Craig Interpolation for Semi-Substructural Logics

Niccolò Veltri and Cheng-Syuan Wan

Department of Software Science, Tallinn University of Technology, Estonia.

Contributing authors: niccolo@cs.ioc.ee; cswan@cs.ioc.ee;

Abstract

This work studies Craig interpolation for the logic `SkNMILL`, a substructural logic admitting only a directed notion of associativity and unitality. In this setting, Craig interpolation cannot be proved by directly employing standard proof-theoretic methods, such as Maehara's method, a situation that `SkNMILL` shares with other logical systems such as the product-free Lambek calculus and the implicational fragment of intuitionistic logic. We show how to overcome this issue and appropriately modify Maehara's method for recovering Craig interpolation. We take one step further and, following the category-theoretic perspective of Čubrić, we produce a proof-relevant version of the interpolation theorem, in which we show that our interpolation procedures are right inverses of the admissible cut rules.

Keywords: Craig interpolation, semi-substructural logic, Maehara's method, proof-relevant interpolation

1 Introduction

Craig interpolation is a fundamental result in first-order logic, named after the logician William Craig [1]. A logic $\mathcal{L}$ has the *Craig interpolation property* if, for any formula $A \to C$ provable in $\mathcal{L}$ (where $\to$ is the implication connective in $\mathcal{L}$), there exists a formula D such that $A \to D$ and $D \to C$ are provable in $\mathcal{L}$, satisfying the variable condition: $\mathsf{var}(D) \subseteq \mathsf{var}(A) \cap \mathsf{var}(C)$, where $\mathsf{var}(A)$ is the set of atomic formulae appearing in A. Craig interpolation has been mostly employed to prove model-theoretical results, including Beth's definability theorem [2], but more recently it has found applications in other areas, e.g. in model checking [3].

From the viewpoint of sequent calculus, substructural logics are defined by the absence of at least one structural rule. A notable instance is Joachim Lambek's syntactic calculus [4], which forbids weakening, contraction and exchange. Its non-associative variant, which has been extensively studied [5], also disallows associativity. Another significant example is linear logic, introduced by Jean-Yves Girard [6], where weakening and contraction are disallowed but can be recovered for specific formulas through modalities. Substructural logics have proven useful for modelling various phenomena in different research areas, from the computational analysis of natural language syntax to the development of programming languages sensitive to resource management.

Craig interpolation for substructural logics has been extensively studied, using either algebraic or proof-theoretic techniques.

For substructural logics that lack a cut-free sequent calculus, such as arbitrary extensions of the full Lambek calculus with exchange ($\mathbf{FL_e}$), Craig interpolation is established using algebraic methods such as amalgamation. For further details on this approach, see [7].

For substructural logics that admit a cut-free sequent calculus, Craig interpolation is typically proven by adapting Maehara's method [8], which originally aimed to prove interpolation for **LK**, a sequent calculus for classical logic. This includes the full Lambek calculus (**FL**) and its extensions that incorporate various combinations of weakening, exchange, and contraction. In the case of **FL**, for instance, the proof starts by establishing a stronger form of interpolation which we call *Maehara interpolation property* (MIP) [9]. The latter property states:

(MIP for **FL**) Given $f : \Gamma \vdash C$ and a partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$ of Γ, there exist a formula D and two derivations $g : \Gamma_1 \vdash D$ and $h : \Gamma_0, D, \Gamma_2 \vdash C$, and $\mathsf{var}(D) \subseteq \mathsf{var}(\Gamma_0) \cap \mathsf{var}(\Gamma_0, \Gamma_1, C)$

Being a partition simply means that the ordered list of formulae Γ is equal to the concatenation of Γ_0, Γ_1 and Γ_2, i.e. $\Gamma = \Gamma_0, \Gamma_1, \Gamma_2$. Maehara interpolation also holds for the non-associative variant of **FL**, for an appropriate reformulation of the principle in which antecedents are trees of formulae instead of lists. Notice that Craig interpolation is a property of a logic (with a notion of implication), while Maehara interpolation is a property of a deductive system in which it is possible to appropriately partition antecedents.

Maehara interpolation is a stronger form of the so-called *deductive interpolation property*. A logic $\mathcal{L}$ has the deductive interpolation property if, for any formulae A and C, whenever $A \vdash C$ (where $\vdash$ is the consequence relation of $\mathcal{L}$), then there exists a formula B such that $A \vdash B$ and $B \vdash C$ while also satisfying the usual variable condition. Furthermore, if the sequent calculus of $\mathcal{L}$ admits the invertibility of implication-right rules (as is the case in **FL** for both left and right implication), Craig interpolation follows immediately as a consequence of deductive interpolation.

While Maehara's method is often applicable to extensions of **FL**, it does not work for some of its fragment, which therefore do not enjoy Maehara interpolation. This is the case for fragments lacking multiplicative and/or additive conjunction, such as the product-free Lambek calculus [10] (with only left and right implications as connectives) and the implicational fragment of intuitionistic logic [11]. The variant of Maehara interpolation satisfied by the product-free Lambek calculus, which we dub *Maehara* *multi-interpolation* (MMIP), is particularly relevant for our work. Here is its statement, which we have slightly modified to better align with our forthcoming discussion:

(MMIP for product-free Lambek calculus) Given $f : \Gamma \vdash C$ and a partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$ of Γ, there exist

- a partition $\langle \Delta_1, \dots, \Delta_n \rangle$ of Γ_1,
- a list of interpolant formulae $D_1, \dots, D_n$,
- a derivation $g : \Gamma_0, D_1, \dots, D_n, \Gamma_2 \vdash C$,
- a derivation $h_i : \Delta_i \vdash D_i$ for all $i \in [1, \dots, n]$, such that
- $\mathsf{var}(D_1, \dots, D_n) \subseteq \mathsf{var}(\Delta_1, \dots, \Delta_n) \cap \mathsf{var}(\Gamma_0, \Gamma_2, C)$.

Differently from Maehara interpolation, in the above property we look for a list of interpolants instead of a single formula. This adjustment allows to overcome the difficulty caused by the absence of conjunction.

In this paper, we aim at proving Craig interpolation for the *semi-substructural* logic `SkNMILL` which we recently introduced in collaboration with Tarmo Uustalu [12]. In our terminology, a logic is semi-substructural if it is an intermediate logic in between (certain fragments of) non-associative and associative intuitionistic linear logic (or the Lambek calculus). Semi-associativity and semi-unitality are encoded as follows. Sequents are in the form $S \mid \Gamma \vdash A$, where the antecedent consists of an optional formula S, called stoup, adapted from Girard [13], and an ordered list of formulae Γ. The succedent is a single formula A. We restrict the application of introduction rules in an appropriate way to allow only one of the directions of associativity and unitality, e.g. only $(A \otimes B) \otimes C \mid \ \vdash A \otimes (B \otimes C)$ is provable in `SkNMILL`, while its inverse $A \otimes (B \otimes C) \mid \ \vdash (A \otimes B) \otimes C$ is not. In other words, only directed variants of the structural rules of associativity and unitality are included, while their inverses are generally disallowed.

The introduction of semi-substructural logics was originally motivated by the study of combinatorial properties of certain categorical structures, called *left skew monoidal categories* [14]. These categories are a weaker variant of MacLane's monoidal categories. In left skew monoidal categories, the structural morphisms of associativity and unitality (which are natural transformations typically called 'associator' and 'unitors') are not required to have an inverse. Instead, they are natural family of morphisms with a specific orientation. For this reason, left skew monoidal categories can be seen as *semi-associative* and *semi-unital* variants of monoidal categories.

Different variants of left skew monoidal categories have led to the development of their corresponding semi-substructural logic. These include (*i*) left skew semigroup [15], (*ii*) left skew monoidal [16], (*iii*) left skew (prounital) closed [17], (*iv*) left skew monoidal closed categories [12, 18, 19], and (*v*) left distributive skew monoidal categories with finite products and coproducts [20]. Each of these logics admits a cut-free sequent calculus. Moreover, they admit a subcalculus of "proofs in normal form", which is inspired by Jean-Marc Andreoli's focusing method [21] and as such provides a way to make root-first proof search more deterministic. Practically, the focusing method is employed for solving the coherence problem for the corresponding variants of left skew monoidal categories. In the case of left skew monoidal closed categories, a solution to the coherence (or word) problem consists of a procedure for deciding equality of parallel morphisms in the free left skew monoidal closed category on a given set At.

In previous work, we showed that the focused subcalculus for `SkNMILL` is a concrete presentation of such free category, so a solution to the coherence problem is obtained by checking whether two morphisms are represented by the same derivation in the focused subcalculus.

To prove Craig interpolation for `SkNMILL`, we need to modify the statement of Maehara interpolation. This modification is required due to issues similar to those encountered in the product-free Lambek calculus [10] and the implicational fragment of intuitionistic logic [11], where Maehara interpolation fails. The main result of the paper is the following:

Theorem 4. *In the sequent calculus for* `SkNMILL`*, the following two interpolation properties hold:*

(sMIP) *Given a derivation $f : S \mid \Gamma \vdash C$ and a partition $\langle \Gamma_0, \Gamma_1 \rangle$ of Γ, there exist*
- *an interpolant formula D,*
- *a derivation $g : S \mid \Gamma_0 \vdash D$,*
- *a derivation $h : D \mid \Gamma_1 \vdash C$, such that*
- $\mathsf{var}(D) \subseteq \mathsf{var}(S, \Gamma_0) \cap \mathsf{var}(\Gamma_1, C)$.

(cMMIP) *Given a derivation $f : S \mid \Gamma \vdash C$ and a partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$ of Γ, there exist*
- *a partition $\langle \Delta_1, \ldots, \Delta_n \rangle$ of Γ_1,*
- *a list of interpolant formulae $D_1, \ldots, D_n$,*
- *$g : S \mid \Gamma_0, D_1, \ldots, D_n, \Gamma_2 \vdash C$,*
- *$h_i : - \mid \Delta_i \vdash D_i$ for all $i \in [1, \ldots, n]$, such that*
- $\mathsf{var}(D_1, \ldots, D_n) \subseteq \mathsf{var}(\Delta_1, \ldots, \Delta_n) \cap \mathsf{var}(S, \Gamma_0, \Gamma_2, C)$.

In `SkNMILL`, the first property sMIP, which stands for *stoup Maehara interpolation*, resembles Maehara interpolation for the **FL**. Whereas cMMIP, which stands for *context Maehara multi-interpolation*, is similar to Maehara multi-interpolation for the product-free Lambek calculus.

Motivated by the categorical interpretation of `SkNMILL`, we take one more step and investigate the interplay between the admissible cut rules (called scut and ccut) and the derivations produced by the interpolation algorithm of Theorem 4. In previous work [12], we introduced an equivalence relation on derivations ($\stackrel{\circ}{=}$) that captures *eta*-conversions and permutative conversions, and is both sound and complete with respect to the categorical semantics. We show that the sMIP and cMMIP procedures of Theorem 4 are right inverses of the admissible rules scut and ccut, respectively. Formally, we prove the following theorem:

Theorem 7.

(i) Let $g : S \mid \Gamma_0 \vdash D$ and $h : D \mid \Gamma_1 \vdash C$ be the derivations obtained by applying the sMIP procedure on a derivation $f : S \mid \Gamma \vdash C$ with the partition $\langle \Gamma_0, \Gamma_1 \rangle$. Then $\mathsf{scut}(g, h) \stackrel{\circ}{=} f$.

(ii) Let $g : S \mid \Gamma_0, D_1, \ldots, D_n, \Gamma_2 \vdash C$ and $h_i : - \mid \Delta_i \vdash D_i$ for $i \in [1, \ldots, n]$ be derivations obtained by applying the cMMIP procedure on a derivation $f : S \mid \Gamma \vdash C$ with the partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$. Then $\mathsf{ccut}^*([h_i], g) \stackrel{\circ}{=} f$.

In the above statement, ccut^* denotes multiple applications of the admissible ccut rule, one for each derivation h_i. Theorems 4 and 7 together show that `SkNMILL` satisfies a *proof-relevant* form of Craig interpolation, in the sense formulated in the early 90s by Čubrić [22] in the setting of intuitionistic propositional logic and recently discussed also by Saurin [23] for (extensions of) classical linear logic.

All the proofs presented in this paper have been formalized in the proof assistant Agda. The full formalization is freely available at the following website:

https://github.com/niccoloveltri/code-skewmonclosed/tree/interpolation.

2 A Sequent Calculus for SkNMILL

We start by recalling the sequent calculus for left skew monoidal closed categories that was introduced in [12], which we name `SkNMILL`. This is a "skew variant" of non-commutative multiplicative intuitionistic linear logic, in a way that will be made precise later in this section.

Formulae are inductively generated by the grammar $A, B ::= X \mid \mathsf{I} \mid A \otimes B \mid A \multimap B$, where X comes from a set At of atoms, I is a multiplicative unit, $\otimes$ is multiplicative conjunction and $\multimap$ is a linear implication. The set of formulae is denoted Fma.

A sequent is a triple of the form $S \mid \Gamma \vdash A$. The antecedent consists of two parts: an optional formula S, called the *stoup* (a terminology that comes from Girard [13]), and an ordered list of formulae Γ, that we call the *context*. The succedent A is a single formula. The symbol S consistently denotes a stoup, meaning S can either be a single formula or empty, indicated as $S = -$. Furthermore, letters X, Y, Z and W always denote atomic formulae.

Derivations are generated recursively by the following rules:

$$\begin{array}{cc}
\dfrac{}{A \mid \quad \vdash A}\;\mathsf{ax} & \dfrac{A \mid \Gamma \vdash C}{- \mid A, \Gamma \vdash C}\;\mathsf{pass} \\[2ex]
\dfrac{- \mid \Gamma \vdash A \quad B \mid \Delta \vdash C}{A \multimap B \mid \Gamma, \Delta \vdash C}\;{\multimap}\mathsf{L} & \dfrac{S \mid \Gamma, A \vdash B}{S \mid \Gamma \vdash A \multimap B}\;{\multimap}\mathsf{R} \\[2ex]
\dfrac{- \mid \Gamma \vdash C}{\mathsf{I} \mid \Gamma \vdash C}\;\mathsf{IL} & \dfrac{}{- \mid \quad \vdash \mathsf{I}}\;\mathsf{IR} \\[2ex]
\dfrac{A \mid B, \Gamma \vdash C}{A \otimes B \mid \Gamma \vdash C}\;{\otimes}\mathsf{L} & \dfrac{S \mid \Gamma \vdash A \quad - \mid \Delta \vdash B}{S \mid \Gamma, \Delta \vdash A \otimes B}\;{\otimes}\mathsf{R}
\end{array} \tag{1}$$

The inference rules in (1) are similar to the ones in the sequent calculus for non-commutative multiplicative intuitionistic linear logic (`NMILL`) [24], but with some crucial differences:

1. The left logical rules IL, $\otimes\mathsf{L}$ and $\multimap\mathsf{L}$, read bottom-up, are only allowed to be applied on the formula in the stoup position.
2. The right tensor rule $\otimes\mathsf{R}$, read bottom-up, splits the antecedent of a sequent $S \mid \Gamma, \Delta \vdash A \otimes B$ and in the case where S is a formula, S is always moved to the stoup of the left premise, even if Γ is empty.
3. The presence of the stoup distinguishes two types of antecedents, $A \mid \Gamma$ and $- \mid A, \Gamma$. The structural rule pass (for 'passivation'), read bottom-up, allows the

moving of the leftmost formula in the context to the stoup position whenever the stoup is empty.

4. The logical connectives of `NMILL` typically include two ordered implications $\multimap$ and $\multimapinv$ (often also called residuals), which are two variants of linear implication arising from the removal of the exchange rule from intuitionistic linear logic. In our logic `SkNMILL`, only the implication $\multimap$ is present, which Lambek would call right implication/residual.

`SkNMILL` can be seen as an intermediate logic between: (i) the $(\mathsf{I}, \otimes, \multimap)$-fragment of non-commutative intuitionistic linear logic (where $\multimap$ is right residual $/$), i.e. the associative Lambek calculus with multiplicative unit, and (ii) the same fragment without I and $\otimes$, i.e. the product-free Lambek calculus[1] without left residual. In fact, every derivation in `SkNMILL` can be replicated in the Lambek calculus. On the other hand, the product-free Lambek calculus without left residual admits a presentation as a semi-substructural logic, corresponding to the fragment of (1) with only rules ax, pass, $\multimap\mathsf{L}$ and $\multimap\mathsf{R}$ [17].

This calculus is cut-free, in the sense that the following two rules are admissible:

$$\frac{S \mid \Gamma \vdash A \quad A \mid \Delta \vdash C}{S \mid \Gamma, \Delta \vdash C}\,\mathsf{scut} \qquad \frac{- \mid \Gamma \vdash A \quad S \mid \Delta_0, A, \Delta_1 \vdash C}{S \mid \Delta_0, \Gamma, \Delta_1 \vdash C}\,\mathsf{ccut} \tag{2}$$

The presence of two cut rules comes from the fact that the cut formula can appear either in the stoup or the context of the second premise.

We introduce a few admissible rules that will be employed later in the paper. First, given a list of formulae $\Delta = A_1, \dots, A_n$, we define an iterated version of the rule $\multimap\mathsf{R}$, consisting of n applications of $\multimap\mathsf{R}$. Below and in the future we write $\Delta \multimap^* B$ for the formula $A_1 \multimap (A_2 \multimap (\dots (A_n \multimap B) \dots))$, which is simply B when Δ is empty. The double-line inference rule denotes an equality of sequents.

$$\frac{\overset{f}{S \mid \Gamma, \Delta \vdash B}}{S \mid \Gamma \vdash \Delta \multimap^* B}\,\multimap\mathsf{R}^* \quad = \quad \begin{array}{c} f \\ S \mid \Gamma, \Delta \vdash B \\ \hline\hline S \mid \Gamma, A_1, A_2, \dots, A_n \vdash B \\ \hline S \mid \Gamma, A_1, A_2, \dots, A_{n-1} \vdash A_n \multimap B \quad \multimap\mathsf{R} \\ \vdots \\ S \mid \Gamma, A_1 \vdash A_2 \multimap (\dots (A_n \multimap B) \dots) \\ \hline S \mid \Gamma \vdash A_1 \multimap (A_2 \multimap (\dots (A_n \multimap B) \dots)) \quad \multimap\mathsf{R} \\ \hline\hline S \mid \Gamma \vdash \Delta \multimap^* B \end{array} \tag{3}$$

If $n = 0$, then $\multimap\mathsf{R}^* f = f$.

Second, given a list of formulae $\Delta = A_1, \dots, A_n$ and a list of derivations $f_i : - \mid \Gamma_i \vdash A_i$ for $i \in [1, \dots, n]$, we define an iterated version of $\multimap\mathsf{L}$, consisting on n

[1] This deductive system traditionally requires antecedents to be non-empty lists of formulae, which is a restriction motivated by linguistic consideration. Here we mean the version of this calculus without such restriction.

applications of $\multimap\mathsf{L}$, one for each derivation f_i:

$$\frac{\overset{[f_i]}{[- \mid \Gamma_i \vdash A_i]_i} \quad \overset{g}{B \mid \Lambda \vdash C}}{\Delta \multimap^* B \mid \Gamma_1, \dots, \Gamma_n, \Lambda \vdash C}\ \multimap\mathsf{L}^*$$
$$= \frac{\dfrac{\overset{f_1}{- \mid \Gamma_1 \vdash A_1} \quad \overset{\overset{\dfrac{\overset{f_n}{- \mid \Gamma_n \vdash A_n} \quad \overset{g}{B \mid \Lambda \vdash C}}{A_n \multimap B \mid \Gamma_n, \Lambda \vdash C}\ \multimap\mathsf{L}}{\vdots}}{A_2 \multimap (\dots(A_n \multimap B)\dots) \mid \Gamma_2, \dots, \Gamma_n, \Lambda \vdash C}}{A_1 \multimap (A_2 \multimap (\dots(A_n \multimap B)\dots)) \mid \Gamma_1, \Gamma_2, \dots, \Gamma_n, \Lambda \vdash C}\ \multimap\mathsf{L}}{\Delta \multimap^* B \mid \Gamma_1, \dots, \Gamma_n, \Lambda \vdash C} \tag{4}$$

The rule $\multimap\mathsf{L}^*$ has $n+1$ premises, the first n are collected in the list of sequents $[- \mid \Gamma_i \vdash A_i]_i$. If $n = 0$, then $\multimap\mathsf{L}^*([\,], g) = g$.

Finally, given a list of derivations $f_i : - \mid \Delta_i \vdash A_i$ for $i \in [1, \dots, n]$, we define an iterated version of ccut, consisting on n applications of ccut, one for each derivation f_i:

$$\frac{\overset{[f_i]}{[- \mid \Delta_i \vdash A_i]_i} \quad \overset{g}{S \mid \Gamma_0, A_1, \dots, A_n, \Gamma_1 \vdash C}}{S \mid \Gamma_0, \Delta_1, \Delta_2, \dots, \Delta_n, \Gamma_1 \vdash C}\ \mathsf{ccut}^*$$
$$= \frac{\overset{f_1}{- \mid \Delta_1 \vdash A_1} \qquad \overset{\overset{\dfrac{\overset{f_n}{- \mid \Delta_n \vdash A_n} \quad \overset{g}{S \mid \Gamma_0, A_1, A_2, \dots, A_n, \Gamma_1 \vdash C}}{S \mid \Gamma_0, A_1, A_2, \dots, \Delta_n, \Gamma_1 \vdash C}\ \mathsf{ccut}}{\vdots}}{S \mid \Gamma_0, A_1, \Delta_2, \dots, \Delta_n, \Gamma_1 \vdash C}}{S \mid \Gamma_0, \Delta_1, \Delta_2, \dots, \Delta_n, \Gamma_1 \vdash C}\ \mathsf{ccut} \tag{5}$$

If $n = 0$, then $\mathsf{ccut}^*([\,], g) = g$.

3 Equivalence of Derivations

Sets of derivations are quotiented by a congruence relation $\overset{\circ}{=}$, generated by the pairs of derivations in Figure 1 and 2. The three equations in Figure 1 are η-conversions, completely characterizing the ax rule on non-atomic formulae. The equations in Figure 2 are permutative conversions.

The generating equations of $\overset{\circ}{=}$ have been carefully selected to appropriately match the equational theory of left skew monoidal closed categories. More information about this relationship in terms of categorical semantics can be found in [12], where a precise correspondence between sequent calculus derivations, the congruence relation $\overset{\circ}{=}$ and left skew monoidal closed categories is described. The latter paper also contains an interpretation of the sequent calculus as a logic of resources, as well as a calculus of derivations in normal form, which completely characterizes proofs modulo the congruence relation $\overset{\circ}{=}$.

$$\dfrac{}{\mathsf{I} \mid\ \vdash \mathsf{I}}\,\mathsf{ax} \quad \doteq \quad \dfrac{\dfrac{}{- \mid\ \vdash \mathsf{I}}\,\mathsf{IR}}{\mathsf{I} \mid\ \vdash \mathsf{I}}\,\mathsf{IL}$$

$$\dfrac{}{A\otimes B \mid\ \vdash A\otimes B}\,\mathsf{ax} \quad \doteq \quad \dfrac{\dfrac{\dfrac{}{A\mid\ \vdash A}\,\mathsf{ax} \quad \dfrac{\dfrac{}{B\mid\ \vdash B}\,\mathsf{ax}}{-\mid B\vdash B}\,\mathsf{pass}}{A \mid B \vdash A\otimes B}\,\otimes\mathsf{R}}{A\otimes B\mid\ \vdash A\otimes B}\,\otimes\mathsf{L}$$

$$\dfrac{}{A\multimap B\mid\ \vdash A\multimap B}\,\mathsf{ax} \quad \doteq \quad \dfrac{\dfrac{\dfrac{\dfrac{}{A\mid\ \vdash A}\,\mathsf{ax}}{-\mid A\vdash A}\,\mathsf{pass}\quad \dfrac{}{B\mid\ \vdash B}\,\mathsf{ax}}{A\multimap B\mid A\vdash B}\,\multimap\mathsf{L}}{A\multimap B\mid\ \vdash A\multimap B}\,\multimap\mathsf{R}$$

Fig. 1 Equivalence of derivations: η-conversions

Moreover, more equations of derivations hold in `SkNMILL` due to the cut-elimination procedures defined in [12, 19]. This set of equations fully describe the possible interactions between cut rules. The first set of equations in Figure 3 shows that parallel composition of cut rules is commutative. The second set of equations in Figure 4 shows that sequential composition of cut rules is associative. Analogous equations have been proved in [16] for the fragment of `SkNMILL` without linear implication. Notice that the each pair of derivations in these equations are *strictly* equal, not merely $\doteq$-related.

Proposition 1. *The commutativity equations in Figure 3 and the associativity equations in Figure 4 are admissible.*

Proof. The proof proceeds by mutual induction on the structure of derivations. There are many cases to consider. We do not include the long proof here and refer the interested reader to consult our Agda formalization. Heavy proofs by pattern matching like this one is where the employment of a proof assistant becomes very helpful, in our experience. □

We conclude this section by introducing a final equation and two equivalences, that will be employed later in Section 6. In the construction of the scut admissibility procedure [12, 19], the case when the first premise is of the form $\multimap\mathsf{R}\ f$ and the second premise of the form $\multimap\mathsf{L}\ (g, h)$ (i.e. a principal cut when the cut formula is an

$$\dfrac{\dfrac{\overset{f}{A' \mid \Gamma \vdash A}}{- \mid A',\Gamma \vdash A}\,\mathsf{pass} \quad \overset{g}{- \mid \Delta \vdash B}}{- \mid A',\Gamma,\Delta \vdash A\otimes B}\,\otimes\mathsf{R} \;\circeq\; \dfrac{\dfrac{\overset{f}{A' \mid \Gamma \vdash A} \quad \overset{g}{- \mid \Delta \vdash B}}{A' \mid \Gamma,\Delta \vdash A\otimes B}\,\otimes\mathsf{R}}{- \mid A',\Gamma,\Delta \vdash A\otimes B}\,\mathsf{pass}$$

$$\dfrac{\dfrac{\overset{f}{- \mid \Gamma \vdash A}}{\mathsf{I} \mid \Gamma \vdash A}\,\mathsf{IL} \quad \overset{g}{- \mid \Delta \vdash B}}{\mathsf{I} \mid \Gamma,\Delta \vdash A\otimes B}\,\otimes\mathsf{R} \;\circeq\; \dfrac{\dfrac{\overset{f}{- \mid \Gamma \vdash A} \quad \overset{g}{- \mid \Delta \vdash B}}{- \mid \Gamma,\Delta \vdash A\otimes B}\,\otimes\mathsf{R}}{\mathsf{I} \mid \Gamma,\Delta \vdash A\otimes B}\,\mathsf{IL}$$

$$\dfrac{\dfrac{\overset{f}{A' \mid B',\Gamma \vdash A}}{A'\otimes B' \mid \Gamma \vdash A}\,\otimes\mathsf{L} \quad \overset{g}{- \mid \Delta \vdash B}}{A'\otimes B' \mid \Gamma,\Delta \vdash A\otimes B}\,\otimes\mathsf{R} \;\circeq\; \dfrac{\dfrac{\overset{f}{A' \mid B',\Gamma \vdash A} \quad \overset{g}{- \mid \Delta \vdash B}}{A' \mid B',\Gamma,\Delta \vdash A\otimes B}\,\otimes\mathsf{R}}{A'\otimes B' \mid \Gamma,\Delta \vdash A\otimes B}\,\otimes\mathsf{L}$$

$$\dfrac{\dfrac{\overset{f}{- \mid \Gamma \vdash C} \quad \overset{g}{D \mid \Delta \vdash A}}{C \multimap D \mid \Gamma,\Delta \vdash A}\,\multimap\mathsf{L} \quad \overset{h}{- \mid \Lambda \vdash B}}{C \multimap D \mid \Gamma,\Delta,\Lambda \vdash A\otimes B}\,\otimes\mathsf{R} \;\circeq\; \dfrac{\overset{f}{- \mid \Gamma \vdash C} \quad \dfrac{\overset{g}{D \mid \Delta \vdash A} \quad \overset{h}{- \mid \Lambda \vdash B}}{D \mid \Delta,\Lambda \vdash A\otimes B}\,\otimes\mathsf{R}}{C \multimap D \mid \Gamma,\Delta,\Lambda \vdash A\otimes B}\,\multimap\mathsf{L}$$

$$\dfrac{\dfrac{\overset{f}{A' \mid \Gamma, A \vdash B}}{A' \mid \Gamma \vdash A \multimap B}\,\multimap\mathsf{R}}{- \mid A',\Gamma \vdash A \multimap B}\,\mathsf{pass} \;\circeq\; \dfrac{\dfrac{\overset{f}{A' \mid \Gamma, A \vdash B}}{- \mid A',\Gamma,A \vdash B}\,\mathsf{pass}}{- \mid A',\Gamma \vdash A \multimap B}\,\multimap\mathsf{R}$$

$$\dfrac{\dfrac{\overset{f}{- \mid \Gamma, A \vdash B}}{- \mid \Gamma \vdash A \multimap B}\,\multimap\mathsf{R}}{\mathsf{I} \mid \Gamma \vdash A \multimap B}\,\mathsf{IL} \;\circeq\; \dfrac{\dfrac{\overset{f}{- \mid \Gamma, A \vdash B}}{\mathsf{I} \mid \Gamma,A \vdash B}\,\mathsf{IL}}{\mathsf{I} \mid \Gamma \vdash A \multimap B}\,\multimap\mathsf{R}$$

$$\dfrac{\dfrac{\overset{f}{A' \mid B',\Gamma, A \vdash B}}{A' \mid B',\Gamma \vdash A \multimap B}\,\multimap\mathsf{R}}{A'\otimes B' \mid \Gamma \vdash A \multimap B}\,\otimes\mathsf{L} \;\circeq\; \dfrac{\dfrac{\overset{f}{A' \mid B',\Gamma, A \vdash B}}{A'\otimes B' \mid \Gamma,A \vdash B}\,\otimes\mathsf{L}}{A'\otimes B' \mid \Gamma \vdash A \multimap B}\,\multimap\mathsf{R}$$

$$\dfrac{\overset{f}{- \mid \Gamma \vdash A'} \quad \dfrac{\overset{g}{B' \mid \Delta, A \vdash B}}{B' \mid \Delta \vdash A \multimap B}\,\multimap\mathsf{R}}{A' \multimap B' \mid \Gamma,\Delta \vdash A \multimap B}\,\multimap\mathsf{L} \;\circeq\; \dfrac{\dfrac{\overset{f}{- \mid \Gamma \vdash A'} \quad \overset{g}{B' \mid \Delta, A \vdash B}}{A' \multimap B' \mid \Gamma, A \vdash B}\,\multimap\mathsf{L}}{A' \multimap B' \mid \Gamma,\Delta \vdash A \multimap B}\,\multimap\mathsf{R}$$

Fig. 2 Equivalence of derivations: permutative conversions

$$\dfrac{\overset{f}{S \mid \Gamma_0 \vdash A} \qquad \dfrac{\overset{g}{- \mid \Gamma_2 \vdash B} \quad \overset{h}{A \mid \Gamma_1, B, \Gamma_3 \vdash C}}{A \mid \Gamma_1, \Gamma_2, \Gamma_3 \vdash C}\,\mathsf{ccut}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3 \vdash C}\,\mathsf{scut}$$

$$= \dfrac{\overset{g}{- \mid \Gamma_2 \vdash B} \qquad \dfrac{\overset{f}{S \mid \Gamma_0 \vdash A} \quad \overset{h}{A \mid \Gamma_1, B, \Gamma_3 \vdash C}}{S \mid \Gamma_0, \Gamma_1, B, \Gamma_3 \vdash C}\,\mathsf{scut}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3 \vdash C}\,\mathsf{ccut}$$

$$\dfrac{\overset{f}{- \mid \Gamma_1 \vdash A} \qquad \dfrac{\overset{g}{- \mid \Gamma_3 \vdash B} \quad \overset{h}{S \mid \Gamma_0, A, \Gamma_2, B, \Gamma_4 \vdash C}}{S \mid \Gamma_0, A, \Gamma_2, \Gamma_3, \Gamma_4 \vdash C}\,\mathsf{ccut}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4 \vdash C}\,\mathsf{ccut}$$

$$= \dfrac{\overset{g}{- \mid \Gamma_3 \vdash B} \qquad \dfrac{\overset{f}{- \mid \Gamma_1 \vdash A} \quad \overset{h}{S \mid \Gamma_0, A, \Gamma_2, B, \Gamma_4 \vdash C}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, B, \Gamma_4 \vdash C}\,\mathsf{ccut}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4 \vdash C}\,\mathsf{ccut}$$

Fig. 3 Commutativity of cut

implication) is defined as follows:

$$\dfrac{\dfrac{\overset{f}{S \mid \Gamma, A \vdash B}}{S \mid \Gamma \vdash A \multimap B}\,\multimap\mathsf{R} \qquad \dfrac{\overset{g}{- \mid \Delta \vdash A} \quad \overset{h}{B \mid \Lambda \vdash C}}{A \multimap B \mid \Delta, \Lambda \vdash C}\,\multimap\mathsf{L}}{S \mid \Gamma, \Delta, \Lambda \vdash C}\,\mathsf{scut}$$

$$= \dfrac{\dfrac{\overset{g}{- \mid \Delta \vdash A} \quad \overset{f}{S \mid \Gamma, A \vdash B}}{S \mid \Gamma, \Delta \vdash B}\,\mathsf{ccut} \qquad \overset{h}{B \mid \Lambda \vdash C}}{S \mid \Gamma, \Delta, \Lambda \vdash C}\,\mathsf{scut}$$

This equation can be generalized to one where $\multimap\mathsf{R}$, $\multimap\mathsf{L}$ and ccut are replaced by their iterated versions $\multimap\mathsf{R}^*$, $\multimap\mathsf{L}^*$ and ccut^* introduced in Equations (3), (4) and (5).

Proposition 2. *Given a list of formulae $\Lambda = A_1, \ldots, A_n$, a derivation $f : S \mid \Gamma_0, \Lambda \vdash B$ and a list of derivations $g_i : - \mid \Delta_i \vdash A_i$ for $i \in [1, \ldots, n]$, the following equation is*

$$\dfrac{\overset{f}{S \mid \Gamma_0 \vdash A} \qquad \dfrac{\overset{g}{A \mid \Gamma_1 \vdash B} \quad \overset{h}{B \mid \Gamma_2 \vdash C}}{A \mid \Gamma_1, \Gamma_2 \vdash C}\,\mathsf{scut}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2 \vdash C}\,\mathsf{scut}$$

$$= \dfrac{\dfrac{\overset{f}{S \mid \Gamma_0 \vdash A} \quad \overset{g}{A \mid \Gamma_1 \vdash B}}{S \mid \Gamma_0, \Gamma_1 \vdash B}\,\mathsf{scut} \qquad \overset{h}{B \mid \Gamma_2 \vdash C}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2 \vdash C}\,\mathsf{scut}$$

$$\dfrac{\overset{f}{- \mid \Gamma_1 \vdash A} \qquad \dfrac{\overset{g}{S \mid \Gamma_0, A, \Gamma_2 \vdash B} \quad \overset{h}{B \mid \Gamma_3 \vdash C}}{S \mid \Gamma_0, A, \Gamma_2, \Gamma_3 \vdash C}\,\mathsf{scut}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3 \vdash C}\,\mathsf{ccut}$$

$$= \dfrac{\dfrac{\overset{f}{- \mid \Gamma_1 \vdash A} \quad \overset{g}{S \mid \Gamma_0, A, \Gamma_2 \vdash B}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2 \vdash B}\,\mathsf{ccut} \qquad \overset{h}{B \mid \Gamma_3 \vdash C}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3 \vdash C}\,\mathsf{scut}$$

$$\dfrac{\overset{f}{- \mid \Gamma_2 \vdash A} \qquad \dfrac{\overset{g}{- \mid \Gamma_1, A, \Gamma_3 \vdash B} \quad \overset{h}{S \mid \Gamma_0, B, \Gamma_4 \vdash C}}{S \mid \Gamma_0, \Gamma_1, A, \Gamma_3, \Gamma_4 \vdash C}\,\mathsf{ccut}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4 \vdash C}\,\mathsf{ccut}$$

$$= \dfrac{\dfrac{\overset{f}{- \mid \Gamma_2 \vdash A} \quad \overset{g}{- \mid \Gamma_1, A, \Gamma_3 \vdash B}}{S \mid \Gamma_1, \Gamma_2, \Gamma_3 \vdash B}\,\mathsf{ccut} \qquad \overset{h}{S \mid \Gamma_0, B, \Gamma_4 \vdash C}}{S \mid \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4 \vdash C}\,\mathsf{ccut}$$

Fig. 4 Associativity of cut

derivable:

$$\dfrac{\dfrac{\overset{f}{S \mid \Gamma_0, \Lambda \vdash B}}{S \mid \Gamma_0 \vdash \Lambda \multimap^* B}\,{\multimap}\mathsf{R}^* \qquad \dfrac{\overset{[g_i]}{[- \mid \Delta_i \vdash A_i]_i} \quad \overset{h}{B \mid \Gamma_1 \vdash C}}{\Lambda \multimap^* B \mid \Delta_1, \dots, \Delta_n, \Gamma_1 \vdash C}\,{\multimap}\mathsf{L}^*}{S \mid \Gamma_0, \Delta_1, \dots, \Delta_n, \Gamma_1 \vdash C}\,\mathsf{scut}$$

$$= \dfrac{\dfrac{\overset{[g_i]}{[- \mid \Delta_i \vdash A_i]_i} \quad \overset{f}{S \mid \Gamma_0, \Delta_1, \dots, \Delta_n \vdash B}}{S \mid \Gamma_0, \Delta_1, \dots, \Delta_n \vdash B}\,\mathsf{ccut}^* \qquad \overset{h}{B \mid \Gamma_1 \vdash C}}{S \mid \Gamma_0, \Delta_1, \dots, \Delta_n, \Gamma_1 \vdash C}\,\mathsf{scut}$$

Proof. Proving the validity of the equation requires various applications of the associativity equations in Proposition 1. □

The admissibility of rule scut is proved in [12, 19] by structural recursion on the derivation of the left premise. This implies that "scut commutes with left rules in first premise", i.e. that $\mathsf{scut}(\odot\mathsf{L}\ f, g) = \odot\mathsf{L}(\mathsf{scut}(f, g)$ for any one-premise left rule $\odot\mathsf{L}$ among IL, $\otimes\mathsf{L}$ and pass, and also $\mathsf{scut}({\multimap}\mathsf{L}(f, f'), g) = {\multimap}\mathsf{L}(f, \mathsf{scut}(f', g)$. It is possible to also show that "scut commutes with right rules in second premise", but only up to equivalence $\circeq$.

Proposition 3. *The following equivalences of derivations involving* scut, $\otimes\mathsf{R}$, *and* ${\multimap}\mathsf{R}$ *are admissible in* `SkNMILL`*:*

$$\dfrac{\overset{f}{S \mid \Gamma \vdash A} \qquad \dfrac{\overset{g}{A \mid \Delta \vdash B} \qquad \overset{h}{- \mid \Lambda \vdash C}}{A \mid \Delta, \Lambda \vdash B \otimes C}\,\otimes\mathsf{R}}{S \mid \Gamma, \Delta, \Lambda \vdash B \otimes C}\,\mathsf{scut} \;\circeq\; \dfrac{\dfrac{\overset{f}{S \mid \Gamma \vdash A} \qquad \overset{g}{A \mid \Delta \vdash B}}{S \mid \Gamma, \Delta \vdash B}\,\mathsf{scut} \qquad \overset{h}{- \mid \Lambda \vdash C}}{S \mid \Gamma, \Delta, \Lambda \vdash B \otimes C}\,\otimes\mathsf{R}$$

$$\dfrac{\overset{f}{S \mid \Gamma \vdash B} \qquad \dfrac{\overset{g}{A \mid \Delta, B \vdash C}}{A \mid \Delta \vdash B \multimap C}\,{\multimap}\mathsf{R}}{S \mid \Gamma, \Delta \vdash B \multimap C}\,\mathsf{scut} \;\circeq\; \dfrac{\dfrac{\overset{f}{S \mid \Gamma \vdash A} \qquad \overset{g}{A \mid \Delta, B \vdash C}}{S \mid \Gamma, \Delta, B \vdash C}\,\mathsf{scut}}{S \mid \Gamma, \Delta \vdash B \multimap C}\,{\multimap}\mathsf{R}$$

Proof. Both equivalences are proved by structural induction on the derivation f. □

4 Failure of Maehara Interpolation

The goal of this paper is proving that the logic `SkNMILL` satisfies the Craig interpolation property. But, as already mentioned in the introductive section, we cannot follow the same proof strategy used in the (associative or non-associative) Lambek calculus, where Craig interpolation follows as a corollary to Maehara interpolation. This is because the sequent calculus of `SkNMILL` does not satisfy Maehara interpolation. Let us see why.

First, in analogy with the presence of two admissible cut rules (2), there are also two different form of interpolation. This is because the subsequence of the antecedents for which we wish to find an interpolant can either contain the stoup or it can be fully included in the context. More explicitly, given an antecedent $S \mid \Gamma$, we can either: (i) split the context $\Gamma = \Gamma_1, \Gamma_2$ in two parts and look for an interpolant of the sub-antecedent $S \mid \Gamma_1$, or (ii) split the context $\Gamma = \Gamma_0, \Gamma_1, \Gamma_2$ in three parts and look for an interpolant of the sub-context Γ_1. The Maehara interpolation property in `SkNMILL` would then consist of two statement, a *stoup Maehara interpolation* (sMIP) and a *context Maehara interpolation* (cMIP):

(sMIP) Given $f : S \mid \Gamma \vdash C$ and a partition $\langle \Gamma_0, \Gamma_1 \rangle$ of Γ, there exists
- an interpolant formula D,
- a derivation $g : S \mid \Gamma_0 \vdash D$,
- a derivation $h : D \mid \Gamma_1 \vdash C$, such that
- $\mathsf{var}(D) \subseteq \mathsf{var}(S, \Gamma_0) \cap \mathsf{var}(\Gamma_1, C)$.

(cMIP) Given $f : S \mid \Gamma \vdash C$ and a partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$ of Γ, there exists
- an interpolant formula D,
- a derivation $g : - \mid \Gamma_1 \vdash D$,

- a derivation $h : S \mid \Gamma_0, D, \Gamma_2 \vdash C$, such that
- $\mathsf{var}(D) \leq \mathsf{var}(\Gamma_1) \cap \mathsf{var}(S, \Gamma_0, \Gamma_2, C)$.

However, this property is not provable in `SkNMILL`. The problem lays in the validity of the second statement cMIP. An attempt to prove this would proceed by induction on the height of the derivation $f : S \mid \Gamma \vdash C$ and then inspecting what is the last rule applied in f. The rules $\otimes\mathsf{R}$ and $\multimap\mathsf{L}$ split the context, so one should be careful to consider all possible ways in which these splittings relate to the given partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$ of Γ.

The critical case is $f = \otimes\mathsf{R}(f', f'')$ with the partition $\langle \Gamma_0, (\Gamma_1', \Gamma_1''), \Gamma_2 \rangle$ and two derivations $f' : S \mid \Gamma_0, \Gamma_1' \vdash A$ and $f'' : - \mid \Gamma_1'', \Gamma_2 \vdash B$. So this is the case when the $\otimes\mathsf{R}$ rule splits Γ_1 in two parts Γ_1', Γ_1''. By induction on f' and the partition $\langle \Gamma_0, \Gamma_1', [\,] \rangle$, we would be given a formula D and derivations $g' : - \mid \Gamma_1' \vdash D$, and $h' : S \mid \Gamma_0, D \vdash A$. By applying the inductive hypothesis on f'' and the partition $\langle [\,], \Gamma_1'', \Gamma_2 \rangle$, we would be given a formula E and derivations $g'' : - \mid \Gamma_1'' \vdash E$ and $h'' : - \mid E, \Gamma_2 \vdash B$. We obtain $\otimes\mathsf{R}(g', g'') : - \mid \Gamma_1', \Gamma_1'' \vdash D \otimes E$, but we are unable to construct the other desired proof of sequent $S \mid \Gamma_0, D \otimes E, \Gamma_1 \vdash A \otimes B$. We get very close via $\otimes\mathsf{R}(h', h'') : S \mid \Gamma_0, D, E, \Gamma_1 \vdash A \otimes B$, but we are unable to merge D and E into $D \otimes E$, since in our calculus the $\otimes\mathsf{L}$ cannot be applied on formulae in context.

For a simple concrete counterexample, consider the derivation

$$\dfrac{\dfrac{\dfrac{}{X \mid \ \vdash X}\,\mathsf{ax} \qquad \dfrac{\dfrac{}{Y \mid \ \vdash Y}\,\mathsf{ax}}{- \mid Y \vdash Y}\,\mathsf{pass}}{X \mid Y \vdash X \otimes Y}\,\otimes\mathsf{R} \qquad \dfrac{\dfrac{}{Z \mid \ \vdash Z}\,\mathsf{ax}}{- \mid Z \vdash Z}\,\mathsf{pass}}{X \mid Y, Z \vdash (X \otimes Y) \otimes Z}\,\otimes\mathsf{R}$$

and the partition $\langle [\,], (Y, Z), [\,] \rangle$. Suppose by contradiction that Maehara interpolation holds, so we would have a formula D and two derivations $g : - \mid Y, Z \vdash D$ and $h : X \mid D \vdash (X \otimes Y) \otimes Z$. The variable condition of Maehara interpolation and the existence of the derivation g ensure that D does not contain atomic formulae other than Y and Z, and the latter must have a unique occurrence in D. Nevertheless, the existence of derivation h is absurd. Since X is atomic, h can only be of the form: (i) $f = \otimes\mathsf{R}(f_1, f_2)$ for some derivations $f_1 : X \mid D \vdash X \otimes Y$ and $f_2 : - \mid \ \vdash Z$, or (ii) $f = \otimes\mathsf{R}(f_1', f_2')$ for some derivations $f_1' : X \mid \ \vdash X \otimes Y$ and $f_2' : - \mid D \vdash Z$. Case (i) is impossible since there is no such f_2, while case (ii) is impossible since there is no such f_1'.

This situation is reminiscent of proving interpolation in the product-free Lambek calculus [10] and in the implicational fragment of intuitionistic logic [11]. In both these cases, Maehara interpolation fails because none of the additive $(\wedge)$ and multiplicative $(\otimes)$ conjunction is present. A concrete counterexample in the product-free Lambek calculus (adapted from Kanazawa [11]) is given by the derivable sequent $W, W\backslash Y, W, W\backslash X, X\backslash(Y\backslash Z) \vdash Z$ with the partition $\langle [\ ], [W, W\backslash Y, W, W\backslash X], [X\backslash(Y\backslash Z)] \rangle$. This can be shown to not satisfy Maehara interpolation property. In the presence of $\otimes$, Maehara's method would produce the interpolant formula $X \otimes Y$. The situation of `SkNMILL` is somewhere inbetween: we have a multiplicative conjunction $\otimes$ but we cannot do much with it if a formula $A \otimes B$ is in context instead of the stoup position, since the rule $\otimes\mathsf{L}$ cannot be applied arbitrarily in the antecedent. The counterexample to MIP in product-free sequent calculus can, when appropriately modified, also works as a counterexample to cMIP in $\mathtt{SkNMILL}$: consider the derivable sequent $X \multimap (Y \multimap Z) \mid W \multimap X, W, W \multimap Y, W \vdash Z$ with the partition $\langle [\,], [W \multimap X, W, W \multimap Y, W], [\,] \rangle$.

5 Craig Interpolation for SkNMILL

In this section, we show that SkNMILL enjoys Craig interpolation, even though it does not generally enjoy Maehara interpolation. This is again in analogy with the product-free Lambek calculus. As mentioned in the introductive section, Pentus [10] proved that the latter satisfies a relaxation of Maehara interpolation, that we dubbed Maehara multi-interpolation (MMIP), which is sufficient to show Craig interpolation.

Here is a brief sketch of the proof. Suppose the formula $A\backslash B$ is provable in product-free Lambek calculus. This implies that there exists a derivation $f : A \vdash B$. Apply the Maehara multi-interpolation procedure to f and the partition $\langle [\,], [A], [\,] \rangle$. This produces a partition $\langle \Delta_1, \dots, \Delta_n \rangle$ of $[A]$. Since $[A]$ is a singleton list, all Δ_i must be empty apart from one which is equal to $[A]$. Maehara multi-interpolation also produces formulae $D_1, \dots, D_n$ satisfying the variable condition $\sigma_X(D_i) \leq \sigma_X(\Delta_i)$ for all i and atomic formulae X. The latter cannot be true if Δ_i is empty, since this logic has no units. This implies $n = 1$. Therefore, Maehara multi-interpolation in this case produces only two derivations $h : A \vdash D_1$ and $g : D_1 \vdash B$, i.e. D_1 is the Craig interpolant of A and B. A similar proof also works for the product-free Lambek calculus enhanced with a multiplicative unit.

We showed in the previous section that SkNMILL does not satisfy the context Maehara interpolation property (cMIP). We prove now that instead it satisfies a *context Maehara multi-interpolation property* (cMMIP). And the stoup Maehara interpolation property (sMIP) also holds.

Theorem 4. *In the sequent calculus for* SkNMILL*, the following two interpolation properties hold:*

(sMIP) *Given a derivation $f : S \mid \Gamma \vdash C$ and a partition $\langle \Gamma_0, \Gamma_1 \rangle$ of Γ, there exist*
- *an interpolant formula D,*
- *a derivation $g : S \mid \Gamma_0 \vdash D$,*
- *a derivation $h : D \mid \Gamma_1 \vdash C$, such that*
- $\mathsf{var}(D) \subseteq \mathsf{var}(S, \Gamma_0) \cap \mathsf{var}(\Gamma_1, C)$.

(cMMIP) *Given a derivation $f : S \mid \Gamma \vdash C$ and a partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$ of Γ, there exist*
- *a partition $\langle \Delta_1, \dots, \Delta_n \rangle$ of Γ_1,*
- *a list of interpolant formulae $D_1, \dots, D_n$,*
- *$g : S \mid \Gamma_0, D_1, \dots, D_n, \Gamma_2 \vdash C$,*
- *$h_i : - \mid \Delta_i \vdash D_i$ for all $i \in [1, \dots, n]$, such that*
- $\mathsf{var}(D_1, \dots, D_n) \subseteq \mathsf{var}(\Delta_1, \dots, \Delta_n) \cap \mathsf{var}(S, \Gamma_0, \Gamma_2, C)$.

These two statements of the theorem are proved mutually by structural induction on derivations. We separate the proofs for readability.

Proof of sMIP. We proceed by induction on the structure of f.
<u>Case $f = \mathsf{ax}$.</u> Suppose $f = \mathsf{ax} : A \mid \;\vdash A$, which forces $\Gamma_0 = \Gamma_1 = [\,]$. In this case, the

interpolant formula is A and $g = h = \mathsf{ax} : A \mid \ \vdash A$, where the variable condition is automatically satisfied.

Case $f = \mathsf{IR}$. Since $f : - \mid \ \vdash \mathsf{I}$, this forces again Γ_0 and Γ_1 to be empty lists. In this case, the interpolant formula is I and $g = \mathsf{IR} : - \mid \ \vdash \mathsf{I}$ and $h = \mathsf{IL}(\mathsf{IR}) : \mathsf{I} \mid \ \vdash \mathsf{I}$, where the variable condition is vacuously satisfied.

Case $f = \mathsf{IL}\ f'$. Given a derivation $f' : - \mid \Gamma \vdash C$, by induction on f' with the same partition $\langle \Gamma_0, \Gamma_1 \rangle$ of Γ we obtain

- a formula D,
- a derivation $g' : - \mid \Gamma_0 \vdash D$,
- a derivation $h' : D \mid \Gamma_1 \vdash C$, such that
- $\mathsf{var}(D) \subseteq \mathsf{var}(\Gamma_0) \cap \mathsf{var}(\Gamma_1, C)$.

In this case, the interpolant formula for f is D and the two desired derivations are $g = \mathsf{IL}\ g'$ and $h = h'$. The variable condition is automatically satisfied.

Cases $f = \otimes\mathsf{L}\ f'$ and $f = \multimap\mathsf{R}\ f'$. Analogous to the previous case.

Case $f = \mathsf{pass}\ f'$. Let $f' : A \mid \Gamma' \vdash C$ and $\Gamma = A, \Gamma'$. There are two subcases determined by the partition $\langle \Gamma_0, \Gamma_1 \rangle$ of Γ. Specifically, either Γ_0 is an empty list or not.

- If $\Gamma_0 = [\ ]$, then the interpolant is I and two desired derivations are IR and $\mathsf{IL}(\mathsf{pass}\ f')$. The variable condition is satisfied because $\mathsf{var}(\mathsf{I}) = \emptyset$.
- If $\Gamma_0 = A, \Gamma'_0$, then by induction on f' with the partition $\langle \Gamma'_0, \Gamma_1 \rangle$ we obtain
  - a formula D,
  - a derivation $g' : A \mid \Gamma'_0 \vdash D$,
  - a derivation $h' : D \mid \Gamma_1 \vdash C$, such that
  - $\mathsf{var}(D) \subseteq \mathsf{var}(A, \Gamma'_0) \cap \mathsf{var}(\Gamma_1, C)$.

  In this case, the interpolant formula for f is D, and two desired derivations are $g = \mathsf{pass}\ g'$ and $h = h'$. The variable condition follows directly from the inductive hypothesis.

Case $f = \otimes\mathsf{R}(f', f'')$. Let $f' : S \mid \Lambda \vdash A$ and $f'' : - \mid \Omega \vdash B$, so that $\Gamma = \Lambda, \Omega$. We need to check how the latter splitting of Γ compares to the given partition $\langle \Gamma_0, \Gamma_1 \rangle$. There are two possibilities:

- Γ_0 is fully contained in Λ. This means that $\Lambda = \Gamma_0, \Gamma'_1$ and $\Gamma_1 = \Gamma'_1, \Omega$. Then $f' : S \mid \Gamma_0, \Gamma'_1 \vdash A$ and $f'' : - \mid \Omega \vdash B$. In this case, by induction on f' with the partition $\langle \Gamma_0, \Gamma'_1 \rangle$ we obtain
  - a formula D,
  - a derivation $g' : S \mid \Gamma_0 \vdash D$,
  - a derivation $h' : D \mid \Gamma'_1 \vdash A$ such that
  - $\mathsf{var}(D) \subseteq \mathsf{var}(S, \Gamma_0) \cap \mathsf{var}(\Gamma'_1, A)$.

  The desired interpolant formula is D and the desired derivations are $g = g'$ and $h = \otimes\mathsf{R}(h', f'') : D \mid \Gamma'_1, \Omega \vdash A \otimes B$. The variable condition is satisfied because $\mathsf{var}(D) \subseteq \mathsf{var}(\Gamma'_1, A) \subseteq \mathsf{var}(\Gamma'_1, \Omega, A \otimes B)$.
- Γ_0 splits between Λ and Ω. This means that $\Gamma_0 = \Lambda, \Gamma'_0$ and $\Omega = \Gamma'_0, \Gamma_1$, and Γ'_0 is non-empty. Then $f' : S \mid \Lambda \vdash A$ and $f'' : - \mid \Gamma'_0, \Gamma_1 \vdash B$. In this case, by induction on f' with the partition $\langle \Lambda, [\ ] \rangle$ and on f'' with the partition $\langle \Gamma'_0, \Gamma_1 \rangle$, respectively, we obtain
  - formulae E and F,
  - derivations $g' : S \mid \Lambda \vdash E$ and $g'' : - \mid \Gamma'_0 \vdash F$,

  – derivations $h' : E \mid \ \vdash A$ and $h'' : F \mid \Gamma_1 \vdash B$, such that
  – $\mathsf{var}(E) \subseteq \mathsf{var}(S, \Lambda) \cap \mathsf{var}(A)$, and
  – $\mathsf{var}(F) \subseteq \mathsf{var}(\Gamma_0') \cap \mathsf{var}(\Gamma_1, B)$.

The desired interpolant formula is $D = E \otimes F$ and the desired derivations are

$$g = \dfrac{\overset{g'}{S \mid \Lambda \vdash E} \qquad \overset{g''}{- \mid \Gamma_0' \vdash F}}{S \mid \Lambda, \Gamma_0' \vdash E \otimes F}\ \otimes\mathsf{R} \qquad\qquad h = \dfrac{\dfrac{\overset{h'}{E \mid \ \vdash A} \qquad \dfrac{\overset{h''}{F \mid \Gamma_1 \vdash B}}{- \mid F, \Gamma_1 \vdash B}\ \mathsf{pass}}{E \mid F, \Gamma_1 \vdash A \otimes B}\ \otimes\mathsf{R}}{E \otimes F \mid \Gamma_1 \vdash A \otimes B}\ \otimes\mathsf{L}$$

The variable condition is satisfied because $\mathsf{var}(E \otimes F) = \mathsf{var}(E) \cup \mathsf{var}(F) \subseteq \mathsf{var}(S, \Lambda) \cup \mathsf{var}(\Gamma_0') = \mathsf{var}(S, \Lambda, \Gamma_0')$ and $\mathsf{var}(E \otimes F) = \mathsf{var}(E) \cup \mathsf{var}(F) \subseteq \mathsf{var}(A) \cup \mathsf{var}(\Gamma_1, B) = \mathsf{var}(A, \Gamma_1, B) = \mathsf{var}(\Gamma_1, A \otimes B)$.

Case $f = \multimap\mathsf{L}(f', f'')$. Let $f' : - \mid \Lambda \vdash A$ and $f'' : B \mid \Omega \vdash C$, so that $\Gamma = \Lambda, \Omega$. Again we check how the latter splitting of Γ compares to the given partition $\langle \Gamma_0, \Gamma_1 \rangle$. There are two possibilities:

- Γ_1 is fully contained in Ω. This means that $\Gamma_0 = \Lambda, \Gamma_0'$ and $\Omega = \Gamma_0', \Gamma_1$. Then $f' : - \mid \Lambda \vdash A$ and $f'' : B \mid \Gamma_0', \Gamma_1 \vdash C$. In this case, by induction on f'' with the partition $\langle \Gamma_0', \Gamma_1 \rangle$ we obtain
  – a formula D,
  – a derivation $g'' : B \mid \Gamma_0' \vdash D$,
  – a derivation $h'' : D \mid \Gamma_1 \vdash C$ such that
  – $\mathsf{var}(D) \subseteq \mathsf{var}(B, \Gamma_0') \cap \mathsf{var}(\Gamma_1, C)$.

  The desired interpolant formula is D and the desired derivations are $g = \multimap\mathsf{L}(f', g'') : A \multimap B \mid \Lambda, \Gamma_0' \vdash D$ and $h = h''$. The variable condition is satisfied because $\mathsf{var}(D) \subseteq \mathsf{var}(B, \Gamma_0') \subseteq \mathsf{var}(A \multimap B, \Lambda, \Gamma_0')$.
- Γ_1 splits between Λ and Ω. This means that $\Lambda = \Gamma_0, \Gamma_1'$ and $\Gamma_1 = \Gamma_1', \Omega$, and Γ_1' is non-empty. Then $f' : - \mid \Gamma_0, \Gamma_1' \vdash A$ and $f'' : B \mid \Omega \vdash C$. Our goal is to find a formula D and derivations $g : A \multimap B \mid \Gamma_0 \vdash D$ and $h : D \mid \Gamma_1', \Omega \vdash C$. By induction on f'' with the partition $\langle [\,], \Omega \rangle$ we obtain
  – a formula E,
  – a derivation $g'' : B \mid \ \vdash E$,
  – a derivation $h'' : E \mid \Omega \vdash C$ such that
  – $\mathsf{var}(E) \subseteq \mathsf{var}(B) \cap \mathsf{var}(\Omega, C)$.

  We also apply the cMMIP procedure (which, remember, is proved by mutual induction with sMIP) on the derivation f' with the partition $\langle \Gamma_0, \Gamma_1', [\,] \rangle$ and obtain
  – a partition $\langle \Delta_1, \ldots, \Delta_n \rangle$ of Γ_1',
  – a list of formulae $D_1, \ldots, D_n$,
  – a derivation $g' : - \mid \Gamma_0, D_1, \ldots, D_n \vdash A$,
  – a list of derivations $h_i' : - \mid \Delta_i \vdash D_i$, for $i \in [1, \ldots, n]$, such that
  – $\mathsf{var}(D_1, \ldots, D_n) \subseteq \mathsf{var}(\Delta_1, \ldots, \Delta_n) \cap \mathsf{var}(\Gamma_0, A)$.

The desired interpolant formula is $D = D_1 \multimap (D_2 \multimap (\dots(D_n \multimap E)\dots))$. The desired derivations g and h are constructed as follows:

$$g = \dfrac{\dfrac{\overset{g'}{- \mid \Gamma_0, D_1, \dots, D_n \vdash A} \quad \overset{g''}{B \mid \ \vdash E}}{A \multimap B \mid \Gamma_0, D_1, \dots, D_n \vdash E}\ \multimap\mathsf{L}}{A \multimap B \mid \Gamma_0 \vdash D_1 \multimap (\dots(D_n \multimap E)\dots)}\ \multimap\mathsf{R}^*$$

$$h = \dfrac{\overset{[h'_i]}{[- \mid \Delta_i \vdash D_i]_i} \quad \overset{h''}{E \mid \Omega \vdash C}}{D_1 \multimap (\dots(D_n \multimap E)\dots) \mid \Delta_1, \dots, \Delta_n, \Omega \vdash C}\ \multimap\mathsf{L}^*$$

Notice that $\Gamma'_1 = \Delta_1, \dots, \Delta_n$, so the variable condition is easy to check.

□

Proof of cMMIP. We proceed by induction on the structure of f.
<u>Case $f = \mathsf{ax}$.</u> Suppose $f = \mathsf{ax} : A \mid \ \vdash A$, which means that $\Gamma_0 = \Gamma_1 = \Gamma_2 = [\,]$. In this case, the desired partition of Γ_1 is the empty one $\langle [\,] \rangle$, i.e. $n = 0$. The desired lists of formulae D_i and of derivations h_i are also empty. The desired derivation g is ax.
<u>Case $f = \mathsf{IR}$.</u> Similar to the previous one.
<u>Case $f = \mathsf{IL}\ f'$.</u> Given a derivation $f' : - \mid \Gamma \vdash C$, by induction on f' with the same partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$ of Γ we obtain

– a partition $\langle \Delta_0, \dots, \Delta_n \rangle$ of Γ_1,
– a list of interpolant formulae $D_1, \dots, D_n$,
– a derivation $g' : - \mid \Gamma_0, D_1, \dots, D_n, \Gamma_2 \vdash C$,
– derivations $h'_i : - \mid \Delta_i \vdash D_i$, for $i \in [1, \dots, n]$, such that
– $\mathsf{var}(D_1, \dots, D_n) \subseteq \mathsf{var}(\Delta_1, \dots, \Delta_n) \cap \mathsf{var}(S, \Gamma_0, \Gamma_2, C)$ for all X.

The desired partition of Γ_1 is $\langle \Delta_0, \dots, \Delta_n \rangle$, the desired list of interpolant formulae is $D_1, \dots, D_n$. The desired derivations are $g = \mathsf{IL}\ g'$ and $h_i = h'_i$ for $i \in [1, \dots, n]$. The variable condition is automatically satisfied.
<u>Cases $f = \otimes\mathsf{L}\ f'$ and $f = \multimap\mathsf{R}\ f'$.</u> Analogous to the previous case.
<u>Case $f = \mathsf{pass}\ f'$.</u> Let $f' : A \mid \Gamma' \vdash C$ and $\Gamma = A, \Gamma'$. There are subcases determined by the partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$ of Γ. The most interesting case is the one where $\Gamma_0 = [\,]$ and $\Gamma_1 = A, \Gamma'_1$, so that $\Gamma' = \Gamma'_1, \Gamma_2$. The other possible cases are handled similarly to the IL case discussed above. We apply the sMIP procedure (which, remember, is proved by mutual induction with cMMIP) on the derivation f' and the partition $\langle \Gamma'_1, \Gamma_2 \rangle$, which gives us

– a formula D,
– a derivation $g' : A \mid \Gamma'_1 \vdash D$,
– a derivation $h' : D \mid \Gamma_2 \vdash C$, such that
– $\mathsf{var}(D) \subseteq \mathsf{var}(A, \Gamma'_1) \subseteq \mathsf{var}(\Gamma_2, C)$.

The desired partition of A, Γ'_1 is the singleton context $[A, \Gamma'_1]$, i.e. $n = 1$. The desired list of interpolant formulae is the singleton $[D]$. The desired derivation g is $\mathsf{pass}\ g' : - \mid A, \Gamma'_1 \vdash D$ and the desired list of derivations h_i is the singleton consisting only of $\mathsf{pass}\ h' : - \mid D, \Gamma_2 \vdash C$. The variable condition follows from the inductive hypothesis.

Case $f = \otimes\mathsf{R}(f', f'')$. Let $f' : S \mid \Lambda \vdash A$ and $f'' : - \mid \Omega \vdash B$, so that $\Gamma = \Lambda, \Omega$. We need to check how the latter splitting of Γ compares to the given partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$. There are three possibilities:

- Γ_1 is fully contained in Ω. This means that $\Gamma_0 = \Lambda, \Gamma'_0$ and $\Omega = \Gamma'_0, \Gamma_1, \Gamma_2$. Then $f' : S \mid \Lambda \vdash A$ and $f'' : - \mid \Gamma'_0, \Gamma_1, \Gamma_2 \vdash B$. By induction on f'' with partition $\langle \Gamma'_0, \Gamma_1, \Gamma_2 \rangle$ we obtain
  - a partition $\langle \Delta_0, \dots, \Delta_n \rangle$ of Γ_1,
  - a list of interpolant formulae $D_1, \dots, D_n$,
  - a derivation $g'' : - \mid \Gamma'_0, D_1, \dots, D_n, \Gamma_2 \vdash B$,
  - derivations $h''_i : - \mid \Delta_i \vdash D_i$, for $i \in [1, \dots, n]$, such that
  - $\mathsf{var}(D_1, \dots, D_n) \subseteq \mathsf{var}(\Delta_1, \dots, \Delta_n) \cap \mathsf{var}(\Gamma'_0, \Gamma_2, B)$.

  The desired partition of Γ_1 is $\langle \Delta_0, \dots, \Delta_n \rangle$. The desired list of interpolant formulae is $D_1, \dots, D_n$. The desired derivation g is $\otimes\mathsf{R}(f', g'')$ and the desired derivation h_i is h''_i for $i \in [1, \dots, n]$. The variable condition is satisfied because $\mathsf{var}(D_1, \dots, D_n) \subseteq \mathsf{var}(\Gamma'_0, \Gamma_2, B) \subseteq \mathsf{var}(S, \Lambda, \Gamma'_0, \Gamma_2, A \otimes B)$.
- Γ_1 is fully contained in Λ. This case is analogous to the one above, but now we have to induct on the derivation f' instead of f''.
- Γ_1 splits between Λ and Ω. This means that $\Gamma_1 = \Gamma'_1, \Gamma''_1$ and $\Lambda = \Gamma_0, \Gamma'_1$ and $\Omega = \Gamma''_1, \Gamma_2$, and Γ''_1 is non-empty. Then $f' : S \mid \Gamma_0, \Gamma'_1 \vdash A$ and $f'' : - \mid \Gamma''_1, \Gamma_2 \vdash B$. By induction on f' with the partition $\langle \Gamma_0, \Gamma'_1, [\,] \rangle$ and on f'' with the partition $\langle [\,], \Gamma''_1, \Gamma_2 \rangle$, respectively, we obtain
  - a partition $\langle \Delta_0, \dots, \Delta_n \rangle$ of Γ'_1 and a partition $\langle \Delta_{n+1}, \dots, \Delta_m \rangle$ of Γ''_1,
  - two lists of interpolant formulae $D_1, \dots, D_n$ and $D_{n+1}, \dots, D_m$,
  - derivations $g' : S \mid \Gamma_0, D_1, \dots, D_n \vdash A$ and $g'' : - \mid D_{n+1}, \dots, D_m, \Gamma_2 \vdash B$,
  - derivations $h'_i : - \mid \Delta_i \vdash D_i$, for $i \in [1, \dots, n]$, and derivations $h'_j : - \mid \Delta_j \vdash D_j$, for $j \in [n+1, \dots, m]$, such that
  - $\mathsf{var}(D_1, \dots, D_n) \subseteq \mathsf{var}(\Delta_1, \dots, \Delta_n) \cap \mathsf{var}(S, \Gamma_0, A)$ and $\mathsf{var}(D_{n+1}, \dots, D_m) \subseteq \mathsf{var}(\Delta_{n+1}, \dots, \Delta_m)\mathsf{var}(\Gamma_2, B)$.

  The desired partition of Γ'_1, Γ''_1 is $\langle \Delta_0, \dots, \Delta_n, \Delta_{n+1}, \dots, \Delta_m \rangle$. The desired list of interpolant formulae is $D_1, \dots, D_n, D_{n+1}, \dots, D_m$. The desired derivation g is

$$\dfrac{\overset{g'}{S \mid \Gamma_0, D_1, \dots, D_n \vdash A} \quad \overset{g''}{- \mid D_{n+1}, \dots, D_m, \Gamma_2 \vdash B}}{S \mid \Gamma_0, D_1, \dots, D_n, D_{n+1}, \dots, D_m, \Gamma_2 \vdash A \otimes B} \otimes\mathsf{R}$$

  while the desired derivation h_i is h'_i for $i \in [1, \dots, m]$. For the variable condition, we have $\mathsf{var}(D_1, \dots, D_m) \subseteq \mathsf{var}(S, \Gamma_0, A, \Gamma_2, B) = \mathsf{var}(S, \Gamma_0, \Gamma_2, A \otimes B)$.

□

Notice that cMMIP is invoked in the proof of sMIP, in the case $f = \multimap\mathsf{L}(f', f'')$. Conversely, sMIP is invoked in the proof of cMMIP, in the case $f = \mathsf{pass}\ f'$. The proof of Theorem 4 describes an effective procedure for building interpolant formulae and derivations. This procedure is terminating, since each recursive call happens on a derivation with height strictly smaller than the one of the derivation in input. This behaviour is further confirmed in our Agda formalization, where the inductive proof of sMIP/cMMIP is accepted by the proof assistant as terminating.

Example 5. *Let us illustrate the interpolation procedure on a simple example. We compute the stoup Maehara interpolant of the end-sequent in the derivation*

$$\dfrac{\dfrac{\dfrac{\dfrac{}{X \mid\ \vdash X}\,\mathsf{ax}}{- \mid X \vdash X}\,\mathsf{pass} \quad \dfrac{\dfrac{}{Y \mid\ \vdash Y}\,\mathsf{ax} \quad \dfrac{\dfrac{}{W \mid\ \vdash W}\,\mathsf{ax}}{- \mid W \vdash W}\,\mathsf{pass}}{Y \mid W \vdash Y \otimes W}\,\otimes\mathsf{R}}{\dfrac{X \multimap Y \mid X, W \vdash Y \otimes W}{- \mid X \multimap Y, X, W \vdash Y \otimes W}\,\mathsf{pass}}\,{\multimap}\mathsf{L} \quad \dfrac{}{Z \mid\ \vdash Z}\,\mathsf{ax}}{(Y \otimes W) \multimap Z \mid X \multimap Y, X, W \vdash Z}\,{\multimap}\mathsf{L} \tag{6}$$

with the partition $\langle [X \multimap Y], [X, W] \rangle$.
Following the procedure in the proof of Theorem 4, we are in the case when the last rule is ${\multimap}\mathsf{L}$ *and both lists in the partition* $\langle [X \multimap Y], [X, W] \rangle$ *move to the context of the left premise. This means that we need to apply the* cMMIP *procedure to the derivation* $\mathsf{pass}({\multimap}\mathsf{L}(\mathsf{pass}\ \mathsf{ax}, \otimes\mathsf{R}(\mathsf{ax}, \mathsf{pass}\ \mathsf{ax}))) : - \mid X \multimap Y, X, W \vdash Y \otimes W$ *(witnessing the left premise of* ${\multimap}\mathsf{L}$*) with the partition* $\langle [X \multimap Y], [X, W], [\ ] \rangle$. *This produces*

- *a partition* $\langle [X], [W] \rangle$ *of* $[X, W]$,
- *a list of interpolant formulae* $[X, W]$, *and*
- *derivations* $\mathsf{pass}({\multimap}\mathsf{L}(\mathsf{pass}\ \mathsf{ax}, \otimes\mathsf{R}(\mathsf{ax}, \mathsf{pass}\ \mathsf{ax}))) : - \mid X \multimap Y, X, W \vdash Y \otimes W$, $\mathsf{pass}\ \mathsf{ax} : - \mid X \vdash X$, *and* $\mathsf{pass}\ \mathsf{ax} : - \mid W \vdash W$,

satisfying the variable condition. Next, we need to apply the sMIP *procedure on the derivation* $\mathsf{ax} : Z \mid\ \vdash Z$ *with the partition* $\langle [\ ], [\ ] \rangle$ *which produces two derivations* $\mathsf{ax} : Z \mid\ \vdash Z$ *and* $\mathsf{ax} : Z \mid\ \vdash Z$. *Then we obtain the desired interpolant formula* $X \multimap (W \multimap Z)$ *and the desired derivations*

$$g \quad = \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{X \mid\ \vdash X}\,\mathsf{ax}}{- \mid X \vdash X}\,\mathsf{pass} \quad \dfrac{\dfrac{}{Y \mid\ \vdash Y}\,\mathsf{ax} \quad \dfrac{\dfrac{}{W \mid\ \vdash W}\,\mathsf{ax}}{- \mid W \vdash W}\,\mathsf{pass}}{Y \mid W \vdash Y \otimes W}\,\otimes\mathsf{R}}{\dfrac{X \multimap Y \mid X, W \vdash Y \otimes W}{- \mid X \multimap Y, X, W \vdash Y \otimes W}\,\mathsf{pass}}\,{\multimap}\mathsf{L} \quad \dfrac{}{Z \mid\ \vdash Z}\,\mathsf{ax}}{(Y \otimes W) \multimap Z \mid X \multimap Y, X, W \vdash Z}\,{\multimap}\mathsf{L}}{(Y \otimes W) \multimap Z \mid X \multimap Y, X \vdash W \multimap Z}\,{\multimap}\mathsf{R}}{(Y \otimes W) \multimap Z \mid X \multimap Y \vdash X \multimap (W \multimap Z)}\,{\multimap}\mathsf{R}$$

$$h \quad = \quad \dfrac{\dfrac{\dfrac{}{X \mid\ \vdash X}\,\mathsf{ax}}{- \mid X \vdash X}\,\mathsf{pass} \quad \dfrac{\dfrac{\dfrac{}{W \mid\ \vdash W}\,\mathsf{ax}}{- \mid W \vdash W}\,\mathsf{pass} \quad \dfrac{}{Z \mid\ \vdash Z}\,\mathsf{ax}}{W \multimap Z \mid W \vdash Z}\,{\multimap}\mathsf{L}}{X \multimap (W \multimap Z) \mid X, W \vdash Z}\,{\multimap}\mathsf{L}$$

Notice that this is crucially different from the result that Maehara's method would produce on the corresponding derivation in the associative Lambek calculus (with $\otimes$*).*

The translation of derivation ([6]) in the associative Lambek calculus is

$$\dfrac{\dfrac{\dfrac{}{X \vdash X}\,\mathsf{ax} \qquad \dfrac{\dfrac{}{Y \vdash Y}\,\mathsf{ax} \quad \dfrac{}{W \vdash W}\,\mathsf{ax}}{Y, W \vdash Y \otimes W}\,{\otimes}\mathsf{R}}{Y/X, X, W \vdash Y \otimes W}\,/\mathsf{L} \qquad \dfrac{}{Z \vdash Z}\,\mathsf{ax}}{Z/(Y \otimes W), Y/X, X, W \vdash Z}\,/\mathsf{L}$$

Using the Maehara interpolation procedure defined in [5], the resulting interpolant formula would be $Z/(X \otimes W)$. Again, X and Y can be tensored in the latter formula since the Lambek calculus admits a general left rule for $\otimes$.

We conclude this section showing how Craig interpolation follows from stoup Maehara interpolation.

Theorem 6. *For any formulae A and C, if $A \multimap C$ is provable in* `SkNMILL`*, then there exists a formula D such that both $A \multimap D$ and $D \multimap C$ are provable, and $\mathsf{var}(D) \subseteq \mathsf{var}(A) \cap \mathsf{var}(C)$.*

Proof. $A \multimap C$ being provable means that there is a derivation $f : - \mid \ \vdash A \multimap C$. By invertibility of the rule $\multimap\mathsf{R}$, we obtain a derivation $f' : - \mid A \vdash C$. Then by running the sMIP procedure on f' with the partition $\langle [A], [\,] \rangle$, we get

- a formula D,
- $g' : - \mid A \vdash D$,
- $h' : D \mid \ \vdash C$, and
- $\mathsf{var}(D) \subseteq \mathsf{var}(A) \cap \mathsf{var}(C)$.

The formulae $A \multimap D$ and $D \multimap C$ are proved by the derivations $\multimap\mathsf{R}\ g' : - \mid \ \vdash A \multimap D$ and $\multimap\mathsf{R}(\mathsf{pass}\ h') : - \mid \ \vdash D \multimap C$, respectively. The variable condition is satisfied automatically. ◻

6 Proof-Relevant Interpolation

So far we have established a procedure sMIP for effectively splitting a derivation $f : S \mid \Gamma_1, \Gamma_2 \vdash C$ in two derivations $g : S \mid \Gamma_1 \vdash D$ and $h : - \mid \Gamma_2 \vdash C$, with D being "minimal" in the sense of satisfying an appropriate variable condition. A natural question arises: what happens when we compose derivations g and h using the admissible scut rule? Intuition suggests that we should get back the original derivation f, at least modulo η-conversions and permutative conversions. This in fact what happens, and this section is dedicated to proving this result.

Analogously, the cMMIP procedure splits a derivation $f : S \mid \Gamma_0, \Gamma_1, \Gamma_2 \vdash C$ in a tuple of derivations $[h_i : - \mid \Delta_i \vdash D_i]_i$ and $g : S \mid \Gamma_0, D_1, \ldots, D_n, \Gamma_2 \vdash C$, with $D_1, \ldots, D_n$ satisfying an appropriate variable condition. If we compose $[h_i]$ and g using the admissible ccut^* rule, we get back the original derivation modulo $\overset{\circ}{=}$.

Similar questions have been considered by Čubrić [22] in the setting of intuitionistic propositional logic and by Saurin [23] for (extensions) of classical linear logic. They call *proof-relevant interpolation* the study of interpolation procedures in relationship to cut rules and equivalence of proofs, like our $\overset{\circ}{=}$. In particular, Čubrić and Saurin show that interpolation procedures are in a way "right inverses" of cut rules. Here we show the same for $\texttt{SkNMILL}$: the sMIP procedure is a right inverse of scut, while the cMMIP procedure is a right inverse of ccut^*.

Theorem 7.

(i) *Let $g : S \mid \Gamma_0 \vdash D$ and $h : D \mid \Gamma_1 \vdash C$ be the derivations obtained by applying the sMIP procedure on a derivation $f : S \mid \Gamma \vdash C$ with the partition $\langle \Gamma_0, \Gamma_1 \rangle$. Then* $\mathsf{scut}(g, h) \doteq f$.

(ii) *Let $g : S \mid \Gamma_0, D_1, \ldots, D_n, \Gamma_2 \vdash C$ and $h_i : - \mid \Delta_i \vdash D_i$ for $i \in [1, \ldots, n]$ be derivations obtained by applying the cMMIP procedure on a derivation $f : S \mid \Gamma \vdash C$ with the partition $\langle \Gamma_0, \Gamma_1, \Gamma_2 \rangle$. Then* $\mathsf{ccut}^*([h_i], g) \doteq f$.

Proof. Similar to the proof of Theorem 4, statements (i) and (ii) are proved by mutual induction on the structure of derivations. We focus on the proof of statement (i), since (ii) is proved in a similar manner. We refer the interested reader to our Agda formalization for all the technical details.

The proof relies on the computational behaviour of the admissible rules scut and ccut. The reader might want to consult our previous work [16, 19] for the explicit construction of the cut rules that we employ in this proof.

$\underline{\text{Case } f = \mathsf{ax}.}$ The goal reduces to $\mathsf{scut}(\mathsf{ax}, \mathsf{ax}) \doteq \mathsf{ax}$, which holds by definition of scut.

$\underline{\text{Case } f = \mathsf{IR}.}$ The goal reduces to $\mathsf{scut}(\mathsf{IR}, \mathsf{IL}\ \mathsf{IR}) \doteq \mathsf{IR}$, which holds by definition of scut.

$\underline{\text{Case } f = \mathsf{IL}\ f'.}$ The goal reduces to $\mathsf{scut}(\mathsf{IL}\ g', h') \doteq \mathsf{IL}\ f'$[2]. By definition of scut we have $\mathsf{scut}(\mathsf{IL}\ g', h') = \mathsf{IL}\ (\mathsf{scut}(g', h'))$. By inductive hypothesis on f' and congruence (which here allow us to rewrite under IL), the latter is $\doteq$-related to $\mathsf{IL}\ f'$.

$\underline{\text{Cases } f = \otimes\mathsf{L}\ f' \text{ and } f = \multimap\mathsf{R}\ f'.}$ Analogous to the previous case. Though the case of $\multimap\mathsf{R}$ requires an additional application of Proposition 3.

$\underline{\text{Case } f = \mathsf{pass}\ f'.}$ Two cases determined by whether Γ_0 is empty or not.

- In the first case, the goal reduces to $\mathsf{scut}(\mathsf{IR}, \mathsf{IL}(\mathsf{pass}\ f')) \doteq \mathsf{pass}\ f'$, which holds by definition of scut.
- In the second case, the goal reduces to $\mathsf{scut}(\mathsf{pass}\ g', h') \doteq \mathsf{pass}\ f'$. By definition of scut we have $\mathsf{scut}(\mathsf{pass}\ g', h') = \mathsf{pass}(\mathsf{scut}(g', h'))$. By inductive hypothesis on f' and congruence, the latter is $\doteq$-related to $\mathsf{pass}\ f'$.

$\underline{\text{Case } f = \otimes\mathsf{R}(f', f'').}$ Two case determined by whether Γ_0 is fully contained in the context of the left premise or not.

- In the first case, the goal reduces to $\mathsf{scut}(g', \otimes\mathsf{R}(h', f'')) \doteq \otimes\mathsf{R}(f', f'')$. By Proposition 3, we have $\mathsf{scut}(g', \otimes\mathsf{R}(h', f'')) \doteq \otimes\mathsf{R}(\mathsf{scut}(g', h'), f'')$. By inductive hypothesis on f' and congruence, the latter is $\doteq$-related to $\otimes\mathsf{R}(f', f'')$.
- In the second case, the goal reduces to showing that the derivation $\mathsf{scut}(\otimes\mathsf{R}(g, h'), \otimes\mathsf{L}\ (\otimes\mathsf{R}(h', \mathsf{pass}`h'')))$ is $\doteq$-related to $\otimes\mathsf{R}(f', f'')$. This is witnessed by the following sequence of equivalences:

$$\begin{aligned}
&\mathsf{scut}(\otimes\mathsf{R}(g, h'), \otimes\mathsf{L}(\otimes\mathsf{R}(h', \mathsf{pass}`h''))) \\
&\quad = \mathsf{scut}(g', \otimes\mathsf{R}(h', \mathsf{scut}(g'', h''))) && \text{(by definition of } \mathsf{scut}) \\
&\quad \doteq \otimes\mathsf{R}(\mathsf{scut}(g', h'), \mathsf{scut}(g'', h'')) && \text{(by Proposition 3)} \\
&\quad \doteq \otimes\mathsf{R}(f', f'') && \text{(by ind. hyp. on } f' \text{ and } f'' \\
&&& \text{and congruence)}
\end{aligned}$$

[2]Here g' and h' are as in the proof of Theorem 4. We follow the same convention for the forthcoming cases too, where name of derivations will match the ones in the proof of Theorem 4

Case $f = \multimap\mathsf{L}(f', f'')$. Two case determined by whether Γ_1 is fully contained in the context of the right premise or not.

- In the first case, the goal reduces to $\mathsf{scut}(\multimap\mathsf{L}(f', g''), h'') \doteq \multimap\mathsf{L}(f', f'')$. By definition of scut, we have $\mathsf{scut}(\multimap\mathsf{L}(f', g''), h'') = \multimap\mathsf{L}(f', (\mathsf{scut}(g'', h'')))$. By inductive hypothesis on f'' and congruence, the latter is $\doteq$-related to $\multimap\mathsf{L}(f', f'')$.
- In the second case, the goal reduces to showing that the derivation $\mathsf{scut}(\multimap\mathsf{R}^*(\multimap\mathsf{L}(g', g'')), \multimap\mathsf{L}^*([h'_i], h''))$ is $\doteq$-related to $\multimap\mathsf{L}(f', f'')$. This is witnessed by the following sequence of equivalences:

$$\begin{aligned}
&\mathsf{scut}(\multimap\mathsf{R}^*(\multimap\mathsf{L}(g', g'')), \multimap\mathsf{L}^*([h'_i], h'')) \\
&\quad \doteq \mathsf{scut}(\mathsf{ccut}^*([h'_i], \multimap\mathsf{L}(g', g'')), h'') && \text{(by Proposition 2)} \\
&\quad = \multimap\mathsf{L}(\mathsf{ccut}^*([h'_i], g'), \mathsf{scut}(h'', g'')) && \text{(by definition of } \mathsf{scut} \text{ and } \mathsf{ccut}^*) \\
&\quad \doteq \multimap\mathsf{L}(f', f'') && \text{(by ind. hyp. on } f' \text{ and } f'' \\
& && \text{and congruence)}
\end{aligned}$$

  The final step employs the "inductive hypothesis" on f', which in this case means the validity of statement (ii) for derivation f' (remember that statements (i) and (ii) are proved simultaneously by structural induction on derivations).

□

7 Conclusions and Future Work

This paper describes a proof of Craig interpolation for the semi-substructural logic `SkNMILL`. It employs proof-theoretic techniques, since it relies on cut elimination and it manipulates sequent calculus derivations directly. As common when proving Craig interpolation for other substructural logics, the proof strategy follows a variant of Maehara's method [8]. `SkNMILL` is an intermediate logic between the $(\mathsf{I}, \otimes, /)$-fragment of non-commutative intuitionistic linear logic and its fragment without I and $\otimes$. Our proof of Craig interpolation clearly reflects this fact: we need to prove a stoup Maehara interpolation property (as in the Lambek calculus [9]) simultaneously with a context Maehara multi-interpolation property (similar to the one used for the product-free fragment [10]).

Following the category-theoretic considerations of Čubrić [22], we proved a proof-relevant form of interpolation, showing that the interpolation procedures are right inverses of the corresponding admissible cut rules. The main aspect missing in our work (and, as far as we know, the whole literature on Craig interpolation) is the characterization of the Craig interpolant via a universal property, in the sense of category theory. In fact, Čubrić's characterization of interpolants is merely an *existence* property, there is no mention of a *uniqueness* property. This is analogous to the distinction between *weak* and *non-weak* (co)limits in category theory. Alternatively, we may ask whether the interpolation procedures are also *left* inverses of the cut rules.

These questions naturally lead to another one: what is the correct notion of "equality" between interpolants? First, notice that in `SkNMILL` interpolants satisfying sMIP for a fixed sequent $S \mid \Gamma \vdash C$ and partition $\langle \Gamma_1, \Gamma_2 \rangle$ of Γ can be organized in a set of triples:

$$\{(D, g : S \mid \Gamma_1 \vdash D, h : D \mid \Gamma_2 \vdash C) \ \mid \ \mathsf{var}(D) \subseteq \mathsf{var}(S, \Gamma_0) \cap \mathsf{var}(\Gamma_1, C)\}$$

Many equivalence relation can potentially be defined on these triples:

- $(D, g, h) \sim (D', g', h')$ if and only if $D = D'$, $g = g'$ and $h = h'$. This option is definitely too strict, since it does not account for the fact that we consider derivations that differ by some η-conversion or permutative conversion as equal. A better option would then be:
- $(D, g, h) \sim (D', g', h')$ if and only if $D = D'$, $g \doteq g'$ and $h \doteq h'$. This might still be too restrictive. For example, in the Lambek calculus, we can run Maehara's method on two derivations f and f' for the same sequent that only differ by a permutative conversion, and obtain different interpolant formulae D and D'. The same phenomenon could also happen in `SkNMILL`. A relaxation of this notion would then be:
- $(D, g, h) \sim (D', g', h')$ if and only if there is an isomorphism $d : D \mid \ \vdash D'$ such that $\mathsf{scut}(g, d) \doteq g'$ and $h \doteq \mathsf{scut}(d, h')$. By isomorphism here we mean that there exists a derivation $h^{-1} : D' \mid \ \vdash D$ such that $\mathsf{scut}(h, h') \doteq \mathsf{ax}$ and $\mathsf{scut}(h', h) \doteq \mathsf{ax}$. Yet another weaker option could be:
- $\sim$ is the equivalence relation generated by the relation: $(D, g, h) \sim_0 (D', g', h')$ if and only if there exists a derivation $d : D \mid \ \vdash D'$ such that $\mathsf{scut}(g, d) \doteq g'$ and $h \doteq \mathsf{scut}(d, h')$. More explicitly, triples (D, g, h) and (D', g', h') are $\sim$-related when there exists a list of formulae $D_1, \ldots, D_n$ and a "zigzag" of derivations like

  $$d_1 : D \mid \ \vdash D_1, \quad d_2 : D_2 \mid \ \vdash D_1, \quad d_3 : D_2 \mid \ \vdash D_3, \quad \ldots \quad d_n : D_n \mid \ \vdash D'$$

  such that, when appropriately composed with these d_i-s, g is $\doteq$-related to g' and h is $\doteq$-related to h'. At first sight, this might seem like a weird notion of equality between interpolants, but it is in fact a very natural one to require from a category-theoretic perspective, since it would be the one characterizing interpolants as some kind of colimit/coend.

The attentive reader might have noticed that, while we consider sets of derivations quotiented by the congruence relation $\doteq$, we do not prove that the interpolation procedures of Theorem 4 are well-defined wrt. $\doteq$, e.g. that sMIP sends $\doteq$-related derivations to the "same" triple (D, g, h). The reason for this omission comes again from that fact that we do not yet know what is the appropriate notion of "sameness" in this case.

We do not want to continue our speculations in this conclusive section and leave further investigations on this topic to future work. Our first step will be understanding the universal property of interpolants for logics in which the Maehara interpolation property is simpler than in `SkNMILL`, e.g. in the associative Lambek calculus. Another venue of future work would be the extension of the results of this paper to other semi-substructural logics, e.g. extensions with a notion of skew exchange [25] or additive connectives [20]. In the latter setting we might also ask whether the logic satisfies a uniform interpolation property in the sense of [26].

References

[1] Craig, W.: Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. Journal of Symbolic Logic **22**(3), 269–285 (1957) https://doi.org/10.2307/2963594

[2] Beth, E.W.: On padoa's method in the theory of definition. Indagationes Mathematicae (Proceedings) **56**, 330–339 (1953) https://doi.org/10.1016/s1385-7258(53)50042-3

[3] Henzinger, T.A., Jhala, R., Majumdar, R., McMillan, K.L.: Abstractions from proofs. ACM SIGPLAN Notices **39**(1), 232–244 (2004) https://doi.org/10.1145/982962.964021

[4] Lambek, J.: The mathematics of sentence structure. American Mathematical Monthly **65**(3), 154–170 (1958) https://doi.org/10.2307/2310058

[5] Moot, R., Retoré, C.: The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics. Springer, ??? (2012). https://doi.org/10.1007/978-3-642-31555-8

[6] Girard, J.: Linear logic. Theoretical Computer Science **50**, 1–102 (1987) https://doi.org/10.1016/0304-3975(87)90045-4

[7] Kihara, H., Ono, H.: Interpolation properties, beth definability properties and amalgamation properties for substructural logics. Journal of Logic and Computation **20**(4), 823–875 (2009) https://doi.org/10.1093/logcom/exn084

[8] Maehara, S.: Craig's interpolation theorem. Mathematics **12**(4), 235–237 (1961) https://doi.org/10.11429/sugaku1947.12.235

[9] Ono, H.: Proof-theoretic methods in nonclassical logic –an introduction. In: Theories of Types and Proofs, pp. 207–254. The Mathematical Society of Japan, ??? (1998). https://doi.org/10.2969/msjmemoirs/00201c060

[10] Pentus, M.: Product-free lambek calculus and context-free grammars. Journal of Symbolic Logic **62**(2), 648–660 (1997) https://doi.org/10.2307/2275553

[11] Kanazawa, M.: Computing interpolants in implicational logics. Annals of Pure and Applied Logic **142**(1-3), 125–201 (2006) https://doi.org/10.1016/J.APAL.2005.12.014

[12] Uustalu, T., Veltri, N., Wan, C.-S.: Proof theory of skew non-commutative mill. In: Indrzejczak, A., Zawidzki, M. (eds.) Proceedings of 10th International Conference on Non-classical Logics: Theory and Applications, NCL 2022. Electronic Proceedings in Theoretical Computer Science, vol. 358, pp. 118–135. Open Publishing Association, ??? (2022). https://doi.org/10.4204/eptcs.358.9

[13] Girard, J.-Y.: A new constructive logic: Classical logic. Mathematical Structures in Computer Science **1**(3), 255–296 (1991) https://doi.org/10.1017/s0960129500001328

[14] Szlachányi, K.: Skew-monoidal categories and bialgebroids. Advances in Mathematics **231**(3–4), 1694–1730 (2012) https://doi.org/10.1016/j.aim.2012.06.027

[15] Zeilberger, N.: A sequent calculus for a semi-associative law. Logical Methods in Computer Science **15**(1) (2019) https://doi.org/10.23638/lmcs-15(1:9)2019

[16] Uustalu, T., Veltri, N., Zeilberger, N.: The sequent calculus of skew monoidal categories. In: Casadio, C., Scott, P.J. (eds.) Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics. Outstanding Contributions to Logic, vol. 20, pp. 377–406. Springer, ??? (2021). https://doi.org/10.1007/978-3-030-66545-6_11

[17] Uustalu, T., Veltri, N., Zeilberger, N.: Deductive systems and coherence for skew prounital closed categories. In: Sacerdoti Coen, C., Tiu, A. (eds.) Proceedings of 15th Workshop on Logical Frameworks and Meta-Languages: Theory and Practice, LFMTP 2020. Electronic Proceedings in Theoretical Computer Science, vol. 332, pp. 35–53. Open Publishing Association, ??? (2021). https://doi.org/10.4204/eptcs.332.3

[18] Veltri, N.: Maximally multi-focused proofs for skew non-commutative MILL. In: Hansen, H.H., Scedrov, A., Queiroz, R.J.G.B. (eds.) Proceedings of 29th International Workshop on Logic, Language, Information, and Computation, WoLLIC 2023. Lecture Notes in Computer Science, vol. 13923, pp. 377–393. Springer, ??? (2023). https://doi.org/10.1007/978-3-031-39784-4_24

[19] Wan, C.-S.: Semi-substructural logics à la Lambek. In: Indrzejczak, A., Zawidzki, M. (eds.) Proceedings of 11th International Conference on Non-classical Logics: Theory and Applications, NCL 2024. Electronic Proceedings in Theoretical Computer Science. Open Publishing Association, ??? (2024). https://doi.org/10.4204/eptcs.?? . To appear.

[20] Veltri, N., Wan, C.-S.: Semi-substructural logics with additives. In: Temur Kutsia, D.M. Daniel Ventura, Morales, J.F. (eds.) Proceedings of 18th International Workshop on Logical and Semantic Frameworks, with Applications and 10th Workshop on Horn Clauses for Verification and Synthesis, LSFA/HCVS 2023. Electronic Proceedings in Theoretical Computer Science, pp. 63–80. Open Publishing Association, ??? (2023). https://doi.org/10.4204/eptcs.402.8

[21] Andreoli, J.-M.: Logic programming with focusing proofs in linear logic. Journal of Logic and Computation **2**(3), 297–347 (1992) https://doi.org/10.1093/logcom/2.3.297

[22] Čubrić, D.: Interpolation property for bicartesian closed categories. Archive for Mathematical Logic **33**(4), 291–319 (1994) https://doi.org/10.1007/bf01270628

[23] Saurin, A.: Interpolation as cut-introduction. manuscript (2024). Available at https://www.irif.fr/_media/users/saurin/pub/interpolation_as_cut_introduction.pdf

[24] Abrusci, V.M.: Non-commutative intuitionistic linear logic. Mathematical Logic Quarterly **36**(4), 297–318 (1990) https://doi.org/10.1002/malq.19900360405

[25] Veltri, N.: Coherence via focusing for symmetric skew monoidal categories. In: Silva, A., Wassermann, R., de Queiroz, R. (eds.) Proceedings of 27th International Workshop on Logic, Language, Information, and Computation, WoLLIC 2021. Lecture Notes in Computer Science, vol. 13028, pp. 184–200. Springer, ??? (2021). https://doi.org/10.1007/978-3-030-88853-4_12

[26] Alizaseh, M., Derakhshan, F., Ono, H.: Uniform interpolation in substructural logics. The Review of Symbolic Logic **7**(3), 455–483 (2014) https://doi.org/10.1017/S175502031400015X