Linear realisability over nets and second order quantification

Adrien Ragot, Thomas Seiller, Lorenzo Tortora de Falco

Linear realisability over nets and second order quantification

Adrien Ragot
Université Sorbonne Paris Nord, LIPN – UMR 7030 CNRS
Dipartimento di Matematica e Fisica – Università degli Studi Roma Tre
Thomas Seiller
CNRS, LIPN – UMR 7030 Université Sorbonne Paris Nord
Lorenzo Tortora de Falco
Dipartimento di Matematica e Fisica – Università degli Studi Roma Tre

Submission to the 7th edition of the International Workshop on Trends in Linear Logic and its Applications, (TLLA2023).

Acknowledgments. A. Ragot is supported by a VINCI PhD fellowship from the Franco-Italian Université. T. Seiller is partially supported by the DIM RFSI project CoHOp, and the ANR ANR-22-CE48-0003-01 project DySCo.

Context

Realisability is a technique that extracts the computational content of proofs [Miq09]. It was first introduced in 1945 by Kleene for Heyting Arithmetic – an Intuitionnistic axiomatization of arithmetic – based on the codes of Gödel's partial recursive functions [Kle45]. Fixing an untyped computational model the methodology of Realisability is based on two aspects:

- Types are given a computational status: the interpretation of a type[1] A is a set of programs $[\![A]\!]$, which behave similarly – its element are called *realizers* of A.
- A simple process transforming the proofs of the realized logic in programs is defined, introducing a non trivial predicate on programs, namely, some programs represent a proof, the *correct* programs, while others do not, the *incorrect* programs.

A theorem of *adequacy* or *soundness* usually follows, e.g. each proof of A corresponds to a realizer of A. However, not all realizers are correct programs thus not all realizers represent a proof. In fact, it was revealed by realisability models based on *orthogonality* that the presence of incorrect programs is crucial to give a computational status to correctness.

The models of Realisability involving a notion of orthogonality appeared years after the introduction of Realisability by Kleene. At the time realisability was only considered for intuistionnistic logics due to their 'constructive' nature, and it is only in 1990 that Jean-Louis Krivine introduced *classical realisability* aiming at extending realisability techniques to classical logic. This construction is based on an extension of the untyped lambda calculus, but, in order to capture a given context (stack) to potentially restore it later, the syntax is not only extended with the call/cc operator but also with a countably infinite set of *stack constants*. As a consequence, (as in Kleene's realisability) only *some* of the programs represent a proof, namely those not containing stack constants.

This introduction of "incorrect" terms is essential, as it introduces in the syntax semantic information [NPS16] that can be used to *test* correct (and incorrect) terms. This concept of testing is captured by the definition of an orthogonality relation (here between terms and stacks), which is used to define the interpretation of types (as the set of terms passing a given set of tests).

In parallel with the work of Krivine, similar realisability constructions have been introduced by Jean-Yves Girard in order to interpret Linear Logic. While the orthogonality construction was clearly put forth in Ludics, the ideas and first occurrences can be traced back to the first model of geometry of interaction (GoI) [Gir87], restricted to multiplicative linear logic, which interpreted proofs as permutations.

Later GoI construction took several diverse forms, generalising permutations by operators in a C*-algebra (GOI1 [Gir89], GOI2 [Gir88]), first-order prefix rewriting (GOI3) [Gir95], or von Neumann Algebras (GOI5) [Gir11].

In a series of recent papers [Sei12; Sei16; Sei17; Sei13; Sei15], Thomas Seiller proposed a combinatorial approach to the Geometry of Interaction, *interaction graphs*, which specialises to all the previous 'geometries' of interaction proposed by Girard. It is crucial to note that this work on GOI constructs the types of Linear Logic via a realisabilty method, involving orthogonality within the computational model of interaction graphs. However, proofs are interpreted in these models as abstract objects (generalisations of dynamical systems) which remain far from the general intuition of what a proof is.

This is where our work starts: we extend the use of realizability techniques to Linear Logic in an *untyped variant* of the well known and 'canonical' context of proof nets; at first to Multiplicative Linear Logic, and then together with second order quantification. We obtain the results of soundness (e.g. adequacy) and completeness both for

[1] Equivalently, having the Curry–Howard correspondence in mind, A is a formula.

MLL and MLL$^{✠}$ – with furthermore assumptions on the interpretation basis. Soundness is also true at the second–order for MLL_2 proofs. Moreover we show that the types constructed by induction for both the multiplicative and second–order preserve the *finite testability* [2]. In particular this is true for the types capturing the proofs of the multiplicative fragment: this is done by encoding the Danos Regnier criterion [DR89] in MLL$^{✠}$ proofs, we provide, to our knowledge, the first proper proof of the folklore result which states that 'tests of A are proofs of its negation'. We are still investigating how to capture the proofs of the second order multiplicative fragment while remaining finitely testable. We believe this will lead to a novel correctness criterion for second order multiplicative proof structures.

Summary of our work

As a computational model we chose the model of *nets* a modern formulation – as hypergraphs – of the model of proof structures introduced by Jean Yves Girard in his seminal paper.

Definition 1 (Directed labelled hypergraph)**.** Given a set V we denote $\mathcal{P}_{\leq}(V)$ the set of finite and totally ordered subsets of V. Suppose given a set L of *labels*.

A *directed (L-labelled) hypergraph* is a tuple $(V, E, \mathrm{s}, \mathrm{t}, \ell)$ where V is a set of positions, and E is a set of elements called *links*, $\mathrm{s} : E \to \mathcal{P}_{\leq}(V)$ is the *source map*, $\mathrm{t} : E \to \mathcal{P}_{\leq}(V)$ is the *target map*, and $\ell : E \to L$ is the *labelling* map.

The conclusion of an hypergraph is a position which is source of no link e. An *ordered hypergraph* $(\mathcal{H}, a)$ is an hypergraph together with an order on its conclusions.

Remark 2. We consider all the hypergraphs to be ordered.

Definition 3. Given two hypergraphs $\mathcal{H}_1 = (V_1, E_1, \mathrm{s}_1, \mathrm{t}_1, \ell_1)$ and $\mathcal{H}_2 = (V_2, E_2, \mathrm{s}_2, \mathrm{t}_2, \ell_2)$ their *sum* is

$$\mathcal{H}_1 + \mathcal{H}_2 \triangleq (V_1 \cup V_2, E_1 \uplus E_2, \mathrm{s}_1 \uplus \mathrm{s}_2, \mathrm{t}_1 \uplus \mathrm{t}_2, \ell_1 \uplus \ell_2).$$

Remark 4. The sum is defined for hypergraphs which have disjoint set of links but the vertices *need not* to be disjoint. The definition can be adapted to any pair of hypergraphs after a carefull renaming of their links. Thus we consider each sum to occur between two hypergraphs with disjoint links.

Notation 5. A link e will be written as $\langle u \rhd_l v \rangle$ where $\mathrm{s}(e) = u$, $\mathrm{t}(e) = v$, and $\ell(e) = l$. With this notation, a directed labelled hypergraph with no isolated position (e.g. that are neither output or input of an edge) can always be written as a formal sum

$$\sum_{e \in E} \langle \mathrm{s}(e) \rhd_{\ell(e)} \mathrm{t}(e) \rangle.$$

The definition is given in all generality but the proof–structures corresponds to a subclass of directed hypergraphs.

Definition 6. An hypergraph $\mathcal{H} = (V, E, \mathrm{t}, \mathrm{s})$ is:

- *surjective* whenever $\bigcup_{e \in E} \mathrm{t}(e) = V$. Meaning each position of the hypergraph is the target of some link.
- *source–disjoint* if the sets $\mathrm{s}(e)$ for $e \in E$ are pairwise disjoint.
- *target–disjoint* if the sets $\mathrm{t}(e)$ for $e \in E$ are pairwise disjoint.

Definition 7 (Multiplicative module)**.** The set of labels is fixed as $L = \{✠, \otimes, ⅋, \mathrm{cut}\}$. A *multiplicative module* is an ordered hypergraph $(\mathcal{H}, a)$ which is target and source disjoint, and such that ✠-labelled links have no inputs, cut-labelled links have exactly two inputs and no outputs, and $\otimes$- and ⅋-labelled links have exactly two inputs and one output.

In other words, $\mathcal{H}$ is a sum (preserving the source–disjoint and target–disjoint properties) of links of the form:

$$\langle \rhd_{✠} p_1, \ldots, p_n \rangle, \langle p_1, p_2 \rhd_{\otimes} p \rangle, \langle p_1, p_2 \rhd_{⅋} p \rangle, \langle p_1, p_2 \rhd_{\mathrm{cut}} \rangle$$

At this point we can define multiplicative proof structures, for simplicity we call them nets (while the terminology proof nets remains unchanged).

Definition 8 (Multiplicative net)**.** A *multiplicative net* is a surjective multiplicative module.

The model of nets comes with a notion of computation which correspond to a rewriting of the hypergraph.

Definition 9 (Types of cut)**.** Given a multiplicative net S the *type* of cut link $c = \langle p, q \rhd \rangle$ occuring in S is the pair of labels of the links of output p and q. Thus there are 6 *types* of cuts (up to symmetry). More precisely, we distinguish:

- *multiplicative* cuts, of type $(\otimes/⅋)$;
- *clash* cuts, of type $(\otimes/\otimes)$ or $(⅋/⅋)$;
- *glueing* cuts, of type $(✠/✠)$;
- *non–homogeneous* cuts, *reversible cuts* of type $(\otimes/✠)$ and *irreversible cuts* of type $(⅋/✠)$.

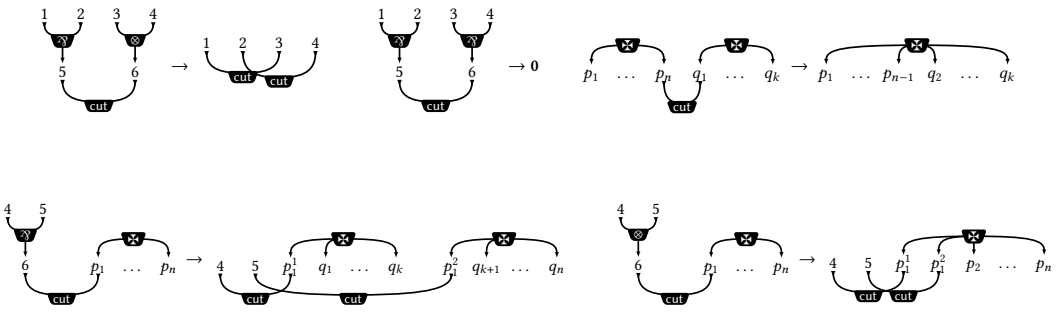

Figure 1: Rules for the homogeneous cut elimination (in the first row) and non homogeneous cut–elimination (in the second row). The non–homogeneous cut elimination of a daimon against a $\parr$–link is non deterministic, $\{q_1, \dots, q_k\}, \{q_{k+1}, \dots, q_n\}$ is a partition of $\{p_2, \dots, p_n\}$.

Notation 10. We write $u \cdot v$ the concatenation of sequences. Given $u = (u_1, \dots, u_n)$ a sequence of element in a set X and an integer $i \in \{1, \dots, n\}$, we denote by $u_{<i}$ (resp. $u_{>i}$) the sequence $(u_1, \dots, u_{i-1})$ (resp. $(u_{i+1}, \dots, u_n)$). Moreover, we write $u[i \leftarrow \epsilon]$ the sequence $u_{<i} \cdot u_{>i}$. Note this induces a reindexing of elements that will need to be handled with care.

Definition 11 (homogeneous cut elimination)**.** The relation of *homogeneous cut elimination* on multiplicative nets is the rewriting relation defined as the contextual closure of the following:

$$\langle \rhd_{\maltese} \overline{p} \rangle + \langle \rhd_{\maltese} \overline{q} \rangle + \langle p_i, q_j \rhd \rangle \to \langle \rhd_{\maltese} \overline{p}[i \leftarrow \epsilon] \cdot \overline{q}[j \leftarrow \epsilon] \rangle$$
$$\langle p_1, p_2 \rhd_{\otimes} p \rangle + \langle q_1, q_2 \rhd_{\parr} q \rangle + \langle p, q \rhd \rangle \to \langle p_1, q_1 \rhd \rangle + \langle p_2, q_2 \rhd \rangle$$

Definition 12 (Non–homogeneous cut elimination)**.** The non homogeneous reduction is denoted $\to_{nh}$, and defined as the extension of the homogeneous cut-elimination with the following rules:

- A redex $\langle \rhd_{\maltese} \overline{p} \rangle + \langle p_i, q \rhd_{\mathrm{cut}} \rangle + \langle q_1, q_2 \rhd_{\parr} q \rangle$ reduces to $\langle \rhd_{\maltese} \overline{u}, r_1 \rangle + \langle \rhd_{\maltese} \overline{v}, r_2 \rangle + \langle r_1, q_1 \rhd_{\mathrm{cut}} \rangle + \langle r_2, q_2 \rhd_{\mathrm{cut}} \rangle$, where $\overline{u}, \overline{v}$ are such that $\overline{u} \cdot \overline{v}$ is a reordering of $\overline{p}[i \leftarrow \epsilon]$, and r_1, r_2 are fresh positions.
- A redex $\langle \rhd_{\maltese} \overline{p} \rangle + \langle p_i, q \rhd_{\mathrm{cut}} \rangle + \langle p_1, p_2 \rhd_{\otimes} p \rangle$ reduces to $\langle \rhd_{\maltese} u \rangle + \langle r_1, q_1 \rhd_{\mathrm{cut}} \rangle + \langle r_2, q \rhd_{\mathrm{cut}} \rangle$, where r_1, r_2 are fresh positions and $u = \overline{p}_{<i} \cdot r_1 \cdot r_2 \cdot \overline{p}_{>i}$.

Remark 13. The non homogeneous cut elimination is not confluent. In particular the reduction of irreversible cuts may make normal forms of the original net unreachable.

Notation 14. Given a net S we use $\mathrm{out}(S)$ to denote the conclusions of S.

Nets can interact in a natural manner: by placing cuts between their outputs.

Definition 15 (Interface)**.** The *interface* of two nets S and R is an injective and functional relation on $\mathrm{out}(S) \times \mathrm{out}(R)$. An interface is *total* whenever it is defined for all conclusions of S.

An interface between two nets $(S, p_1 < \dots < p_n)$ and $(R, q_1 < \dots < q_k)$. is *regular* whenever it contains only pair of the form (p_i, q_i). The *identity interface* is the relation, denoted $\mathtt{id}$, defined when $n = k$ by $\mathtt{id} = \{(p_i, q_i) \mid 1 \le i \le n\}$. The *projective interface* of size $l \le n$ is the regular interface between S and R that is defined exactly for the elements of $\{p_1, \dots, p_l\}$ we denote it $\mathtt{pr}^n_l$.

Definition 16 (Interaction of two nets)**.** Let S, T be two multiplicative modules and σ and interface of S and T. The *interaction* of the nets S and T along the interface σ, is denoted $S ::_\sigma T$ and corresponds to

$$S ::_\sigma T \quad \triangleq \quad S + T + \sum_{(p,q) \in \sigma} \langle p, q \rhd_{\mathrm{cut}} \rangle.$$

Definition 17 (Orthogonality)**.** Two nets S_1 and S_2 are *orthogonal* if there exists an interface σ such that $S_1 ::_\sigma S_2 \to^* \langle \rhd \rangle$. In that case we write $S_1 \perp S_2$.

Notation 18. For the purpose of readability given two nets S and T with respective conclusions $p_1, \dots, p_n$ and $q_1, \dots, q_k$, say $m = min(n, k)$, we will denote the interaction $S ::_{\mathtt{pr}_m} T$ by $S :: T$, whenever there is no ambiguity. For instance if S and T have the same number of conclusion, $S :: T$ denotes $S ::_{\mathtt{id}} T$.

Definition 19 (Types)**.** The orthogonal $A^\perp$ of a set of multiplicative nets A is defined by $\{P \mid \forall a \in A, P \perp a\}$. A *type* $\mathbf{A}$ is a set of multiplicative nets such that $\mathbf{A}^{\perp\perp} = \mathbf{A}$, or equivalently such that $\mathbf{A} = B^\perp$ for some set B.

[2] A type $\mathbf{A}$ is finitely testable if there exists a finite set B such that $\mathbf{A} = B^\perp$.

Remark 20. The nets in a type $\mathbf{A}$ need to all have the same number of conclusion.

Notation 21. Given a net S we let $\mathrm{Pos}(S)$ denote its set of positions.

Definition 22 (Constructions on types)**.** The *parallel sum* of two types $\mathbf{A}$ and $\mathbf{B}$ is defined as

$$\mathbf{A} \parallel \mathbf{B} = \{a + b \mid a \in \mathbf{A}, b \in \mathbf{B}, \mathrm{Pos}(a) \cap \mathrm{Pos}(b) = \emptyset\}^{\perp\perp}.$$

The *functional composition* of two types $\mathbf{A}$ and $\mathbf{B}$, is defined as:

$$\mathbf{A} \cdot \mathbf{B} = \{S \mid \text{for any } a \in \mathbf{A}^{\perp}, S :: a \in \mathbf{B}\}^{\perp\perp}$$

Notation 23. Given a net S with its conclusion ordered as $p_1 < \cdots < p_n$ for an integer $1 \leq i \leq n$ we denote $S(i)$ the conclusion p_i of S.

Definition 24 (Sequential construction on types)**.** Given $\mathbf{A}$ and $\mathbf{B}$ two types we define two constructions:

- The tensor product of two types, $\mathbf{A} \otimes \mathbf{B} = \{a + b + \langle a(1), b(1) \rhd_{\otimes} p\rangle \mid \mathrm{Pos}(a) \cap \mathrm{Pos}(b) = \emptyset, a \in \mathbf{A}, b \in \mathbf{B}\}^{\perp\perp}$.
- The $\parr$–product of two types, $\mathbf{A} \parr \mathbf{B} = (\mathbf{A}^{\perp} \otimes \mathbf{B}^{\perp})^{\perp}$.

Definition 25 (Interpretation Basis)**.** An *interpretation basis* $\mathcal{B}$ is a function that associate to each atomic proposition X a type $[\![X]\!]_{\mathcal{B}}$, the *interpretation* of X, such that

- Each net in $[\![X]\!]_{\mathcal{B}}$ has one conclusion.
- For any atomic proposition X we have $[\![X^{\perp}]\!]_{\mathcal{B}} = [\![X]\!]_{\mathcal{B}}^{\perp}$.

Definition 26 (Realizer of a formula)**.** Given an interpretation basis $\mathcal{B}$, the *interpretation* of a formula is lifted from atomic formula's to any formula of MLL by induction;

$$[\![A \otimes B]\!]_{\mathcal{B}} \triangleq [\![A]\!]_{\mathcal{B}} \otimes [\![B]\!]_{\mathcal{B}}.$$
$$[\![A \parr B]\!]_{\mathcal{B}} \triangleq [\![A]\!]_{\mathcal{B}} \parr [\![B]\!]_{\mathcal{B}}.$$

If there is no ambiguity we relax the notation $[\![A]\!]_{\mathcal{B}}$ to $[\![A]\!]$. A *realizer* of a formula A is a net S belonging to $[\![A]\!]$; this is denoted $S \Vdash_{\mathcal{B}} A$. Eventually we might relax the notation to $S \Vdash A$.

A multiplicative net S *realizes* a sequent $A_1, \ldots, A_n$ of MLL formulas whenever S belongs to the type $[\![A_1]\!] \bullet \cdots \bullet [\![A_n]\!]$. In that case we denote $S \Vdash_{\mathcal{B}} A_1, \ldots, A_n$, and the set of realizers is denoted $[\![A_1, \ldots, A_n]\!]_{\mathcal{B}}$.

Notation 27. A well–known inductive process maps a proof π from MLL, $\mathsf{MLL}^{\maltese}$ (or MLL_2) to a class of (second order) multiplicative net $[\![\pi]\!]$ called its *representant*. Whenever a net S represents a proof π of Γ from a proof system $\mathcal{S}$ we denote $S \vdash_{\mathcal{S}} \Gamma$.

Definition 28 (Approximable interpretation basis)**.** An interpretation basis $\mathcal{B}$ is *approximable* whenever for any propositional variable X the interpretation $[\![X]\!]_{\mathcal{B}}$ contains the unary daimon link $\langle \rhd_{\maltese} p\rangle$.

Theorem 29 (Soundness)**.** *Given Γ a sequent of MLL and S a multiplicative net.*

- *For any interpretation basis $\mathcal{B}$; $S \vdash_{\mathsf{MLL}} \Gamma \Rightarrow S \Vdash_{\mathcal{B}} \Gamma$.*
- *For any approximable interpretation basis $\mathcal{B}$; $S \vdash_{\mathsf{MLL}^{\maltese}} \Gamma \Rightarrow S \Vdash_{\mathcal{B}} \Gamma$.*

Notation 30. Given an interpretation basis $\mathcal{B}$ we let $\overline{\mathcal{B}}$ denote the interpretation basis such that for any atomic formula X we have $[\![X]\!]_{\overline{\mathcal{B}}} = [\![X]\!]_{\mathcal{B}}^{\perp}$.

Definition 31. A multiplicative net is *cut–free* when it does not contain any cut–links.

Theorem 32 ($\mathsf{MLL}^{\maltese}$ completeness)**.** *Given some sequent Γ and S a cut–free net and $\mathcal{B}$ an approximable interpretation basis, if S realizes both $[\![\Gamma]\!]_{\mathcal{B}}$ and $[\![\Gamma]\!]_{\overline{\mathcal{B}}}$ then S represents a proof of Γ from $\mathsf{MLL}^{\maltese}$.*

Remark 33. The previous theorems imply that in particular – for cut–free nets – $\bigcap_{\mathcal{B}:approx} [\![\Gamma]\!]_{\mathcal{B}}$ corresponds to the proofs of Γ in $\mathsf{MLL}^{\maltese}$.

Notation 34. Given an interpretation basis $\mathcal{B}$ and a type $\mathbf{A}$ we denote $\mathcal{B}\{X \mapsto \mathbf{A}\}$ the base which maps X to $\mathbf{A}$ and $Y \neq X$ to $[\![Y]\!]_{\mathcal{B}}$.

Definition 35 (Intersection and union type)**.** Let $\mathcal{B}$ be an interpretation basis, and Ω be a set of types with one output. Given a Γ a sequent of MLL formulas and X a propositional variable the *intersection type* on Ω of Γ in X w.r.t. to $\mathcal{B}$ is defined as follow;

$$[\![\bigcap_{X \in \Omega} \Gamma]\!]_{\mathcal{B}} \triangleq \bigcap_{R \in \Omega} [\![\Gamma]\!]_{\mathcal{B}\{X \mapsto R\}}.$$

Furthermore we define the dual construction the *union type* of A over X as its orthogonal,

$$[\![\bigcup_{X \in \Omega} \Gamma]\!]_{\mathcal{B}} \triangleq \left(\bigcup_{R \in \Omega} [\![\Gamma]\!]_{\mathcal{B}\{X \mapsto R\}}\right)^{\perp\perp}.$$

Definition 36. A *cyclic test* is a net $\langle \rhd_{\maltese} p_1, p_2\rangle + \langle p_1, p_2 \rhd_{\otimes} p\rangle$, it is denoted $T(\otimes)$. A *disjoint test* is a net $\langle \rhd_{\maltese} p_1\rangle + \langle \rhd_{\maltese} p_2\rangle + \langle p_1, p_2 \rhd_{⅋} p\rangle$, it is denoted $T(⅋)$.

The *cyclic* resp. *disjoint types* are the types generated by the cyclic (resp. disjoint) tests. They are denoted $\mathsf{T}(\otimes)$ and $\mathsf{T}(⅋)$.

Given an interpretation basis $\mathcal{B}$ and X a propositional variable the set of test for X relatively to $\mathcal{B}$ is denoted $\mathsf{T}_{\mathcal{B}}(X)$ and corresponds to the set made of three types $\{[\![X]\!]_{\mathcal{B}}, \mathsf{T}(\otimes), \mathsf{T}(⅋)\}$.

Definition 37. A multiplicative net is *proof like* whenever its daimon links have exactly two outputs.

Theorem 38 (MLL completeness)**.** *Given S a proof like and cut–free net and $\mathcal{B}$ some approximable interpretation basis. If S belongs to $\bigcap_{X\in\mathcal{V}}[\![\bigcap_{X\in\mathsf{T}_{\mathcal{B}}(X)}\Gamma]\!]_{\mathcal{B}}$ and $\bigcap_{X\in\mathcal{V}}[\![\bigcap_{X\in\mathsf{T}_{\mathcal{B}}(X)}\Gamma]\!]_{\overline{\mathcal{B}}}$ then S is the image of a proof in* MLL*.*

Remark 39. Let Ω be the set of types with one output, $\bigcap_{X\in\mathcal{V}}[\![\bigcap_{X\in\Omega}\Gamma]\!]_{\mathcal{B}}$ is contained in $\bigcap_{X\in\mathcal{V}}[\![\bigcap_{X\in\mathsf{T}_{\mathcal{B}}(X)}\Gamma]\!]_{\mathcal{B}}$. Furthermore, since a cut free element of $\bigcap_{X\in\mathcal{V}}[\![\bigcap_{X\in\mathsf{T}_{\mathcal{B}}(X)}\Gamma]\!]_{\mathcal{B}}$ is a proof (and so cut free nets form a filter[3] of that type) the soundness for MLL proofs ensure that this type belongs to the intersection $\bigcap_{X\in\mathcal{V}}[\![\bigcap_{X\in\Omega}\Gamma]\!]_{\mathcal{B}}$. Hence these two sets are the same.

This is important to ensure the finiteness of the testability of the intersection type on Ω: the finite intersections of finitely testable types remain finitely testable.

Definition 40 (realizers of MLL_2)**.** Let $\mathcal{B}$ be an approximable interpretation basis and Ω denote the set of types with one output. Given a formula A of MLL_2 its set of *realizers* is given by the following induction:

$$[\![A\otimes B]\!] \triangleq [\![A]\!]\otimes[\![B]\!] \qquad [\![\forall X\ A]\!] \triangleq \{S + \langle S(1) \rhd_{\forall} q\rangle \mid S \in [\![\textstyle\bigcap_{X\in\Omega} A]\!]\}$$
$$[\![A ⅋ B]\!] \triangleq [\![A]\!] ⅋ [\![B]\!] \qquad [\![\exists X\ A]\!] \triangleq \{S + \langle S(1) \rhd_{\exists} q\rangle \mid S \in [\![\textstyle\bigcup_{X\in\Omega} A]\!]\}^{\perp\perp}$$

Theorem 41 (Soundness for MLL_2)**.** *Let $\mathcal{B}$ be an approximable interpretation basis. Given S a proof–like multiplicative second order net. If S represents a proof of the sequent Γ then S belongs to $[\![\Gamma]\!]_{\mathcal{B}}$.*

Bibliography

[DR89] Vincent Danos and Laurent Regnier. "The structure of multiplicatives". In: *Archive for Mathematical Logic* 28.3 (Oct. 1989), pp. 181–203. ISSN: 1432-0665. DOI: 10.1007/BF01622878. URL: https://doi.org/10.1007/BF01622878.

[Gir11] Jean-Yves Girard. "Geometry of Interaction V: Logic in the hyperfinite factor". In: *Theor. Comput. Sci.* 412.20 (2011), pp. 1860–1883. DOI: 10.1016/j.tcs.2010.12.016. URL: https://doi.org/10.1016/j.tcs.2010.12.016.

[Gir87] Jean-Yves Girard. "Multiplicatives". In: *Logic and Computer Science: New Trends and Applications*. Ed. by G. Lolli. Rosenberg & Sellier, 1987, pp. 11–34.

[Gir88] Jean-Yves Girard. "Geometry of interaction 2: deadlock-free algorithms". In: *COLOG-88, International Conference on Computer Logic, Tallinn, USSR, December 1988, Proceedings*. Ed. by Per Martin-Löf and Grigori Mints. Vol. 417. Lecture Notes in Computer Science. Springer, 1988, pp. 76–93. DOI: 10.1007/3-540-52335-9_49. URL: https://doi.org/10.1007/3-540-52335-9%5C_49.

[Gir89] Jean-Yves Girard. "Geometry of Interaction 1: Interpretation of System F". In: *Logic Colloquium '88*. Ed. by R. Ferro et al. Vol. 127. Studies in Logic and the Foundations of Mathematics. Elsevier, 1989, pp. 221–260. DOI: https://doi.org/10.1016/S0049-237X(08)70271-4. URL: https://www.sciencedirect.com/science/article/pii/S0049237X08702714.

[Gir95] J.-Y. Girard. "Geometry of interaction III: accommodating the additives". In: *Advances in Linear Logic*. Ed. by Jean-Yves Girard, Yves Lafont, and LaurentEditors Regnier. London Mathematical Society Lecture Note Series. Cambridge University Press, 1995, pp. 329–389. DOI: 10.1017/CBO9780511629150.017.

[Kle45] S. C. Kleene. "On the interpretation of intuitionistic number theory". In: *The Journal of Symbolic Logic* 10.4 (1945), pp. 109–124. DOI: 10.2307/2269016.

[Miq09] Alexandre Miquel. "De la formalisation des preuves à l'extraction de programmes". In: *Habilitation at Université Paris Diderot* (2009). URL: https://www.fing.edu.uy/~amiquel/publis/hdr.pdf.

[NPS16] Alberto Naibo, Mattia Petrolo, and Thomas Seiller. "On the Computational Meaning of Axioms". In: *Epistemology, Knowledge and the Impact of Interaction*. Ed. by Juan Redmond, Olga Pombo Martins, and Ángel Nepomuceno Fernández. Cham: Springer International Publishing, 2016, pp. 141–184. ISBN: 978-3-319-26506-3. DOI: 10.1007/978-3-319-26506-3_5. URL: https://doi.org/10.1007/978-3-319-26506-3_5.

[Sei12] Thomas Seiller. "Interaction Graphs: Multiplicatives". In: *Annals of Pure and Applied Logic* 163.12 (2012), pp. 1808–1837. DOI: 10.1016/j.apal.2012.04.005.

[Sei13] Thomas Seiller. "Interaction Graphs: Exponentials". In: *Log. Methods Comput. Sci.* 15 (2013).

[Sei15] Thomas Seiller. "Interaction Graphs: Full Linear Logic". In: *CoRR* abs/1504.04152 (2015). arXiv: 1504.04152. URL: http://arxiv.org/abs/1504.04152.

[Sei16] Thomas Seiller. "Interaction Graphs: Additives". In: *Annals of Pure and Applied Logic* 167.2 (2016), pp. 95–154. DOI: 10.1016/j.apal.2015.10.001.

[Sei17] Thomas Seiller. "Interaction Graphs: Graphings". In: *Annals of Pure and Applied Logic* 168.2 (2017), pp. 278–320. DOI: 10.1016/j.apal.2016.10.007.

[3] F is a filter of the type $\mathbf{A}$ if any net in $\mathbf{A}$ has a redex in F.