, replacing it by two conclusions.



122 These rules can only act on specific proof-structures, and indeed, capture a lot of their
 123 structure: \otimes_i can be applied to a MELL proof-structure R if and only if R has a \otimes -cell in
 124 the conclusion i (as opposed to, say, an axiom). So, in particular, every proof-structure is
 125 completely characterized by any sequence rewriting it to the empty proof-structure.

126 **Naturality** The same rules act also on sets of resource proof-structures, defining the functor
 127 $\mathfrak{P}\mathbf{qDiLL}_0^{\boxtimes}$ from the rewrite steps into the category **Rel** (Definition 17). When carefully
 128 defined, the Taylor expansion induces a *natural transformation* from $\mathfrak{P}\mathbf{qDiLL}_0^{\boxtimes}$ to $\mathbf{qMELL}^{\boxtimes}$

³ Similarly, in the λ -calculus, there is no closed λ -term of type $X \rightarrow Y$ with $X \neq Y$ atomic, but the resource λ -term $(\lambda f.f)[\]$ can be given that type: the empty bag $[\]$ kills any information on the argument.

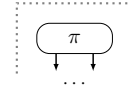
23:4 Glueability of resource proof-structures

(Theorem 18). By applying this naturality repeatedly, we get our characterization (Theorem 20): a set of resource proof-structures Π is a subset of the Taylor expansion of a MELL proof-structure iff there is a sequence rewriting Π to the singleton of the *empty* proof-structure.

The naturality property is not only a mean to get our characterization, but also an interesting result in itself: natural transformations can often be used to express fundamental properties in a mathematical context. In this case, the *Taylor expansion is natural* with respect to the possibility to build a proof-structure (both MELL or resource) by adding a cell to its conclusions or boxing it. Said differently, naturality of the Taylor expansion roughly means that the rewrite rules that deconstruct a MELL proof-structure R and a set of resource proof-structures in the Taylor expansion of R mimic each other.

Quasi-proof-structures and mix Our rewrite rules consume proof-structures from their conclusions. The rule corresponding to boxes in MELL opens a box by deleting its principal door (a !-cell) and its border, while for a resource proof-structure it deletes a !-cell and separates the different copies of the content of the box (possibly) represented by such a !-cell. This operation is problematic in a twofold way. In a resource proof-structure, where the border of boxes is not marked, it is not clear how to identify such copies. On the other side, in a MELL proof-structure the content of a box is not to be treated as if it was at the same level as what is outside of the box: it can be copied many times or erased, while what is outside boxes cannot, and treating the content in the same way as the outside suppresses this distinction, which is crucial in LL. So, we need to remember that the content of a box, even if it is at depth 0 (*i.e.* not contained in any other box) after erasing the box wrapping it by means of our rewrite rules, is not to be mixed with the rest of the structure at depth 0.

In order for our proof-structures to provide this information, we need to generalize them and consider that a proof-structure can have not just a tree of boxes, but a *forest*: this yields the notion of *quasi-proof-structure* (Definition 1).

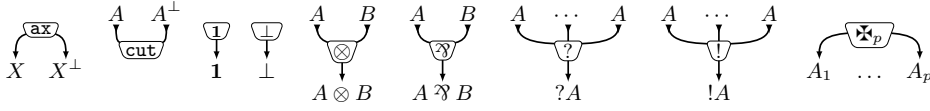


In this way, according to our rewrite rules, opening a box by deleting its principal door amounts to taking a box in the tree and disconnecting it from its root, creating a new tree. We draw this by surrounding elements having the same root with a dashed box, open from the bottom, remembering the phantom presence of the border of the box, below, even if it were erased. This allows one to open the box only when it is “alone” (see Definition 11).

This is not merely a technical remark, as this generalization gives a status to the *mix* rule of LL: indeed, mixing two proofs amounts to taking two proofs and considering them one, without any other modifications. Here, it amounts to taking two proofs, each with its box-tree, and considering them as one by merging the roots of their trees (see the *mix* step of Definition 11). We embed this design decision up to the level of formulæ, which are segregated in different zones that have to be mixed before interacting (see the notion of partition of a finite sequence of formulas in Section 3).

Geometric invariance and emptiness: the filled Taylor expansion The use of forests instead of trees for the nesting structure of boxes, where the different roots are thought of as the contents of long-gone boxes, has an interesting consequence in the Taylor expansion: indeed, an element of the Taylor expansion of a proof-structure contains an arbitrary number of copies of the contents of the boxes, in particular *zero*. If we think of the part at depth 0 of a MELL proof-structure as inside an invisible box, its content can be deleted in some elements of the Taylor expansion just as any other box.⁴ As erasing completely conclusions

⁴ The dual case, of copying the contents of a box, poses no problem in our approach.



■ **Figure 1** Cells, with their labels and their typed inputs and outputs (ordered from left to right).

would cause the Taylor expansion not preserve the conclusions (which would lead to technical complications), we introduce the *filled Taylor expansion* (Definition 8), which contains not only the elements of the usual Taylor expansion, but also elements of the Taylor expansion where one component has been erased and replaced by a \boxtimes -cell (*daimon*), representing a lack of information, apart from the number and types of the conclusions.

Atomic axioms Our paper first focuses on the case where proof-structures are restricted to *atomic axioms*. In Section 7 we sketch how to adapt our method to the non-atomic case.

3 Proof-structures and the Taylor expansion

MELL formulæ and (quasi-)proof-structures Given a countably infinite set of propositional variables X, Y, Z, \dots , MELL *formulæ* are defined by the following inductive grammar:

$$A, B ::= X \mid X^\perp \mid \mathbf{1} \mid \perp \mid A \otimes B \mid A \wp B \mid !A \mid ?A$$

Linear negation is defined via De Morgan laws $\mathbf{1}^\perp = \perp$, $(A \otimes B)^\perp = A^\perp \wp B^\perp$ and $(!A)^\perp = ?A$, so as to be involutive, *i.e.* $A^{\perp\perp} = A$. Given a list $\Gamma = (A_1, \dots, A_m)$ of MELL formulas, a *partition* of Γ is a list $(\Gamma_1, \dots, \Gamma_n)$ of lists of MELL formulas such that there are $0 = i_0 < \dots < i_n = m$ with $\Gamma_j = (A_{i_{j-1}+1}, \dots, A_{i_j})$ for all $1 \leq j \leq n$; such a partition of Γ is also denoted by $(A_1, \dots, A_{i_1}; \dots; A_{i_{n-1}+1}, \dots, A_m)$, with lists separated by semi-colons.

We reuse the syntax of proof-structures given in [17] and sketch here its main features. We suppose known definitions of graphs, rooted trees, and morphisms of these structures. In what follows we will speak of *tails* in a graph: “hanging” edges with only one vertex. This can be implemented either by adding special vertices or using [2]’s graphs.

If an edge e is incoming in (resp. outgoing from) a vertex v , we say that e is a *input* (resp. *output*) of v . The reflexive-transitive closure of a tree τ is denoted by τ° : the operator $(\cdot)^\circ$ lifts to a functor from the category of trees to the category of directed graphs.

► **Definition 1.** A module M is a (finite) directed graph with:

- vertices v labeled by $\ell(v) \in \{\mathbf{ax}, \mathbf{cut}, \mathbf{1}, \perp, \otimes, \wp, ?, !\} \cup \{\boxtimes_p \mid p \in \mathbb{N}\}$, the type of v ;
 - edges e labeled by a MELL formula $c(e)$, the type of e ;
 - an order $<_M$ that is total on the tails of $|M|$ and on the inputs of each vertex of type \wp, \otimes .
- Moreover, all the vertices verify the conditions of Figure 1.⁵

A quasi-proof-structure is a triple $R = (|R|, \mathcal{F}, \mathbf{box})$ where:

- $|R|$ is a module with no input tails, called the module of R ;
- \mathcal{F} is a forest of rooted trees with no input tails, called the box-forest of R ;
- $\mathbf{box}: |R| \rightarrow \mathcal{F}^\circ$ is a morphism of directed graphs, the box-function of R , which induces a partial bijection from the inputs of the vertices of type $!$ and the edges in \mathcal{F} , and such that:

⁵ Note that there are no conditions on the types of the outputs of vertices of type \boxtimes (*i.e.* of type \boxtimes_p for some $p \in \mathbb{N}$); and the outputs of vertices of type \mathbf{ax} must have *atomic* types.

206 ■ for any vertices v, v' with an edge from v' to v , if $\text{box}(v) \neq \text{box}(v')$ then $\ell(v) \in \{!, ?\}$.⁶
 207 Moreover, for any output tails e_1, e_2, e_3 in $|R|$ which are outputs of the vertices v_1, v_2, v_3 ,
 208 respectively, if $e_1 <_{|R|} e_2 <_{|R|} e_3$ then it is impossible that $\text{box}(v_1) = \text{box}(v_3) \neq \text{box}(v_2)$.⁷

209 A quasi-proof-structure $R = (|R|, \mathcal{F}, \text{box})$ is:

- 210 1. MELL^{\boxtimes} if all vertices in $|R|$ of type $!$ have exactly one input, and the partial bijection
- 211 induced by box from the inputs of the vertices of type $!$ in $|R|$ and the edges in \mathcal{F} is total.
- 212 2. MELL if it is MELL^{\boxtimes} and, for every vertex v in $|R|$ of type \boxtimes , one has $\text{box}^{-1}(\text{box}(v)) = \{v\}$
- 213 and $\text{box}(v)$ is not a root of the box-forest \mathcal{F} of R .
- 214 3. $\text{DiLL}_0^{\boxtimes}$ if the box-forest \mathcal{F} of R is just a juxtaposition of roots.
- 215 4. DiLL_0 (or resource) if it is $\text{DiLL}_0^{\boxtimes}$ and there is no vertex in $|R|$ of type \boxtimes .

216 For the previous systems, a proof-structure is a quasi-proof-structure whose box-forest is a tree.

217 Our MELL proof-structure (i.e. a MELL quasi-proof-structure that is also a proof-structure)
 218 corresponds to the usual notion of MELL proof-structure (as in [8]) except that we also allow
 219 the presence of a box filled only by a *daimon* (i.e. a vertex of type \boxtimes). The *empty* (DiLL_0 and
 220 MELL) proof-structure—whose module and box-forest are empty graphs—is denoted by ε .

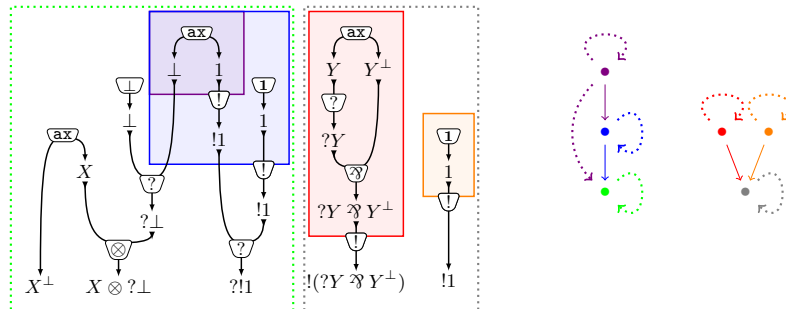
221 Given a quasi-proof-structure $R = (|R|, \mathcal{F}, \text{box})$, the output tails of $|R|$ are the *conclusions*
 222 of R . So, the pre-images of the roots of \mathcal{F} via box partition the conclusions of R in a list of
 223 lists of such conclusions. The *type* of R is the list of lists of the types of these conclusions.
 224 We often identify the conclusions of R with a finite initial segment of \mathbb{N} .

225 By definition of graph morphism, two conclusions in two distinct lists in the type of a
 226 quasi-proof-structure R are in two distinct connected components of $|R|$; so, if R is not a
 227 proof-structure then $|R|$ contains several connected components. Thus, R can be seen as a
 228 list of proof-structures, its *components*, one for each root in its box-forest.

229 A non-root vertex v in the box-forest \mathcal{F} induces a subgraph of \mathcal{F}° of all vertices above it
 230 and edges connecting them. The pre-image of this subgraph through box is the *box* of v and
 231 the conditions on box in Definition 1 translate the usual nesting condition for LL boxes.

232 In quasi-proof-structures, we speak of *cells* instead of vertices, and, for a cell of type l ,
 233 a l -cell. A \boxtimes -cell is a \boxtimes_p -cell for some $p \in \mathbb{N}$. An *hypothesis cell* is a cell without inputs.

234 ► **Example 2.** The graph in Figure 2 is a MELL quasi-proof-structure. The colored areas
 235 represent the pre-images of boxes, and the dashed boxes represent the pre-images of roots.



■ **Figure 2** A MELL quasi-proof-structure R , its box-forest \mathcal{F}_R (without dotted lines) and the reflexive-transitive closure \mathcal{F}_R° of \mathcal{F}_R (with also dotted lines).

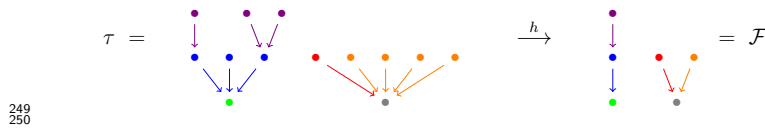
⁶ Roughly, it says that the border of a box is made of (inputs of) vertices of type $!$ or $?$.

⁷ This is a technical condition that simplifies the definition of the rewrite rules in Section 4. Note that $\text{box}(v_1), \text{box}(v_2), \text{box}(v_3)$ are necessarily roots in \mathcal{F} , since box is a morphism of directed graphs.

236 **The Taylor expansion** Proof-structures have a tree structure made explicit by their box-
 237 function. Following [17], the definition of the Taylor expansion uses this tree structure: first,
 238 we define how to “*expand*” a tree—and more generally a forest—via a generalization of the
 239 notion of thick subtree [3] (Definition 3; roughly, a thick subforest of a box-forest says the
 240 number of copies of each box to be taken, iteratively), we then take all the expansions of the
 241 tree structure of a proof-structure and we *pull* the approximations *back* to the underlying
 242 graphs (Definition 5), finally we *forget* the tree structures associated with them (Definition 6).

243 ► **Definition 3** (thick subforest). *Let τ be a forest of rooted trees. A thick subforest of τ is a*
 244 *pair (σ, h) of a forest σ of rooted trees and a graph morphism $h: \sigma \rightarrow \tau$ whose restriction to*
 245 *the roots of σ is bijective.*

246 ► **Example 4.** The following is a graphical presentation of a thick subforest (τ, h) of the
 247 box-forest \mathcal{F} of the quasi-proof-structure in Figure 2, where the graph morphism $h: \tau \rightarrow \mathcal{F}$
 248 is depicted chromatically (same color means same image via h).



249
 250
 251 Intuitively, it means that τ is obtained from \mathcal{F} by taking 3 copies of the blue box, 1 copy of
 252 the red box and 4 copies of the orange box; in the first (resp. second; third) copy of the blue
 253 box, 1 copy (resp. 0 copies; 2 copies) of the purple box has been taken.

254 ► **Definition 5** (proto-Taylor expansion). *Let $R = (|R|, \mathcal{F}_R, \text{box}_R)$ be a quasi-proof-structure.*
 255 *The proto-Taylor expansion of R is the set $\mathcal{T}^{\text{proto}}(R)$ of thick subforests of \mathcal{F}_R .*

256 *Let $t = (\tau_t, h_t) \in \mathcal{T}^{\text{proto}}(R)$. The t -expansion of R is the pullback (R_t, p_t, p_R) below,*
 257 *computed in the category of directed graphs and graph morphisms.*

$$\begin{array}{ccc}
 R_t & \xrightarrow{p_t} & \tau_t^\circ \\
 \downarrow p_R & \lrcorner & \downarrow h_t^\circ \\
 |R| & \xrightarrow{\text{box}_R} & \mathcal{F}_R^\circ
 \end{array}$$

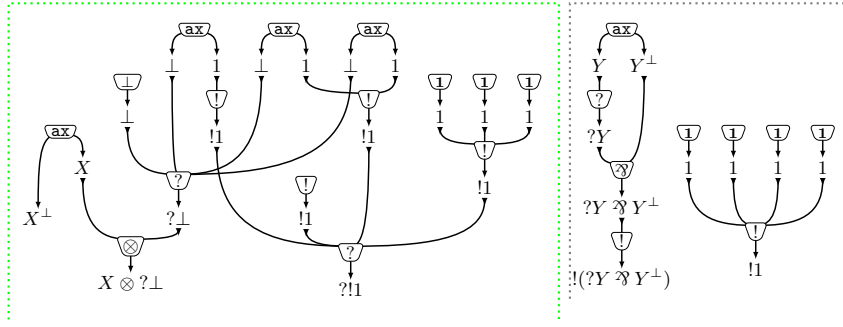
258

259 Given a quasi-proof-structure R and $t = (\tau_t, h_t) \in \mathcal{T}^{\text{proto}}(R)$, the directed graph R_t
 260 inherits labels on vertices and edges by composition with the graph morphism $p_R: R_t \rightarrow |R|$.

261 Let $[\tau_t]$ be the forest made up of the roots of τ_t and $\iota: \tau_t \rightarrow [\tau_t]$ be the graph morphism
 262 sending each vertex of τ_t to the root below it; ι° induces by post-composition a morphism
 263 $\bar{h}_t = \iota^\circ \circ p_t: R_t \rightarrow [\tau_t]^\circ$. The triple $(R_t, [\tau_t], \bar{h}_t)$ is a DiLL_0 quasi-proof-structure, and it is a
 264 DiLL_0 proof-structure if R is a proof-structure. We can then define the *Taylor expansion* $\mathcal{T}(R)$
 265 of a quasi-proof-structure R (an example of an element of a Taylor expansion is in Figure 3).

266 ► **Definition 6** (Taylor expansion). *Let R be a quasi-proof-structure. The Taylor expansion of*
 267 *R is the set of DiLL_0 quasi-proof-structures $\mathcal{T}(R) = \{(R_t, [\tau_t], \bar{h}_t) \mid t = (\tau_t, h_t) \in \mathcal{T}^{\text{proto}}(R)\}$.*

268 An element $(R_t, [\tau_t], \bar{h}_t)$ of the Taylor expansion of a quasi-proof-structure R has much
 269 less structure than the pullback (R_t, p_t, p_R) : the latter indeed is a DiLL_0 quasi-proof-structure
 270 R_t coming with its projections $|R| \xleftarrow{p_R} R_t \xrightarrow{p_t} \tau_t^\circ$, which establish a precise correspondence
 271 between cells and edges of R_t and cells and edges of R : a cell in R_t is labeled (via the
 272 projections) by both the cell of $|R|$ and the branch of the box-forest of R it arose from. But
 273 $(R_t, [\tau_t], \bar{h}_t)$ where R_t is without its projections p_t and p_R loses the correspondence with R .



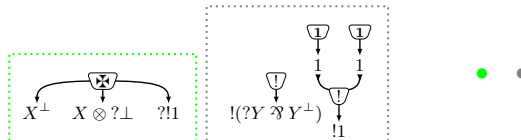
■ **Figure 3** The element of the Taylor expansion of the MELL quasi-proof-structure R in Figure 2, obtained from the element of $\mathcal{T}^{\text{proto}}(R)$ depicted in Example 4.

274 ▶ **Remark 7.** By definition, the Taylor expansion preserves conclusions: there is a bijection φ
 275 from the conclusions of a quasi-proof-structure R to the ones in each element ρ of $\mathcal{T}(R)$ such
 276 that i and $\varphi(i)$ have the same type; and the types of R and ρ are the same (as a list of lists).

277 **The filled Taylor expansion** As discussed in Section 2 (p. 4), our method needs to “represent”
 278 the emptiness introduced by the Taylor expansion (taking 0 copies of a box) so as to preserve
 279 the conclusions. So, an element of the *filled Taylor expansion* $\mathcal{T}^{\boxtimes}(R)$ of a quasi-proof-structure
 280 R (an example is in Figure 4) is obtained from an element of $\mathcal{T}(R)$ where a whole component
 281 can be erased and replaced by a \boxtimes -cell with the same conclusions (hence $\mathcal{T}(R) \subseteq \mathcal{T}^{\boxtimes}(R)$).

282 ▶ **Definition 8** (filled Taylor expansion). *The emptying of a DiLL₀ quasi-proof-structure*
 283 $\rho = (|\rho|, \mathcal{F}, \text{box})$ *relatively to some roots* r_1, \dots, r_n *of* \mathcal{F} *is the same as* ρ *but with the*
 284 *components of* r_1, \dots, r_n *replaced by a* \boxtimes -*cell with the same conclusions as in* ρ .

285 *The filled Taylor expansion* $\mathcal{T}^{\boxtimes}(R)$ *of a quasi-proof-structure* R *is the set of all the*
 286 *emptyings of the elements of its Taylor expansion* $\mathcal{T}(R)$.



■ **Figure 4** An element of the filled Taylor expansion of the MELL quasi-proof-structure in Figure 2.

287 4 Means of destruction: unwinding MELL quasi-proof-structures

288 Our aim is to deconstruct proof-structures (be they MELL $^{\boxtimes}$ or DiLL₀) from their conclusions.
 289 To do that, we introduce a category of rules of deconstruction. The morphisms of this category
 290 are sequences of deconstructing rules, acting on lists of lists of formulæ. These morphisms
 291 act through functors on quasi-proof-structures, exhibiting their sequential structure.

292 ▶ **Definition 9** (the category **Path**). *Let* **Path** *be the category whose*

293 ■ *objects are lists* $\Gamma = (\Gamma_1; \dots; \Gamma_n)$ *of lists of MELL formulæ;*

294 ■ *arrows are freely generated by the elementary paths in* Figure 5.

295 *We call a path any arrow* $\xi: \Gamma \rightarrow \Gamma'$. *We write the composition of paths without symbols and*
 296 *in the diagrammatic order, so, if* $\xi: \Gamma \rightarrow \Gamma'$ *and* $\xi': \Gamma' \rightarrow \Gamma''$, $\xi\xi': \Gamma \rightarrow \Gamma''$.

$$\begin{array}{l}
(\Gamma_1; \dots; \Gamma_k, c(i), c(i+1), \Gamma'_k; \dots; \Gamma_n) \xrightarrow{\text{exc}_i} (\Gamma_1; \dots; \Gamma_k, c(i+1), c(i), \Gamma'_k; \dots; \Gamma_n) \\
(\Gamma_1; \dots; \Gamma_k, c(i), c(i+1), \Gamma'_k; \dots; \Gamma_n) \xrightarrow{\text{mix}_i} (\Gamma_1; \dots; \Gamma_k, c(i); c(i+1), \Gamma'_k; \dots; \Gamma_n) \\
(\Gamma_1; \dots; \Gamma_k; c(i), c(i+1); \Gamma_{k+2}; \dots; \Gamma_n) \xrightarrow{\text{ax}_i} (\Gamma_1; \dots; \Gamma_k; \Gamma_{k+2}; \dots; \Gamma_n) \quad \text{with } c(i) = A = c(i+1)^\perp \\
(\Gamma_1; \dots; \Gamma_k; \dots; \Gamma_n) \xrightarrow{\text{cut}^\perp} (\Gamma_1; \dots; \Gamma_k, c(i), c(i+1); \dots; \Gamma_n) \quad \text{with } c(i) = A = c(i+1)^\perp \\
(\Gamma_1; \dots; \Gamma_k; \Gamma_{k+1}, c(i); \Gamma_{k+2}; \dots; \Gamma_n) \xrightarrow{\mathbf{x}_i} (\Gamma_1; \dots; \Gamma_k; \Gamma_{k+2}; \dots; \Gamma_n) \\
(\Gamma_1; \dots; \Gamma_k; c(i); \Gamma_{k+2}; \dots; \Gamma_n) \xrightarrow{\mathbf{1}_i} (\Gamma_1; \dots; \Gamma_k; \Gamma_{k+2}; \dots; \Gamma_n) \quad \text{with } c(i) = \mathbf{1} \\
(\Gamma_1; \dots; \Gamma_k; c(i); \Gamma_{k+2}; \dots; \Gamma_n) \xrightarrow{\perp_i} (\Gamma_1; \dots; \Gamma_k; \Gamma_{k+2}; \dots; \Gamma_n) \quad \text{with } c(i) = \perp \\
(\Gamma_1; \dots; \Gamma_k, c(i); \dots; \Gamma_n) \xrightarrow{\otimes_i} (\Gamma_1; \dots; \Gamma_k, A, B; \dots; \Gamma_n) \quad \text{with } c(i) = A \otimes B \\
(\Gamma_1; \dots; \Gamma_k, c(i); \dots; \Gamma_n) \xrightarrow{\wp_i} (\Gamma_1; \dots; \Gamma_k, A, B; \dots; \Gamma_n) \quad \text{with } c(i) = A \wp B \\
(\Gamma_1; \dots; \Gamma_k, c(i); \dots; \Gamma_n) \xrightarrow{?_i} (\Gamma_1; \dots; \Gamma_k, ?A, ?A; \dots; \Gamma_n) \quad \text{with } c(i) = ?A \\
(\Gamma_1; \dots; \Gamma_k, c(i); \dots; \Gamma_n) \xrightarrow{?_i} (\Gamma_1; \dots; \Gamma_k, A; \dots; \Gamma_n) \quad \text{with } c(i) = ?A \\
(\Gamma_1; \dots; \Gamma_k; c(i); \Gamma_{k+2}; \dots; \Gamma_n) \xrightarrow{?_i} (\Gamma_1; \dots; \Gamma_k; \Gamma_{k+2}; \dots; \Gamma_n) \quad \text{with } c(i) = ?A \\
(\Gamma_1; \dots; ?\Gamma_k, c(i); \dots; \Gamma_n) \xrightarrow{\text{Box}_i} (\Gamma_1; \dots; ?\Gamma_k, A; \dots; \Gamma_n) \quad \text{with } c(i) = !A
\end{array}$$

■ **Figure 5** The generators of **Path**. In the source $\Gamma = (A_1, \dots, A_{i_1}; \dots; A_{i_{m-1}+1}, \dots, A_{i_n})$ of each arrow, $c(i)$ denotes the i^{th} formula in the flattening $(A_1, \dots, A_{i_1}, \dots, A_{i_{m-1}+1}, \dots, A_{i_n})$ of Γ .

297 ▶ **Example 10.** $\wp_1 \wp_2 \wp_3 \otimes_1 \otimes_3 \text{exc}_1 \text{exc}_2 \text{mix}_2 \text{ax}_1 \text{exc}_2 \text{mix}_2 \text{ax}_1 \text{ax}_1$ is a path of type
298 $((X \otimes Y^\perp) \wp ((Y \otimes Z^\perp) \wp (X^\perp \wp Z))) \longrightarrow \varepsilon$.

299 We will tend to forget about exchanges and perform them silently (as it is customary, for
300 instance, in most presentations of sequent calculi).

301 The category **Path** acts on $\text{MELL}^{\mathbf{x}}$ quasi-proof-structures, exhibiting a sequential struc-
302 ture in their construction. For Γ a list of list of MELL formulæ, $\mathbf{qMELL}^{\mathbf{x}}(\Gamma)$ is the set of
303 $\text{MELL}^{\mathbf{x}}$ quasi-proof-structures of type Γ . To ease the reading of the rewrite rules acting on a
304 $\text{MELL}^{\mathbf{x}}$ quasi-proof-structures R , we will only draw the parts of R belonging to the relevant
305 component; so, for instance, if we are interested in an ax -cell rooted in the conclusions i and

306 $i+1$, which is the only cell in a component, we will write $\overset{\text{ax}}{\underbrace{\quad}_i \quad}_{i+1}$ ignoring the rest.

307 ▶ **Definition 11** (action of paths on MELL quasi-proof-structures). *An elementary path $a: \Gamma \rightarrow$*
308 *Γ' defines a relation $\mathfrak{a} \subseteq \mathbf{qMELL}^{\mathbf{x}}(\Gamma) \times \mathbf{qMELL}^{\mathbf{x}}(\Gamma')$ (the action of a) as the smallest*
309 *relation containing all the cases in Figure 6, with the following remarks:*

310 **mix** read in reverse, a quasi-proof-structure with two components is in relation with a proof-
311 structure with the same module but the two roots of said components merged.

312 **hypothesis** if $a \in \{\text{ax}_i, \mathbf{x}_i, \mathbf{1}_i, \perp_i, ?_w i\}$, the rules have all in common to act by deleting a cell
313 without inputs that is the only cell in its component. We have drawn the axiom case in
314 Figure 6c, the others vary only by their number of conclusions.

315 **cut** read in reverse, a quasi-proof-structure with two conclusions i and $i+1$ is in relation
316 with the quasi-proof-structure where these two conclusions are cut. This rule, from left to
317 right, is non-deterministic (as there are many possible cuts).

318 **binary multiplicatives** these rules delete a binary connective. We have only drawn the \otimes
319 case in Figure 6e, the \wp case is similar.

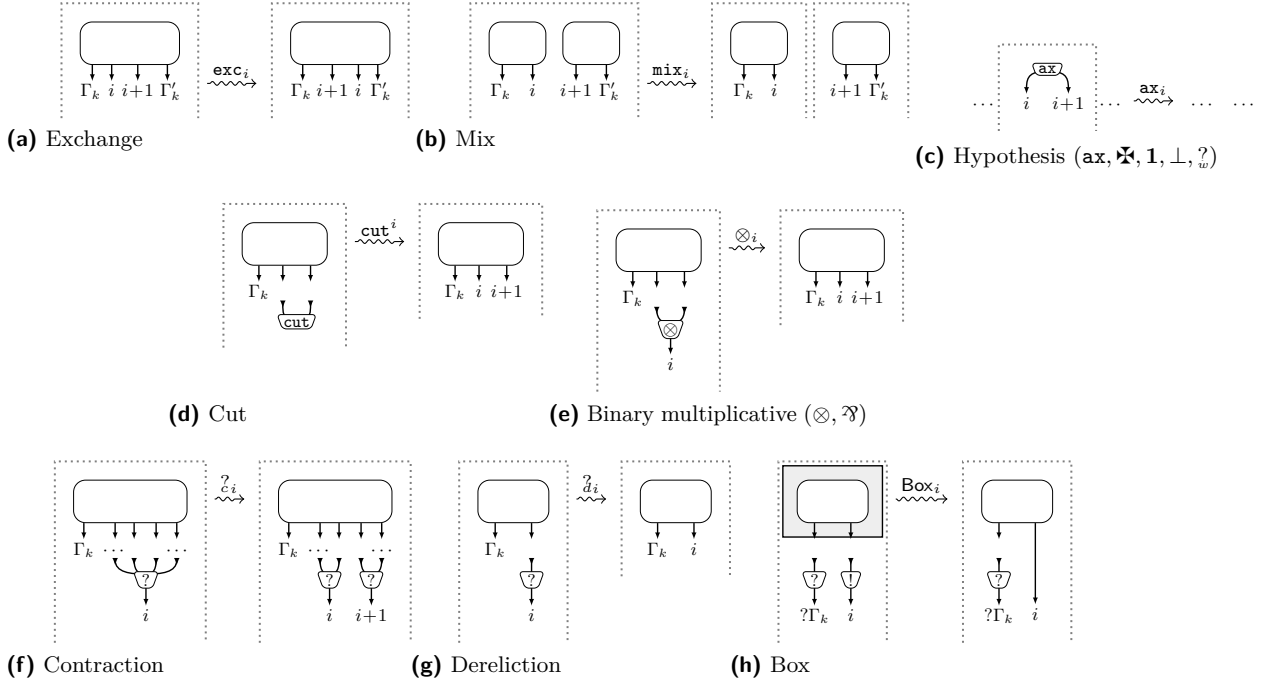
320 **contraction** splits a $?$ -cell with $h+k+2$ inputs into two $?$ -cells with $h+1$ and $k+1$ inputs,
321 respectively.

322 **dereliction** only applies if the $?$ -cell (with 1 input) does not shift a level in the box-forest.

323 **box** only applies if a box (and its frontier) is alone in its component.

324 This definition of the rewrite system is extended to define a relation $\mathfrak{L} \subseteq \mathbf{qMELL}^{\mathbf{x}}(\Gamma) \times$
325 $\mathbf{qMELL}^{\mathbf{x}}(\Gamma')$ (the action of any path $\xi: \Gamma \rightarrow \Gamma'$) by composition of relations.

23:10 Glueability of resource proof-structures



■ **Figure 6** Actions of elementary paths on $\text{MELL}^{\mathfrak{X}}$ quasi-proof-structures.

326 Given two $\text{MELL}^{\mathfrak{X}}$ quasi-proof-structures R and R' , we say that a rule a applies to R if
 327 there is a finite sequence of exchanges $\text{exc}_{i_1} \cdots \text{exc}_{i_n}$ such that $R \xrightarrow{\text{exc}_{i_1} \cdots \text{exc}_{i_n} a} R'$.

328 ► **Definition 12** (the functor $\text{qMELL}^{\mathfrak{X}}$). We define a functor $\text{qMELL}^{\mathfrak{X}}: \text{Path} \rightarrow \text{Rel}$ by:

329 ■ on objects: $\text{qMELL}^{\mathfrak{X}}(\Gamma)$ is the set of $\text{MELL}^{\mathfrak{X}}$ quasi-proof-structures of type Γ ;

330 ■ on morphisms: for $\xi: \Gamma \rightarrow \Gamma'$, $\text{qMELL}^{\mathfrak{X}}(\xi) = \xi$ (see Definition 11).

331 Our rewrite rules enjoy two useful properties, expressed by Propositions 13 and 15.

332 ► **Proposition 13** (co-functionality). Let $\xi: \Gamma \rightarrow \Gamma'$ be a path. The relation ξ is a co-function
 333 on the sets of underlying graphs, that is, a function $\xi^{\text{op}}: \text{qMELL}^{\mathfrak{X}}(\Gamma') \rightarrow \text{qMELL}^{\mathfrak{X}}(\Gamma)$.

334 ► **Lemma 14** (applicability of rules). Let R be a non-empty $\text{MELL}^{\mathfrak{X}}$ quasi-proof-structure.
 335 There exists a conclusion i such that:

336 ■ either a rule in $\{\text{ax}_i, \mathbf{1}_i, \perp_i, \otimes_i, \wp_i, ?_i, ?_{w_i}, \text{cut}^i, \mathfrak{X}_i, \text{Box}_i\}$ applies to R ;

337 ■ or $R \xrightarrow{\text{mix}_i} R'$ (where the conclusions affected by mix_i are $i-k, \dots, i, i+1, \dots, i+l$) and
 338 $i-k, \dots, i$ are all the conclusions of either a box or an hypothesis cell, and one of the
 339 components of R' coincides with this cell or box (and its border).

340 Proposition 13 and Lemma 14 are proven by simple inspection of the rewrite rules of Figure 6.

341 ► **Proposition 15** (termination). Let R be a $\text{MELL}^{\mathfrak{X}}$ quasi-proof-structure of type Γ . There
 342 exists a path $\xi: \Gamma \rightarrow \varepsilon$ such that $R \xrightarrow{\xi} \varepsilon$

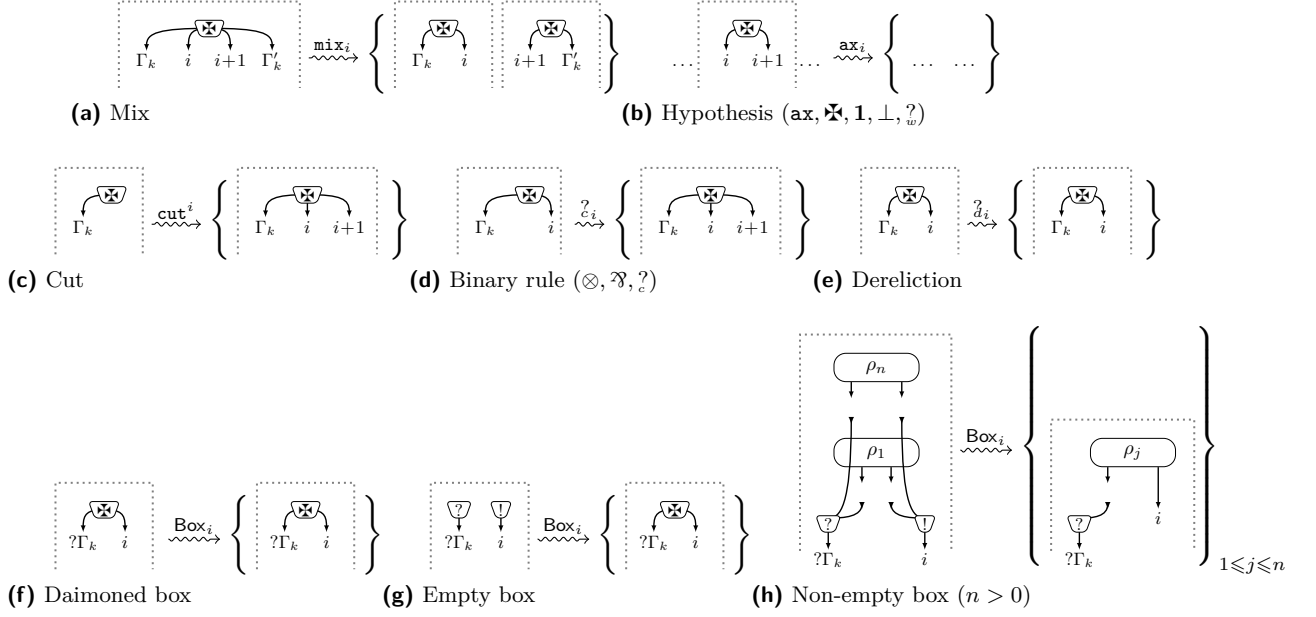
343 To prove Proposition 15, it is enough to apply Lemma 14 and show that the size of $\text{MELL}^{\mathfrak{X}}$
 344 quasi-proof-structures decreases for each application of the rules in Figure 6, according to
 345 the following definition of size. The size of a proof-structure R is the couple (p, q) where

346 ■ p is the multiset of the number of inputs of each $?$ -cell in R ;

347 ■ q is the number of cells not labeled by \mathfrak{X} in R .

348 The size of a quasi-proof-structure R is the (finite) multiset of the sizes of its components.

349 Multisets are ordered as usual, couples are ordered lexicographically.



■ **Figure 7** Actions of elementary paths on λ -cells and on a box in $\mathbf{qDiLL}_0^{\lambda}$.

5 Naturality of unwinding $\mathbf{DiLL}_0^{\lambda}$ quasi-proof-structures

For Γ a list of lists of MELL formulæ, $\mathbf{qDiLL}_0^{\lambda}(\Gamma)$ is the set of $\mathbf{DiLL}_0^{\lambda}$ quasi-proof-structures of type Γ . For any set X , its powerset is denoted by $\mathfrak{P}(X)$.

► **Definition 16** (action of paths on $\mathbf{DiLL}_0^{\lambda}$ quasi-proof-structures). *An elementary path $a: \Gamma \rightarrow \Gamma'$ defines a relation $\mathfrak{a} \subseteq \mathbf{qDiLL}_0^{\lambda}(\Gamma) \times \mathfrak{P}(\mathbf{qDiLL}_0^{\lambda}(\Gamma'))$ (the action of a) by the rules in Figure 6 (except Figure 6h, and with all the already remarked notes) and in Figure 7.*

We extend this relation on $\mathfrak{P}(\mathbf{qDiLL}_0^{\lambda}(\Gamma)) \times \mathfrak{P}(\mathbf{qDiLL}_0^{\lambda}(\Gamma'))$ by the monad multiplication of $X \mapsto \mathfrak{P}(X)$ and define $\mathfrak{a} \circ \mathfrak{b}$ (the action of any path $\xi: \Gamma \rightarrow \Gamma'$) by composition of relations.

Roughly, all the rewrite rules in Figure 7—except Figure 7h—mimic the behavior of the corresponding rule in Figure 6 using a λ -cell. Note that in Figure 7g a λ -cell is created.

The non-empty box rule in Figure 7h requires that, on the left of \mathfrak{Box}_i , ρ_j is not connected to $\rho_{j'}$ for $j \neq j'$, except for the $!$ -cell and the $?$ -cells in the conclusions. Read in reverse, the rule associates with a non-empty finite set of \mathbf{DiLL}_0 quasi-proof-structures $\{\rho_1, \dots, \rho_n\}$ the merging of ρ_1, \dots, ρ_n , that is the \mathbf{DiLL}_0 quasi-proof-structure depicted on the left of \mathfrak{Box}_i .

► **Definition 17** (the functor $\mathfrak{P}\mathbf{qDiLL}_0^{\lambda}$). *We define a functor $\mathfrak{P}\mathbf{qDiLL}_0^{\lambda}: \mathbf{Path} \rightarrow \mathbf{Rel}$ by:*

- on objects: for Γ a list of lists of MELL formulæ, $\mathfrak{P}\mathbf{qDiLL}_0^{\lambda}(\Gamma) = \mathfrak{P}(\mathbf{qDiLL}_0^{\lambda}(\Gamma))$, the set of sets of $\mathbf{DiLL}_0^{\lambda}$ proof-structures of type Γ ;
- on morphisms: for $\xi: \Gamma \rightarrow \Gamma'$, $\mathfrak{P}\mathbf{qDiLL}_0^{\lambda}(\xi) = \mathfrak{a}$ (see Definition 16).

► **Theorem 18** (naturality). *The filled Taylor expansion defines a natural transformation $\mathfrak{T}^{\lambda}: \mathfrak{P}\mathbf{qDiLL}_0^{\lambda} \Rightarrow \mathbf{qMELL}^{\lambda}: \mathbf{Path} \rightarrow \mathbf{Rel}$ by: $(\Pi, R) \in \mathfrak{T}_F^{\lambda}$ iff $\Pi \subseteq \mathcal{T}^{\lambda}(R)$ and the type of R is Γ . Moreover, if Π is a set of \mathbf{DiLL}_0 proof-structures with $\Pi \mathfrak{a} \Pi'$ and $\Pi' \subseteq \mathcal{T}(R')$, then R is a MELL proof-structure and $\Pi \subseteq \mathcal{T}(R)$, where R is such that $R \mathfrak{a} R'$.⁸*

⁸ The part of the statement after “moreover” is our way to control the presence of λ -cells.

23:12 Glueability of resource proof-structures

372 In other words, the following diagram commutes for every path $\xi: \Gamma \rightarrow \Gamma'$.

$$\begin{array}{ccc}
 \mathfrak{P}\mathfrak{qDiLL}_0^{\mathfrak{X}}(\Gamma) & \xrightarrow{\mathfrak{P}\mathfrak{qDiLL}_0^{\mathfrak{X}}(\xi)} & \mathfrak{P}\mathfrak{qDiLL}_0^{\mathfrak{X}}(\Gamma') \\
 \downarrow \mathfrak{T}_\Gamma^{\mathfrak{X}} & & \downarrow \mathfrak{T}_{\Gamma'}^{\mathfrak{X}} \\
 \mathfrak{qMELL}^{\mathfrak{X}}(\Gamma) & \xrightarrow{\mathfrak{qMELL}^{\mathfrak{X}}(\xi)} & \mathfrak{qMELL}^{\mathfrak{X}}(\Gamma')
 \end{array}$$

373

374 It means that given $\Pi \rightsquigarrow \Pi'$, where $\Pi' \subseteq \mathcal{T}^{\mathfrak{X}}(R')$, we can simulate backwards the rewrit-
 375 ing to R (this is where the co-functionality of the rewriting steps expressed by Proposition 13
 376 comes handy); and given $R \rightsquigarrow R'$, we can simulate the rewriting for any $\Pi \subseteq \mathcal{T}^{\mathfrak{X}}(R)$.

6 Glueability of DiLL_0 quasi-proof-structures

378 Naturality (Theorem 18) allows us to characterize the sets of DiLL_0 proof-structures that are
 379 in the Taylor expansion of some MELL proof-structure (Theorem 20 below).

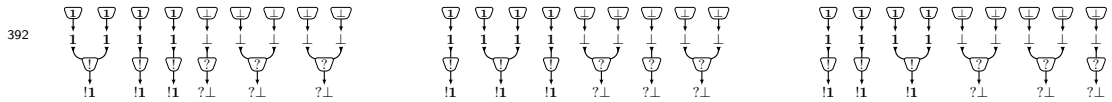
380 ► **Definition 19** (glueability). *We say that a set Π of $\text{DiLL}_0^{\mathfrak{X}}$ quasi-proof-structures is glueable,*
 381 *if there exists a path ξ such that $\Pi \xrightarrow{\xi} \{\varepsilon\}$.*

382 ► **Theorem 20** (gluing criterion). *Let Π be a set of DiLL_0 proof-structures: Π is glueable if*
 383 *and only if $\Pi \subseteq \mathcal{T}(R)$ for some MELL proof-structure R .*

384 **Proof.** If $\Pi \subseteq \mathcal{T}(R)$ for some MELL proof-structure R , then by termination (Proposition 15)
 385 $R \xrightarrow{\xi} \varepsilon$ for some path ξ , and so $\Pi \xrightarrow{\xi} \{\varepsilon\}$ by naturality (Theorem 18, as $\mathcal{T}^{\mathfrak{X}}(\varepsilon) = \{\varepsilon\}$).

386 Conversely, if $\Pi \xrightarrow{\xi} \{\varepsilon\}$ for some path ξ , then by naturality (Theorem 18, as $\mathcal{T}(\varepsilon) = \{\varepsilon\}$
 387 and Π is a set of DiLL_0 proof-structures) $\Pi \subseteq \mathcal{T}(R)$ for some MELL proof-structure R . ◀

388 ► **Example 21.** The three DiLL_0 proof-structures ρ_1, ρ_2, ρ_3 below are not glueable as a
 389 whole, but are glueable two by two. In fact, there is no MELL proof-structure whose Taylor
 390 expansion contains ρ_1, ρ_2, ρ_3 , but any pair of them is in the Taylor expansion of some MELL
 391 proof-structure. This is a slight variant of the example in [25, pp. 244-246].



393 An example of the action of a path starting from a DiLL_0 proof-structure ρ and ending in
 394 $\{\varepsilon\}$ can be found in Figures 8 and 9. Note that it is by no means the shortest possible path.
 395 When replayed backwards, it induces a MELL proof-structure R such that $\rho \in \mathcal{T}(R)$.

7 Non-atomic axioms

397 From now on, we relax the definition of quasi-proof-structure (Definition 1 and Figure 1) so
 398 that the outputs of any ax -cell are labeled by dual MELL formulæ, not necessarily atomic. We
 399 can extend our results to this more general setting, with some technical complications. Indeed,
 400 the rewrite rule for contraction has to be modified. Consider a set of DiLL_0 proof-structures
 401 consisting of just a singleton which is a \mathfrak{X} -cell. The contraction rule rewrites it as:

402

$$\begin{array}{c} \mathfrak{X} \\ \downarrow \downarrow \\ !A^\perp \quad !A^\perp \quad ?A \end{array} \xrightarrow{\mathfrak{C}_3} \left\{ \begin{array}{c} \mathfrak{X} \\ \downarrow \downarrow \\ !A^\perp \quad !A^\perp \quad ?A \quad ?A \end{array} \right\} \text{ which is then in the Taylor expansion of } \begin{array}{c} \text{ax} \\ \text{ax} \\ \downarrow \downarrow \\ !A^\perp \quad !A^\perp \quad ?A \quad ?A \end{array}$$

403

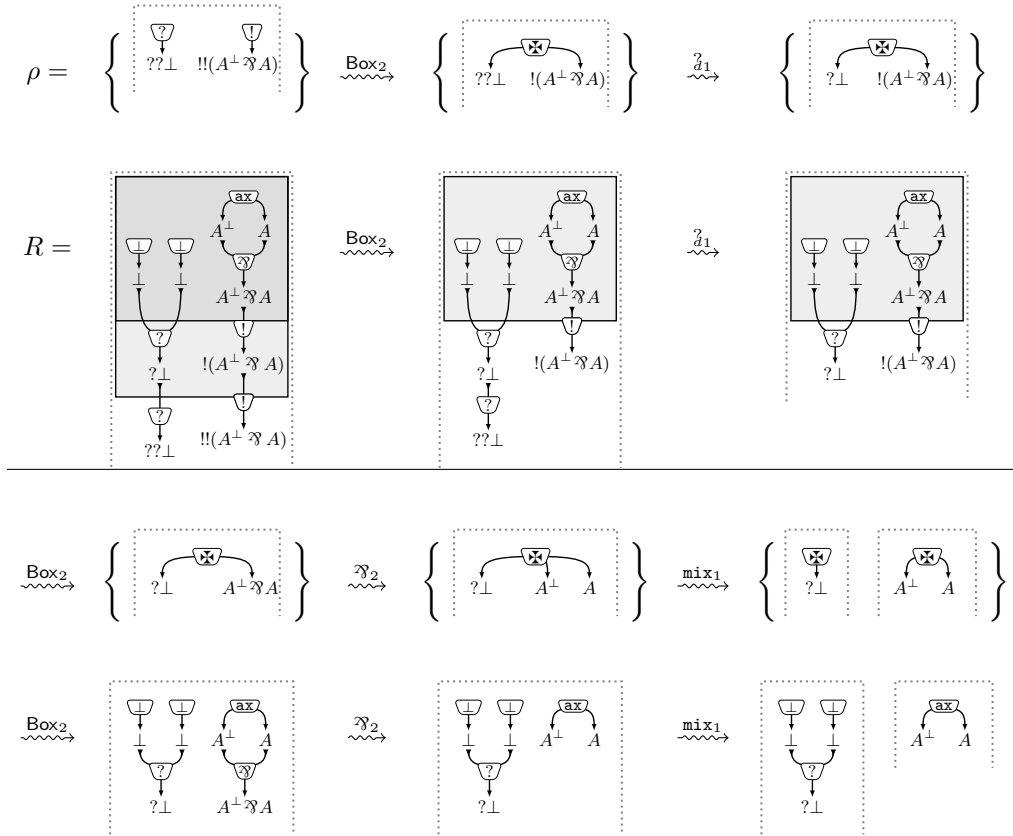


Figure 8 The path $\text{Box}_2 \overset{?}{d}_1 \text{Box}_2 \overset{?}{d}_2 \text{mix}_1 \text{ax}_{2,3} \overset{?}{c}_1 \overset{?}{d}_2 \text{mix}_1 \perp_2 \overset{?}{d}_1 \perp_1$ witnessing that $\rho \in \mathcal{T}(R)$ (to be continued on Figure 9).

404 on which no contraction rewrite rule $?$ can be applied backwards, breaking the naturality.
 405 The failure of the naturality is actually due to the failure of Proposition 13 in the case of the
 406 rewrite rule $?: \overset{?}{\rightsquigarrow}^{\text{op}}$ is functional but not total.

407 The solution to this conundrum lies in changing the contraction rule for $\text{DiLL}_0^{\boxtimes}$ quasi-
 408 proof-structures, by explicitly adding $?$ -cells. Hence, the application of a contraction step $?$
 409 in the $\text{DiLL}_0^{\boxtimes}$ quasi-proof-structures precludes the possibility of anything else but a $?$ -cell on
 410 the MELL^{\boxtimes} side, which allows the contraction step $?$ to be applied backwards.

411 In turn, this forces us to change the definition of the filled Taylor expansion into a η -filled
 412 Taylor expansion, which has to include elements where a \boxtimes -cell (representing an empty
 413 component) has some of its outputs connected to $?$ -cells.

414 ► **Definition 22** (η -filled Taylor expansion). *The η -emptying of DiLL_0 quasi-proof-structure*
 415 $\rho = (|\rho|, \mathcal{F}, \text{box})$ *relatively to some roots r_1, \dots, r_n of \mathcal{F} it is the same as ρ but with the*
 416 *components of r_1, \dots, r_n replaced by a \boxtimes -cell with the same conclusions as in ρ with its*
 417 *outputs possibly connected to a $?$ -cell in conclusion i , if there is a $?$ -cell in conclusion i in R .*

418 *The η -filled Taylor expansion $\mathcal{T}_\eta^{\boxtimes}(R)$ of a quasi-proof-structure R is the set of all the*
 419 *emptyings of the elements of its Taylor expansion $\mathcal{T}(R)$, relatively to all components, and all*
 420 *possible choices of $?$ -cells conclusions of R .*

421 Note that the η -filled Taylor expansion contains all the elements of the filled Taylor
 422 expansion and some more, such as the one in Figure 10.

23:14 Glueability of resource proof-structures

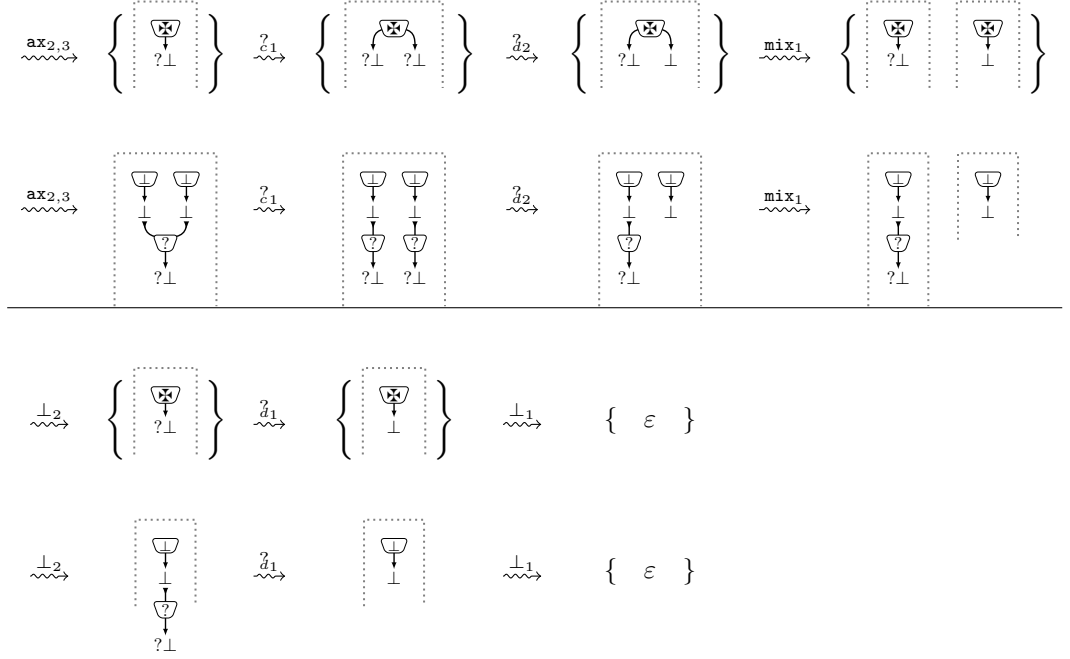


Figure 9 The path $\text{Box}_2 \stackrel{?}{d}_1 \text{Box}_2 \stackrel{?}{d}_2 \text{mix}_1 \text{ax}_{2,3} \stackrel{?}{c}_1 \stackrel{?}{d}_2 \text{mix}_1 \perp_2 \stackrel{?}{d}_1 \perp_1$ witnessing that $\rho \in \mathcal{T}(R)$ (continued from Figure 8).

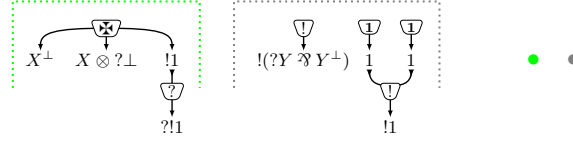
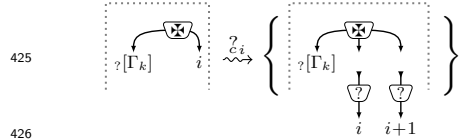


Figure 10 An element of the η -filled Taylor expansion of the MELL quasi-proof-structure in Fig. 2.

423 Functors $\mathbf{qMELL}^{\boxtimes}$ and $\mathfrak{P}\mathbf{qDiLL}_0^{\boxtimes}$ are defined as before (Def. 12 and 17, respectively),⁹
 424 except that the image of $\mathfrak{P}\mathbf{qDiLL}_0^{\boxtimes}$ on the generator $\stackrel{?}{c}_i$ (Figure 7d) is changed to



426

427 where $?\Gamma_k$ signifies that some of the conclusions of Γ_k might be connected to the \boxtimes -cell
 428 through a $?$ -cell. We can prove similarly our main results.

429 **► Theorem 23** (naturality with η). *The η -filled Taylor expansion defines a natural transform-*
 430 *ation $\mathfrak{T}_\eta^{\boxtimes}: \mathfrak{P}\mathbf{qDiLL}_0^{\boxtimes} \Rightarrow \mathbf{qMELL}^{\boxtimes}: \mathbf{Path} \rightarrow \mathbf{Rel}$ by: $(\Pi, R) \in \mathfrak{T}_\eta^{\boxtimes} \Gamma$ iff $\Pi \subseteq \mathcal{T}_\eta^{\boxtimes}(R)$ and the*
 431 *type of R is Γ . Moreover, if Π is a set of \mathbf{DiLL}_0 proof-structures with $\Pi \stackrel{\xi}{\sim} \Pi'$ and $\Pi' \subseteq \mathcal{T}(R')$,*
 432 *then R is a MELL proof-structure and $\Pi \subseteq \mathcal{T}(R)$, where R is such that $R \stackrel{\xi}{\sim} R'$.*

433 **► Theorem 24** (gluing criterion with η). *Let Π be a set of \mathbf{DiLL}_0 proof-structures, not*
 434 *necessarily with atomic axioms: Π is glueable iff $\Pi \subseteq \mathcal{T}(R)$ for some MELL proof-structure R .*

⁹ Remember that now, for Γ a list of list of MELL formulæ, $\mathbf{qMELL}^{\boxtimes}(\Gamma)$ (resp. $\mathbf{qDiLL}_0^{\boxtimes}(\Gamma)$) is the set of MELL ^{\boxtimes} (resp. $\mathbf{DiLL}_0^{\boxtimes}$) quasi-proof-structures of type Γ , possibly with non-atomic axioms.

8 Conclusions and perspectives

✕-cells inside boxes Our gluing criterion (Theorem 20) solves the inverse Taylor expansion problem in a “asymmetric” way: we characterize the sets of DiLL_0 proof-structures that are included in the Taylor expansion of some MELL proof-structure, but DiLL_0 proof-structures have no occurrences of ✕-cells, while a MELL proof-structure possibly contains ✕-cells inside boxes (see Definition 1). Not only this asymmetry is technically inevitable, but it reflects on the fact that some glueable set of DiLL_0 proof-structure might not contain any information on the content of some box (which is reified in MELL by a ✕-cell), or worse that, given the types, no content can fill that box. Think of the DiLL_0 proof-structure ρ made only of a !-cell with no inputs and one output of type ! X , where X is atomic: $\{\rho\}$ is glueable but the only MELL proof-structure R such that $\{\rho\} \subseteq \mathcal{T}(R)$ is made of a box containing a ✕-cell.

This asymmetry is also present in Pagani and Tasson’s characterization [22], even if not particularly emphasized: their Theorem 2 (analogous to the left-to-right part of our Theorem 20) assumes not only that the rewriting starting from a finite set of DiLL_0 proof-structures terminates but also that it ends on a MELL proof-structure (without ✕-cells, which ensures that there exists a MELL proof-structure without ✕-cells filling all the empty boxes).

The λ -calculus, connection and coherence Our rewriting system and glueability criterion should help to prove the existence of a binary coherence for elements of the Taylor expansion of a fragment of MELL-proof-structures, extending the one that exists for resource λ -terms. We can remark that the glueability criterion is actually an extension of the criterion for resource λ -terms: indeed, in the case of the λ -calculus, there are three rewriting steps, corresponding to the abstraction, the application and the variable (which can be encoded in our rewriting steps), and coherence is defined inductively: if a set of resource λ -terms is coherent, then any set of resource λ -term that rewrites to it is also coherent.

Presented in this way, the main difference lies not in the rewriting system but in that, in the λ -terms’ case, the structure of any resource λ -term determines the rewriting path, while, for DiLL_0 proof-structures, we have to quantify existentially over all possible paths. This can not be avoided and is a consequence of the fact that proof-structures do not have a tree-structure, contrary to λ -terms.

Moreover, it is possible to match and mix different sequences of rewritings. Indeed, consider three DiLL_0 proof-structures pairwise glueable. Proving that they are glueable as a whole amounts to computing a rewriting path from the rewriting paths witnessing the three glueabilities. Our paths were designed with that mixing-and-matching operation in mind, in the particular case where the boxes are connected. This is reminiscent of [16], where we also showed that a certain property enjoyed by the λ -calculus can be extended to proof-structures, provided they are connected inside boxes. We leave that work to a subsequent article.

Functoriality and naturality Our functorial point of view on proof-structures can unify many results. Let us cite two:

- a sequent calculus proof of $\vdash \Gamma$ can be translated into a path from the empty sequence into Γ . This could be the starting point for the formulation of a new correctness criterion;
- the category **Path** can be extended with higher structure, allowing to represent cut-elimination. The functors \mathbf{qMELL}^{\times} and $\mathfrak{P}\mathbf{qDiLL}_0^{\times}$ can also be extended to such higher functors, proving the commutation of cut-elimination and the Taylor expansion.

478 **Acknowledgments** This work has been partially funded by the EPSRC grant EP/R029121/1
 479 “Typed Lambda-Calculi with Sharing and Unsharing” and the ANR project Rapido (ANR-
 480 14-CE35-0007).

 481 **References**

- 482 **1** Denis B echet. Minimality of the correctness criterion for multiplicative proof nets. *Mathematical*
 483 *Structures in Computer Science*, 8(6):543–558, 1998.
- 484 **2** Dennis V. Borisov and Yuri I. Manin. *Generalized Operads and Their Inner Cohomomorphisms*,
 485 volume 265 of *Progress in Mathematics*, pages 247–308. Birkh user Basel, Basel, 2007.
 486 doi:10.1007/978-3-7643-8608-5_4.
- 487 **3** Pierre Boudes. Thick subtrees, games and experiments. In *Typed Lambda Calculi and*
 488 *Applications, 9th International Conference (TLCA 2009)*, volume 5608 of *Lecture Notes in*
 489 *Computer Science*, pages 65–79. Springer, 2009. doi:10.1007/978-3-642-02273-9_7.
- 490 **4** Pierre Boudes, Fanny He, and Michele Pagani. A characterization of the Taylor expansion
 491 of lambda-terms. In *Computer Science Logic 2013 (CSL 2013)*, volume 23 of *LIPICs*, pages
 492 101–115. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013. doi:10.4230/LIPICs.
 493 CSL.2013.101.
- 494 **5** G erard Boudol. The lambda-calculus with multiplicities (abstract). In *4th International*
 495 *Conference on Concurrency Theory (CONCUR ’93)*, volume 715 of *Lecture Notes in Computer*
 496 *Science*, pages 1–6. Springer, 1993. doi:10.1007/3-540-57208-2_1.
- 497 **6** Daniel de Carvalho. The relational model is injective for multiplicative exponential linear
 498 logic. In *25th Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of
 499 *LIPICs*, pages 41:1–41:19. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:
 500 10.4230/LIPICs.CSL.2016.41.
- 501 **7** Daniel de Carvalho. Taylor expansion in linear logic is invertible. *Logical Methods in Computer*
 502 *Science*, 14(4), 2018. doi:10.23638/LMCS-14(4:21)2018.
- 503 **8** Daniel de Carvalho and Lorenzo Tortora de Falco. The relational model is injective for
 504 multiplicative exponential linear logic (without weakenings). *Annals of Pure and Applied Logic*,
 505 163(9):1210–1236, 2012. doi:10.1016/j.apal.2012.01.004.
- 506 **9** Thomas Ehrhard. On K othe sequence spaces and linear logic. *Mathematical Structures in*
 507 *Computer Science*, 12(5):579–623, 2002. doi:10.1017/S0960129502003729.
- 508 **10** Thomas Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–
 509 646, 2005. doi:10.1017/S0960129504004645.
- 510 **11** Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer*
 511 *Science*, 309(1-3):1–41, 2003. doi:10.1016/S0304-3975(03)00392-X.
- 512 **12** Thomas Ehrhard and Laurent Regnier. B ohm Trees, Krivine’s Machine and the Taylor
 513 Expansion of Lambda-Terms. In *Second Conference on Computability in Europe (CiE 2006)*,
 514 volume 3988 of *Lecture Notes in Computer Science*, pages 186–197. Springer, 2006. doi:
 515 10.1007/11780342_20.
- 516 **13** Thomas Ehrhard and Laurent Regnier. Differential interaction nets. *Theoretical Computer*
 517 *Science*, 364(2):166–195, 2006. doi:10.1016/j.tcs.2006.08.003.
- 518 **14** Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor expansion of ordinary
 519 lambda-terms. *Theoretical Computer Science*, 403(2-3):347–372, 2008. doi:10.1016/j.tcs.
 520 2008.06.001.
- 521 **15** Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. doi:
 522 10.1016/0304-3975(87)90045-4.
- 523 **16** Giulio Guerrieri, Luc Pellissier, and Lorenzo Tortora de Falco. Computing connected proof(-
 524 structure)s from their Taylor expansion. In *1st International Conference on Formal Structures*
 525 *for Computation and Deduction (FSCD 2016)*, volume 52 of *LIPICs*, pages 20:1–20:18. Schloss
 526 Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.FSCD.2016.20.

- 527 **17** Giulio Guerrieri, Luc Pellissier, and Lorenzo Tortora de Falco. Proof-net as graph, Taylor
528 expansion as pullback. In *Logic, Language, Information, and Computation - 26th International*
529 *Workshop (WoLLIC 2019)*, volume 11541 of *Lecture Notes in Computer Science*, pages 282–300.
530 Springer, 2019. doi:10.1007/978-3-662-59533-6_18.
- 531 **18** Emma Kerinec, Giulio Manzonetto, and Michele Pagani. Revisiting call-by-value Böhm trees
532 in light of their Taylor expansion. *CoRR*, abs/1809.02659, 2018. URL: <http://arxiv.org/abs/1809.02659>.
- 533 **19** Giulio Manzonetto and Domenico Ruoppolo. Relational graph models, Taylor expansion
534 and extensionality. *Electronic Notes in Theoretical Computer Science*, 308:245–272, 2014.
535 doi:10.1016/j.entcs.2014.10.014.
- 536 **20** Damiano Mazza and Michele Pagani. The separation theorem for differential interaction
537 nets. In *Logic for Programming, Artificial Intelligence, and Reasoning, 14th International*
538 *Conference (LPAR 2007)*, volume 4790 of *Lecture Notes in Computer Science*, pages 393–407.
539 Springer, 2007. doi:10.1007/978-3-540-75560-9_29.
- 540 **21** Damiano Mazza, Luc Pellissier, and Pierre Vial. Polyadic approximations, fibrations and
541 intersection types. *PACMPL*, 2(POPL):6:1–6:28, 2018. doi:10.1145/3158094.
- 542 **22** Michele Pagani and Christine Tasson. The inverse Taylor expansion problem in linear logic.
543 In *24th Annual Symposium on Logic in Computer Science (LICS 2009)*, pages 222–231. IEEE
544 Computer Society, 2009. doi:10.1109/LICS.2009.35.
- 545 **23** Michele Pagani, Christine Tasson, and Lionel Vaux. Strong normalizability as a finiteness struc-
546 ture via the Taylor expansion of λ -terms. In *Foundations of Software Science and Computation*
547 *Structures - 19th International Conference (FOSSACS 2016)*, volume 9634 of *Lecture Notes in*
548 *Computer Science*, pages 408–423. Springer, 2016. doi:10.1007/978-3-662-49630-5_24.
- 549 **24** Christian Retoré. A semantic characterisation of the correctness of a proof net. *Mathematical*
550 *Structures in Computer Science*, 7(5):445–452, 1997. doi:10.1017/S096012959700234X.
- 551 **25** Christine Tasson. *Sémantiques et Syntaxes Vectorielles de la Logique Linéaire*. PhD thesis,
552 Université Paris Diderot, France, December 2009. URL: <https://tel.archives-ouvertes.fr/tel-00440752>.
- 553 **26** Lionel Vaux. Taylor expansion, lambda-reduction and normalization. In *26th EACSL Annual*
554 *Conference on Computer Science Logic (CSL 2017)*, volume 82 of *LIPICs*, pages 39:1–39:16.
555 Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.CSL.2017.39.