Benchmarking Linear Logic Translations

Carlos Olarte
Universidade Federal do Rio Grande do Norte, Brazil
carlos.olarte@gmail.com

Valeria de Paiva
Nuance Communications, USA
valeria.depaiva@gmail.com

Elaine Pimentel*
Universidade Federal do Rio Grande do Norte, Brazil
elaine.pimentel@gmail.com

Giselle Reis†
Carnegie Mellon University, Qatar
giselle@cmu.edu

Benchmarking automated theorem proving (ATP) systems using standardized problem sets is a well-established method for measuring their performance. However, the availability of such libraries for non-classical logics is very limited. In this work we seek to start a discussion of benchmarks for Girard's linear logic and some of its variants. For some quick bootstrapping of the collection of problems, we use translations of the collection of Kleene's intuitionistic theorems in the traditional monograph *Introduction to Metamathematics*. We analyze four different translations of intuitionistic logic into linear logic and compare their proofs using linear logic based provers with focusing.

1 Introduction

Benchmarking automated theorem proving (ATP) systems using standardized problem sets is a well-established method for measuring their performance. However, the availability of such libraries for *non-classical* logics is very limited. For intuitionistic logic several small collections of formulas have been published and used for testing ATP systems and Raths, Otten and Kreitz [15] consolidated and extended these small sets to provide the ILTP Library `http://www.cs.uni-potsdam.de/ti/iltp/`. For modal systems there is at least the QMLTP library [18, 14].

In this paper, we aim to start the discussion for a similar benchmark for Girard's linear logic [6] and some of its variants. Linear logic is a substructural logic that is a refinement of classical and intuitionistic logic, combining the dualities of the former with many of the constructive properties of the latter. Ideas from linear logic have been influential in fields such as programming languages, game semantics, quantum physics, as well as linguistics, particularly because of its emphasis on resource-boundedness, duality, and interaction. In particular, linear logic has had an important role as a logical framework for specifying and reasoning about logical and computational systems (the list is long; some examples are [3, 11, 4, 13]). As a consequence, several provers have been built for linear logic for different purposes.[1] However, so far, there has been no discussion of the *efficiency* or *adequacy* of these provers. In this work we set up to construct a collection of propositional tests, and to verify their proofs, as a first approximation for the desired benchmark.

When designing a benchmark, one has to carefully decide on a set of formulas that is *meaningful* in, at least, three ways: (1) the formulas should be able to distinguish several different characteristics

*Olarte and Pimentel are supported by CAPES, CNPq and the project FWF START Y544-N23.

†This paper was made possible by grant NPRP 7-988-1-178 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

[1] Listing some: LLprover: `http://bach.istc.kobe-u.ac.jp/llprover/`, linTAP: `http://www.leancop.de/lintap/`, LL prover explorer: `https://github.com/andykitchen/linear-logic`, Lolli: `http://www.lix.polytechnique.fr/~dale/lolli/`, Alcove: `http://cic.puj.edu.co/~caolarte/alcove2/`.

Submitted to:
Linearity & TLLA 2018

of the logical systems and provers; (2) the set should contain important theorems and paradigmatic formulas (non necessarily provable); and (3) the set should be large enough, so to serve as a comparison for different provers and systems. In this work, we will not be much concerned about (3), but rather concentrate on (1) and (2), benchmarking translations of a set of intuitionistic formulas into linear logic.

It turns out that propositional linear logic (LL) has many aspects that need to be considered. For example, one could adopt its classical (CLL) or intuitionistic (ILL) versions. Hence one important task would be to determine the difference in provability between them, and this is already far from trivial. While it is possible to differentiate the syntax of formulas and the presentation of the inference rules by the standard restriction on the right context to having at most one formula in ILL[2], FILL [2] is a multiple-conclusion system with the same connectives and rules as in CLL, but restricting the form of the application of such rules. Restricting ourselves to formulas with the same syntax in classical and intuitionistic versions of LL, the first interesting question would be which formulas are provable in CLL but not in ILL. This is the same issue *e.g.* when building a benchmark for intuitionistic logic versus the existing ones for classical logic. But the linear case is far more complicated, since the lack of the structural rules of weakening and contraction in both ILL and CLL makes these systems "closer" to each other than in the case of classical and intuitionistic logics. Indeed, only very recently [8] the first conservativity results presented in [16] were generalized.

Another important aspect to be taken into consideration is *focusing* [1]. It turns out that both ILL and CLL admit complete focused proof systems, and provers can be built using proof search strategies based on this discipline, which reduces the proof search space. This has an immediate effect on the proposal of formulas composing a possible benchmark, since the amount of exponentials in a formula can make a significant difference on the performance of provers.

Finally, concerning (2), there is no consensus in the community on a set of "principal" theorems that should be used as a test for LL-based theorem provers. In this work, we will consider the translation of a fragment of Kleene's list for intuitionistic logic (IL). The first challenge is to understand *how* these intuitionistic theorems should be interpreted in LL. A first answer would be: use one of the well known translations of IL into LL. This naive approach has, at least, two problems. First, it is not adequate to elect *one* translation, since different translations have very different computational behaviors, as it will be clear in Section 2.2. Second, some translations would not give the best interpretation of linear logic formulas. As a simple example, $A \to A$ should most probably be translated as $A \multimap A$, without bangs since this is equivalent, as a theorem, to the identity. But *any* sound translation from IL to LL adds bangs to implicational formulas. Hence none of them would preserve the formula's interpretation.

We analyze four different translations from IL to LL using Kleene's collection of IL theorems. Provability and proof time of the 244 generated sequents are compared using our ILL prover based on focusing. Since one of the considered translations is not validity preserving, we propose provable versions (not following any systematic translation) for those 27 formulas. Finally, we add two CLL formulas, not provable in ILL. The whole set will not only provide some interesting insights on different behaviors of LL formulas coming from different translations in the literature, but also present a significative set of 273 formulas for benchmarking linear logic based provers.

Outline. The rest of the paper is organized as follows. Section 2 presents LL, focusing, translations and decorations; Section 3 presents Kleene's list and their linearization; Section 4 concludes the paper and presents some future research directions.

[2] We note that in the literature there are two versions of ILL, having *at most* or *exactly* one formula in the right context. This is similar to the problem of considering intuitionistic/minimal logics. Since in this work we will use a multiplicative fragment of ILL, we opted for the version of ILL having $\bot$ in the grammar.

2 Linear Logic

Although we assume that the reader is familiar with linear logic, we review some of its basic proof theory (see [17] for more details).

Linear logic is a substructural logic proposed by Girard [6] as a refinement of classical and intuitionistic logic, joining the dualities of the former with many of the constructive properties of the latter. Formulas for propositional linear logic (LL) are built from the following grammar

$$F ::= p \mid 1 \mid 0 \mid \top \mid \bot \mid F \otimes F \mid F \parr F \mid F \,\&\, F \mid F \oplus F \mid F \multimap F \mid ?F \mid !F$$

where atomic formulas p or their negations $p^{\perp} = p \multimap \bot$ are called *literals*. The logical connectives for LL can be divided into the following groups: the *multiplicative* version of conjunction, true, disjunction, and false, which are written as $\otimes$, 1, $\parr$, $\bot$, respectively; and the *additive* version of these connectives, which are written as &, $\top$, $\oplus$, 0, respectively; and the *exponentials* ! and ?. LL *sequents* have the form $\Gamma \vdash \Delta$ where Γ, Δ are multisets of formulas. We will consider first the two sided sequent formulation of classical linear logic (presented in Figure 2), to be able to smoothly extend the discussion to the intuitionistic case. We recall that contraction and weakening of formulas are controlled using the exponentials and rules $\mathsf{cont}_R, \mathsf{cont}_L, \mathsf{weak}_R, \mathsf{weak}_L$. The rules of ILL are depicted in Figure 3.

2.1 Focusing

Andreoli introduced in [1] a notion of normal form for cut-free derivations in linear logic. The connectives of LL can be divided into two classes: *negative* ($\parr$, $\bot$, &, $\top$, and ?) and *positive* ($\otimes$, 1, $\oplus$, 0, and !). Note that the dual of a negative connective is positive and vice-versa. In general, the introduction rules for negative connectives are all invertible, meaning that the conclusion of any of these introduction rules is equivalent to its premises. The introduction rules for the positive connectives are not necessarily invertible and may require a choice or a context restriction on the application of rules. The notions of negative and positive polarities are extended to formulas in the natural way by considering the outermost connective, *e.g.*, $A \otimes B$ is positive while $A \parr B$ is negative. Any bias can be assigned to atomic formulas.

A focused proof is organized around two "phases" of proof construction: the *negative phase* for introducing negative connectives and the *positive phase* for the positive connectives. In the focusing discipline, negative formulae are decomposed eagerly until only positive formulae are left, then one of them is non-deterministically chosen to be focused on. Figures 4 and 5 present the systems LLF and ILLF, the focused versions of CLL and ILL, respectively.

2.2 Translations and Decorations

A naive approach for building a set of test formulas for LL based provers would be to use one of the well known translations of intuitionistic (or classical) formulas into LL. Since there are several ways of translating a formula from IL to LL, we asked ourselves which one would be the best option, if any. Each translation characterizes a different linear view of intuitionistic formulas and it is interesting and relevant to establish a comparison between them. We analyze four different translations: a multiplicative translation, the so called Girard's translation, Girard's positive translation and Miller and Liang's $0/1$ translation.

The multiplicative translation trivially substitutes the intuitionistic connectives by their multiplicative linear version

$$\begin{array}{rclrclrcl} (p)^m & = & p & (A \to B)^m & = & A^m \multimap B^m & (A \wedge B)^m & = & A^m \otimes B^m \\ (t)^m & = & 1 & (A \vee B)^m & = & A^m \parr B^m & (f)^m & = & \bot \end{array}$$

Translation of sequents is given by $(\Gamma \vdash A)^m = \Gamma^m \vdash A^m$. Observe that this translation *does not* preserve provability: for instance, diagonals $A \otimes A \to A$ exist in IL, but not in LL.

Girard's translation [6], also known as *call-by-name*, is the most well known translation of IL into LL.

$$\begin{array}{rclrclrcl} (p)^g & = & p & (A \to B)^g & = & !A^g \multimap B^g & (A \wedge B)^g & = & A^g \,\&\, B^g \\ (t)^g & = & \top & (A \vee B)^g & = & !A^g \oplus !B^g & (f)^g & = & 0 \end{array}$$

Sequents are translated as $(\Gamma \vdash A)^g = !\Gamma^g \vdash A^g$. Girard's translation preserves provability but is not a *decoration* in the sense of [5], namely, a proof of A in IL is transformed into a proof of A^g in LL which is not isomorphic to the original one.

Girard proposed in the same paper [6] another translation, known as *call-by-value*. Henceforth, we will call this translation *positive*, since IL formulas become positive LL formulas.

$$\begin{array}{rclrclrcl} (p)^p & = & !\,p & (A \to B)^p & = & !(A^p \multimap B^p) & (A \wedge B)^p & = & A^p \otimes B^p \\ (t)^p & = & 1 & (A \vee B)^p & = & A^p \oplus B^p & (f)^p & = & 0 \end{array}$$

Sequents are translated as $(\Gamma \vdash A)^p = \Gamma^p \vdash A^p$. It is easy to see that the positive translation is a decoration: proofs of A^p in LL are isomorphic to proofs of A in IL.

Another interesting translation is the $0/1$ translation [9], which distinguishes the polarity of formulas in a sequent.

$$\begin{array}{rclrclrcl} (p)^0 & = & p & (A \to B)^0 & = & !A^1 \multimap !B^0 & (A \wedge B)^0 & = & !A^0 \,\&\, !B^0 \\ (t)^0 & = & \top & (A \vee B)^0 & = & !A^0 \oplus !B^0 & (f)^0 & = & 0 \end{array}$$

$$\begin{array}{rclrclrcl} (p)^1 & = & p & (A \to B)^1 & = & !(!A^0 \multimap B^1) & (A \wedge B)^1 & = & !(A^1 \,\&\, B^1) \\ (t)^1 & = & 1 & (A \vee B)^1 & = & !A^1 \oplus !B^1 & (f)^1 & = & 0 \end{array}$$

The translation of sequents is given by $(\Gamma \vdash A)^{0/1} = !\Gamma^0 \vdash A^1$. Using this translation, *focused proofs* in LLF are in bijetive correspondence with proofs in IL. In a loose sense, this can be considered a decoration if the isomorphism is interpreted "modulo focusing". In the focusing context, this is referred to as *adequacy on the level of derivations* [12].

Basically these four translations differ on their use of bangs and their polarization of atoms. The multiplicative translation introduces no bangs; Girard's translation forces atoms to have negative polarity and backchaining proofs; the positive translation selects the global preference to be forward-chaining and all atoms have positive polarity; the $0/1$ translation is asymmetric, and it does not impose any restriction on atoms. We will show an interesting comparison of the implementation of these translations in Section 4.

3 Kleene's Examples and their Linearization

Kleene's "Introduction to Metamathematics" [7] has a collection of interesting intuitionistic theorems. They are rather straightforward, thus they would not be especially useful for testing the efficiency of a prover. Instead, they can be regarded as a minimal set of intuitionistic theorems that a *sound* prover should be able to derive. As such, they can be valuable in uncovering bugs and sources of unsoundness. Our goal is to set up a similar set for LL.

The first challenge is to understand how these intuitionistic theorems should be interpreted in LL. Deciding whether to translate intuitionistic disjunction as the multiplicative disjunction $\parr$ of Linear

Logic or the additive disjunction $\oplus$ changes the target system under consideration, thus we prefer to not consider the intuitionistic disjunction to begin with. Hence we will consider what we call the *rudimentary fragment of* IL, which is very well-behaved. Semantically this fragment corresponds to cartesian closed categories.

The following 61 theorems are collected from [7], from page 113 onwards, and contain only the $(\rightarrow, \wedge)$ fragment. The bi-implication is defined as $A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$.

1. $\vdash A \rightarrow A$
2. $A \rightarrow B, B \rightarrow C \vdash A \rightarrow C$
3. $A \rightarrow (B \rightarrow C) \vdash B \rightarrow (A \rightarrow C)$
4. $A \rightarrow (B \rightarrow C) \vdash A \wedge B \rightarrow C$
5. $A \wedge B \rightarrow C \vdash A \rightarrow (B \rightarrow C)$
6. $A \rightarrow B \vdash (B \rightarrow C) \rightarrow (A \rightarrow C)$
7. $A \rightarrow B \vdash (C \rightarrow A) \rightarrow (C \rightarrow B)$
8. $A \rightarrow B \vdash A \wedge C \rightarrow B \wedge C$
9. $A \rightarrow B \vdash C \wedge A \rightarrow C \wedge B$
10. $\neg A \vdash A \rightarrow B$
11. $A \vdash \neg A \rightarrow B$
12. $B \vdash A \rightarrow B$
13. $A \rightarrow B \vdash \neg B \rightarrow \neg A$
14. $A \rightarrow \neg B \vdash \neg\neg B \rightarrow \neg A$
15. $A \rightarrow B, B \rightarrow A \vdash A \leftrightarrow B$
16. $A \leftrightarrow B \vdash A \rightarrow B$
17. $A \leftrightarrow B \vdash B \rightarrow A$
18. $A \leftrightarrow B, A \vdash B$
19. $A \leftrightarrow B, B \vdash A$
20. $\vdash A \leftrightarrow A$
21. $A \leftrightarrow B \vdash B \leftrightarrow A$
22. $A \leftrightarrow B, B \leftrightarrow C \vdash A \leftrightarrow C$
23. $A \rightarrow (B \rightarrow C), \neg\neg A, \neg\neg B \vdash \neg\neg C$
24. $\neg\neg(A \rightarrow B) \vdash \neg\neg A \rightarrow \neg\neg B$
25. $\neg\neg(A \rightarrow B), \neg\neg(B \rightarrow C) \vdash \neg\neg(A \rightarrow C)$
26. $\vdash \neg\neg(A \wedge B) \leftrightarrow (\neg\neg A \wedge \neg\neg B)$
27. $\vdash \neg\neg(A \leftrightarrow B) \leftrightarrow (\neg\neg(A \rightarrow B) \wedge \neg\neg(B \rightarrow A))$
28. $A \leftrightarrow B \vdash (A \rightarrow C) \leftrightarrow (B \rightarrow C)$
29. $A \leftrightarrow B \vdash (C \rightarrow A) \leftrightarrow (C \rightarrow B)$
30. $A \leftrightarrow B \vdash (A \wedge C) \leftrightarrow (B \wedge C)$
31. $A \leftrightarrow B \vdash (C \wedge A) \leftrightarrow (C \wedge B)$
32. $A \leftrightarrow B \vdash \neg A \leftrightarrow \neg B$
33. $\vdash ((A \wedge B) \wedge C) \leftrightarrow (A \wedge (B \wedge C))$
34. $\vdash (A \wedge B) \leftrightarrow (B \wedge A)$
35. $\vdash (A \wedge A) \leftrightarrow A$
36. $A \vdash (A \rightarrow B) \leftrightarrow B$
37. $B \vdash (A \rightarrow B) \leftrightarrow B$
38. $\neg A \vdash (A \rightarrow B) \leftrightarrow \neg A$
39. $\neg B \vdash (A \rightarrow B) \leftrightarrow \neg A$
40. $B \vdash (A \wedge B) \leftrightarrow A$
41. $\neg B \vdash (A \wedge B) \leftrightarrow B$
42. $\vdash A \rightarrow \neg\neg A$
43. $\vdash \neg\neg\neg A \leftrightarrow \neg A$
44. $\vdash \neg(A \wedge \neg A)$
45. $\vdash \neg(A \leftrightarrow \neg A)$
46. $\vdash \neg\neg(\neg\neg A \rightarrow A)$
47. $\vdash (A \wedge (B \wedge \neg B)) \leftrightarrow (B \wedge \neg B)$
48. $\vdash (A \rightarrow B) \rightarrow \neg(A \wedge \neg B)$
49. $\vdash (A \rightarrow \neg B) \leftrightarrow (\neg(A \wedge B))$
50. $\vdash (\neg(A \wedge B)) \leftrightarrow (\neg\neg A \rightarrow \neg B)$
51. $\neg\neg B \rightarrow B \vdash (\neg\neg A \rightarrow B) \leftrightarrow (A \rightarrow B)$
52. $\neg\neg B \rightarrow B \vdash (A \rightarrow B) \leftrightarrow (\neg(A \wedge \neg B))$
53. $\vdash (\neg\neg A \rightarrow B) \rightarrow \neg(A \wedge \neg B)$
54. $\vdash (A \wedge B) \rightarrow \neg(A \rightarrow \neg B)$
55. $\vdash (A \wedge \neg B) \rightarrow \neg(A \rightarrow B)$
56. $\vdash \neg\neg A \wedge B \rightarrow \neg(A \rightarrow \neg B)$
57. $\vdash (\neg\neg A \wedge \neg B) \leftrightarrow \neg(A \rightarrow B)$
58. $\vdash \neg(A \rightarrow B) \leftrightarrow \neg\neg(A \wedge \neg B)$
59. $\vdash \neg\neg(A \rightarrow B) \leftrightarrow \neg(A \wedge \neg B)$
60. $\vdash \neg(A \wedge \neg B) \leftrightarrow (A \rightarrow \neg\neg B)$
61. $\vdash (A \rightarrow \neg\neg B) \leftrightarrow (\neg\neg A \rightarrow \neg\neg B)$

3.1 Tests

We specified in rewriting logic (RW, see e.g., [10]) and implemented in Maude (`http://maude.cs.uiuc.edu`) a very basic prover for IL as well as for ILLF and LLF. The use of RW leads to a clear separation between deterministic inference rules that can be eagerly applied (as those in the negative phase) and non-deterministic inference rules where backtracking may be needed (as those in the positive phase). Moreover, the minimal distance between the represented logic (IL, ILLF and LLF) and its specification in RW allowed us to quickly implement a good prototypical tool useful for our tests. Although more efficient provers can be built by e.g., including sophisticated heuristics and specialized

data structures, our prototypical implementations were enough to compare the different translations. We have implemented the proofs of the original IL sequent, together with the derivation tree of each of the corresponding ILL sequents, when provable. The results are summarized in Figure 1 (note that some provable sequents timed out). The code in Maude, the pdf file containing all the proofs and the list of formulas used here in a format similar to that of the TPTP Problem Library can be found at `https://github.com/carlosolarte/Benchmarking-Linear-Logic`.

Applying each translation defined in Section 2.2 to each of the 61 sequents presented in the last section gives rise to 244 different LL sequents. As already noted, provability is not preserved in the multiplicative translation. The reason for that, other than the obvious absence of structural rules in the left context, is that the multiplicative false $\bot$ is relevant, so while $0 \vdash B$ for any B in IL, $A \otimes (A \multimap \bot) \nvdash B$ in LL. The other three translations fix this by systematically adding bangs and additive connectives. This procedure, as seen in the tests, does not give the best translation to formulas for LL provers.

We present below an alternative description for the 27 the sequents not provable in ILL (see Appendix B for the whole list) for the multiplicative version of the Kleene sequents, with a small set of bangs and/or additives.

10. $A \multimap 0 \vdash A \multimap B$
11. $A \vdash (A \multimap 0) \multimap B$
12. $B \vdash\, !A \multimap B$
16. $(A \multimap B) \otimes\, !(B \multimap A) \vdash A \multimap B$
17. $\,!(A \multimap B) \otimes (B \multimap A) \vdash B \multimap A$
18. $A \multimap\!\circ\, B, A \vdash B \otimes (B \multimap A)$
19. $A \multimap\!\circ\, B, B \vdash A \otimes (A \multimap B)$
26. a. $\vdash ((A \,\&\, B)^{\bot\bot}) \multimap (A^{\bot\bot} \,\&\, B^{\bot\bot})$
    b. $\vdash (A^{\bot\bot} \otimes B^{\bot\bot}) \multimap (A \otimes B)^{\bot\bot}$
27. a. $\vdash (\,!(A \multimap B) \otimes\, !(B \multimap A))^{\bot\bot} \multimap [(A \multimap B)^{\bot\bot} \,\&\, (B \multimap A)^{\bot\bot}]$
    b. $\vdash (A \multimap B)^{\bot\bot} \otimes (B \multimap A)^{\bot\bot} \multimap (A \multimap\!\circ\, B)^{\bot\bot}$
35. $\vdash (\,!A \otimes\, !A) \multimap\!\circ\, !A$
36. $A \vdash ((A \multimap B) \multimap B) \otimes (B \multimap (\,!A \multimap B))$
37. $B \vdash (\,!(A \multimap B) \multimap B) \otimes (B \multimap (\,!A \multimap B))$
38. $A^{\bot} \vdash (\,!(A \multimap B) \multimap A^{\bot}) \otimes ((A \multimap 0) \multimap (A \multimap B))$
39. $B \multimap 0 \vdash (A \multimap B) \multimap\!\circ\, (A \multimap 0)$
40. $B \vdash ((A \otimes\, !B) \multimap A) \otimes (A \multimap (A \otimes B))$
41. $B \multimap 0 \vdash ((\,!A \otimes B) \multimap B) \otimes (B \multimap (A \otimes B))$
45. $\vdash (\,!(A \multimap A^{\bot}) \otimes ((\,!A)^{\bot} \multimap\, !A))^{\bot}$
46. $\vdash (\,!(\,!(A^{\bot} \multimap 0)) \multimap A)^{\bot})^{\bot}$
47. $\vdash A \otimes (B \otimes (B \multimap 0)) \multimap\!\circ\, (B \otimes (B \multimap 0))$
57. a. $\vdash (A^{\bot\bot} \otimes B^{\bot}) \multimap (A \multimap B)^{\bot}$
    b. $\vdash (\,!A \multimap B)^{\bot} \multimap ((A \multimap 0)^{\bot} \,\&\, B^{\bot})$
58. a. $\vdash\, !((\,!A \multimap B)^{\bot}) \multimap ((A \otimes B^{\bot}) \multimap 0)^{\bot}$
    b. $\vdash (A \otimes B^{\bot})^{\bot\bot} \multimap (A \multimap B)^{\bot}$
59. a. $\vdash (A \multimap B)^{\bot\bot} \multimap (A \otimes B^{\bot})^{\bot}$
    b. $\vdash ((A \otimes B^{\bot}) \multimap 0) \multimap (\,!(\,!A \multimap B)^{\bot})^{\bot}$

As a "bonus", there are two more formulas we think are of interest when planning a benchmark for LL: the classical linear version of Pierce's Law and a minimal example of a non-provable ILL formula *without bottom* that is a CLL theorem [8].

62. $\vdash ((A \multimap ?B) \multimap A) \multimap ?A$
63. $\vdash (((A \otimes \top) \,\&\, (B \otimes \top)) \multimap 0) \multimap ((A \multimap C) \oplus (B \multimap D))$

4 Conclusion

In this work we benchmarked different translations from IL into ILL, having as a result an initial set of formulas for benchmarking linear logic based provers. Starting with the $(\rightarrow, \wedge)$-fragment of Kleene's theorems, we generated 244 different ILL sequents using 4 automatic translations: *multiplicative*, *Girard's*, *positive* and *0/1*. The first translation is the only one that does not preserve provability. For each of those 27 ILL sequents that are not provable via the multiplicative translation, we proposed an alternative provable sequent with a small set of additives and bangs added. This makes these particular sequents

#	LJ	m	g	p	0/1
1	31	46	43	83	59
2	48	111	168	226	185
3	49	94	128	208	189
4	48	97	326	219	313
5	38	98	108	182	197
6	48	106	157	252	214
7	51	108	156	261	201
8	34	91	114	191	271
9	35	89	111	173	257
10	34	19 (x)	79	98	98
11	34	18 (x)	82	97	97
12	19	20 (x)	42	74	59
13	47	89	137	184	180
14	62	96	186	297	757
15	48	156	181	527	331
16	34	19 (x)	91	139	137
17	35	21(x)	94	132	142
18	34	24 (x)	97	115	114
19	35	20 (x)	92	113	115
20	18	90	43	126	108

#	LJ	m	g	p	0/1
21	54	160	196	515	1825
22	150	228	◷	◷	◷
23	2318	164	◷	◷	◷
24	211	142	4694	5522	◷
25	4063	202	◷	◷	◷
26	140	20 (x)	27482	◷	◷
27	◷	23 (x)	◷	◷	◷
28	86	227	◷	◷	◷
29	86	240	◷	◷	◷
30	48	191	292	3424	◷
31	54	209	353	3752	◷
32	83	202	12683	◷	◷
33	21	166	123	281	609
34	18	131	81	217	276
35	21	18 (x)	50	180	153
36	35	19 (x)	107	209	184
37	18	19 (x)	67	151	139
38	54	21 (x)	157	333	319
39	67	22 (x)	271	595	626
40	21	18 (x)	67	165	178

#	LJ	m	g	p	0/1
41	33	21 (x)	130	172	209
42	41	61	83	120	122
43	96	183	271	528	7431
44	34	59	88	102	141
45	96	19 (x)	◷	5241	◷
46	66	25 (x)	185	250	427
47	61	19 (x)	234	186	875
48	48	94	146	181	370
49	66	161	204	535	46292
50	94	245	2618	18580	◷
51	882	295	◷	◷	◷
52	112	257	◷	◷	◷
53	67	126	255	335	14764
54	49	74	115	170	268
55	49	92	136	187	345
56	64	97	181	253	3946
57	385	20 (x)	◷	◷	◷
58	118	20 (x)	◷	◷	◷
59	168	20 (x)	◷	◷	◷
60	96	214	4004	8427	◷
61	9785	288	◷	◷	◷

Figure 1: Comparison of translations: x indicates that the formula is not provable; ◷ indicates timeout (over 60 seconds). Times are measured in **miliseconds**.

amenable to the use of all the power of focusing theorem provers. In fact, the excess of bangs in formulas tends to neutralize the efficacy of focusing, due to the positive/negative behavior of the exponentials. To emphasize the crucial differences between ILL and CLL we added the classical linear version of Pierce's Law and a minimal counter-example of conservativity from ILL to CLL. Thus our initial proposal for a suitable benchmark for ILL has 273 formulas, testing aspects like provability and focusing.

It is worth noticing that (1) we include $\bot$ in the grammar of ILL; (2) all the sequents of our collection can also serve as tests in CLL. The decision in (1) was motivated by the fact that the resulting sequents fall into the multiplicative fragment of ILL. But observe that one could clearly exchange $\bot$ for 0 in the multiplicative translation (that would not be multiplicative anymore) and still obtain a significant set of 23 formulas not provable via this new translation. The same will happen in (2), since some sequents involving double negations become provable.

For a first experiment with the proposed set of sequents, we have implemented provers for LJ, ILL and CLL. All three are focused-based, but bear in mind that the LJ prover does not have positive phases, it is only doing the invertible part of the proof eagerly. The results presented in Figure 1 serve as an initial comparison between the different translations chosen for generating our set of sequents, not for comparing different linear logic provers.

For future work, we intend to collect some more test-formulas, specially those involving disjunction, and to test different provers already available online.

References

[1] Jean-Marc Andreoli (1992): *Logic Programming with Focusing Proofs in Linear Logic*. Journal of Logic and Computation 2(3), pp. 297–347.

[2] Torben Braüner & Valeria de Paiva (1996): *Cut-Elimination for Full Intuitionistic Linear Logic*. Technical Report, BRICS Report Series.

[3] Iliano Cervesato & Frank Pfenning (2002): *A Linear Logical Framework*. Information & Computation 179(1), pp. 19–75.

[4] Kaustuv Chaudhuri & Giselle Reis (2015): *An Adequate Compositional Encoding of Bigraph Structure in Linear Logic with Subexponentials*. In: *LPAR-20*, pp. 146–161.

[5] Vincent Danos, Jean-Baptiste Joinet & Harold Schellinx (1995): *On the linear decoration of intuitionistic derivations*. Arch. Math. Log. 33(6), pp. 387–412, doi:10.1007/BF02390456.

[6] Jean-Yves Girard (1987): *Linear Logic*. Theoretical Computer Science 50, pp. 1–102.

[7] S. Kleene (1952): *Introduction to Metamathematics*.

[8] Olivier Laurent (2018): *Around Classical and Intuitionistic Linear Logics*. In: *33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*.

[9] Chuck Liang & Dale Miller (2009): *Focusing and polarization in linear, intuitionistic, and classical logics*. Theor. Comput. Sci. 410(46), pp. 4747–4768, doi:10.1016/j.tcs.2009.07.041.

[10] José Meseguer (2012): *Twenty years of rewriting logic*. J. Log. Algebr. Program. 81(7-8), pp. 721–781.

[11] Dale Miller & Elaine Pimentel (2013): *A formal framework for specifying sequent calculus proof systems*. Theor. Comput. Sci. 474, pp. 98–116.

[12] Vivek Nigam & Dale Miller (2010): *A Framework for Proof Systems*. Journal of Automated Reasoning 45(2), pp. 157–188, doi:10.1007/s10817-010-9182-1.

[13] Carlos Olarte & Elaine Pimentel (2017): *On concurrent behaviors and focusing in linear logic*. Theor. Comput. Sci. 685, pp. 46–64, doi:10.1016/j.tcs.2016.08.026.

[14] Thomas Raths & Jens Otten (2012): *The QMLTP problem library for first-order modal logics*. In: *International Joint Conference on Automated Reasoning*, pp. 454–461.

[15] Thomas Raths, Jens Otten & Christoph Kreitz (2007): *The ILTP problem library for intuitionistic logic*. Journal of Automated Reasoning 38(1-3), pp. 261–271.

[16] Harold Schellinx (1991): *Some Syntactical Observations on Linear Logic*. J. Log. Comput. 1(4), pp. 537–559, doi:10.1093/logcom/1.4.537.

[17] Anne S. Troelstra (1992): *Lectures on Linear Logic*. CSLI Lecture Notes 29, Center for the Study of Language and Information, Stanford, California.

[18] Max Wisniewski, Alexander Steen & Christoph Benzmüller (2016): *TPTP and Beyond: Representation of Quantified Non-Classical Logics*. In Christoph Benzmüller & Jens Otten, editors: *ARQNL 2016. Automated Reasoning in Quantified Non-Classical Logics*.

A Some sequent systems

$$\frac{}{p \vdash p}\ \mathsf{init} \qquad \frac{}{\vdash 1}\ 1_R \qquad \frac{}{\Gamma \vdash \Delta, \top}\ \top_R \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \bot}\ \bot_R \qquad \frac{\Gamma \vdash \Delta}{\Gamma, 1 \vdash \Delta}\ 1_L \qquad \frac{}{\bot \vdash}\ \bot_L \qquad \frac{}{\Gamma, 0 \vdash \Delta}\ 0_L$$

$$\frac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, F \parr G}\ \parr_R \qquad \frac{\Gamma \vdash \Delta, F \quad \Gamma \vdash \Delta, G}{\Gamma \vdash \Delta, F \,\&\, G}\ \&_R \qquad \frac{\Gamma_1 \vdash \Delta_1, F \quad \Gamma_2 \vdash \Delta_2, G}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, F \otimes G}\ \otimes_R \qquad \frac{\Gamma \vdash \Delta, F_i}{\Gamma \vdash \Delta, F_1 \oplus F_2}\ \oplus_{R_i}$$

$$\frac{\Gamma_1, F \vdash \Delta_1 \quad \Gamma_2, G \vdash \Delta_2}{\Gamma_1, \Gamma_2, F \parr G \vdash \Delta_1, \Delta_2}\ \parr_L \qquad \frac{\Gamma, F_i \vdash \Delta}{\Gamma, F_1 \,\&\, F_2 \vdash \Delta}\ \&_{L_i} \qquad \frac{\Gamma, F, G \vdash \Delta}{\Gamma, F \otimes G \vdash \Delta}\ \otimes_L \qquad \frac{\Gamma, F \vdash \Delta \quad \Gamma, G \vdash \Delta}{\Gamma, F \oplus G \vdash \Delta}\ \oplus_L$$

$$\frac{\Gamma, F \vdash \Delta, G}{\Gamma \vdash \Delta, F \multimap G}\ \multimap_R \qquad \frac{\Gamma_1 \vdash \Delta_1, F \quad \Gamma_2, G \vdash \Delta_2}{\Gamma_1, \Gamma_2, F \multimap G \vdash \Delta_1, \Delta_2}\ \multimap_L$$

$$\frac{\Gamma \vdash \Delta, ?F, ?F}{\Gamma \vdash \Delta, ?F}\ \mathsf{cont}_R \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, ?F}\ \mathsf{weak}_R \qquad \frac{\Gamma \vdash \Delta, F}{\Gamma \vdash \Delta, ?F}\ \mathsf{der}_R \qquad \frac{!\Gamma \vdash F, ?\Delta}{!\Gamma \vdash !F, ?\Delta}\ !_R$$

$$\frac{\Gamma, !F, !F \vdash \Delta}{\Gamma, !F \vdash \Delta}\ \mathsf{cont}_L \qquad \frac{\Gamma \vdash \Delta}{\Gamma, !F \vdash \Delta}\ \mathsf{weak}_L \qquad \frac{\Gamma, F \vdash \Delta}{\Gamma, !F \vdash \Delta}\ \mathsf{der}_L \qquad \frac{!\Gamma, F \vdash ?\Delta}{!\Gamma, ?F \vdash ?\Delta}\ ?_L$$

Figure 2: Sequent system CLL.

$$\frac{}{p \vdash p}\ \mathsf{init} \qquad \frac{}{\vdash 1}\ 1_R \qquad \frac{}{\Gamma \vdash \top}\ \top_R \qquad \frac{\Gamma \vdash}{\Gamma \vdash \bot}\ \bot_R \qquad \frac{\Gamma \vdash C}{\Gamma, 1 \vdash C}\ 1_L \qquad \frac{}{\bot \vdash}\ \bot_L \qquad \frac{}{\Gamma, 0 \vdash C}\ 0_L$$

$$\frac{\Gamma \vdash F \quad \Gamma \vdash G}{\Gamma \vdash F \,\&\, G}\ \&_R \qquad \frac{\Gamma_1 \vdash F \quad \Gamma_2 \vdash G}{\Gamma_1, \Gamma_2 \vdash F \otimes G}\ \otimes_R \qquad \frac{\Gamma \vdash F_i}{\Gamma \vdash F_1 \oplus F_2}\ \oplus_{R_i} \qquad \frac{\Gamma, F_i \vdash C}{\Gamma, F_1 \,\&\, F_2 \vdash C}\ \&_{L_i}$$

$$\frac{\Gamma, F, G \vdash C}{\Gamma, F \otimes G \vdash C}\ \otimes_L \qquad \frac{\Gamma, F \vdash C \quad \Gamma, G \vdash C}{\Gamma, F \oplus G \vdash C}\ \oplus_L \qquad \frac{\Gamma, F \vdash G}{\Gamma \vdash F \multimap G}\ \multimap_R \qquad \frac{\Gamma_1 \vdash F \quad \Gamma_2, G \vdash C}{\Gamma_1, \Gamma_2, F \multimap G \vdash C}\ \multimap_L$$

$$\frac{!\Gamma \vdash F}{!\Gamma \vdash !F}\ !_R \qquad \frac{\Gamma, !F, !F \vdash C}{\Gamma, !F \vdash C}\ \mathsf{cont}_L \qquad \frac{\Gamma \vdash C}{\Gamma, !F \vdash C}\ \mathsf{weak}_L \qquad \frac{\Gamma, F \vdash C}{\Gamma, !F \vdash C}\ \mathsf{der}_L$$

Figure 3: System ILL.

The (one-sided) focused system LLF for classical linear logic is presented in Figure. 4. There are two kinds of arrows in this proof system and a pair of contexts to the left of the arrows: Θ is a set of formulas whose main connective is a question-mark, being hence the bounded context, while Γ is a multi-set of linear formulas, behaving as the bounded context. Sequents with the $\Downarrow$ belong to the positive phase and introduce the logical connective of the "focused" formula (the one to the right of the arrow): building proofs of such sequents may require non-invertible proof steps to be taken. Sequents with the $\Uparrow$ belong to the negative phase and decompose the multiset of formulas on the right in such a way that only inference rules over negative formulas are applied: the others are "stored" in the linear context using $R\Uparrow$. The structural rules D_1, D_2 and $R\Downarrow$ make the transition between negative and positive phases. The *positive* phase begins by choosing a positive formula F on which to focus (using D_1, D_2). Positive rules are applied to F until either 1 or a negated atom is encountered (and the proof must end by applying the

initial rules) or the promotion rule (!) is applied or a negative subformula is encountered ($R\Downarrow$) when the proof switches to the negative phase.

Formulas in LLF are taken to be in *negation normal form* using the standard classical linear logic dualities, e.g., $(F \otimes G)^{\perp} \equiv F^{\perp} \parr G^{\perp}$. Hence negation has only atomic scope. As usual, we represent $A \multimap B$ as $A^{\perp} \parr B$.

Introduction Rules

$$\frac{\vdash \Theta : \Gamma \Uparrow L}{\vdash \Theta : \Gamma \Uparrow L, \bot}\ \bot \qquad \frac{}{\vdash \Theta : \Gamma \Uparrow L, \top}\ \top \qquad \frac{}{\vdash \Theta :\Downarrow 1}\ 1 \qquad \frac{\vdash \Theta : \Gamma \Uparrow L, F, G}{\vdash \Theta : \Gamma \Uparrow L, F \parr G}\ \parr$$

$$\frac{\vdash \Theta : \Gamma \Uparrow L, F \quad \vdash \Theta : \Gamma \Uparrow L, G}{\vdash \Theta : \Gamma \Uparrow L, F \,\&\, G}\ \& \qquad \frac{\vdash \Theta : \Gamma \Downarrow F \quad \vdash \Theta : \Gamma' \Downarrow G}{\vdash \Theta : \Gamma, \Gamma' \Downarrow F \otimes G}\ \otimes \qquad \frac{\vdash \Theta : \Gamma \Downarrow F_i}{\vdash \Theta : \Gamma \Downarrow F_1 \oplus F_2}\ \oplus_i$$

$$\frac{\vdash \Theta, F : \Gamma \Uparrow L}{\vdash \Theta : \Gamma \Uparrow L, ?F}\ ? \qquad \frac{\vdash \Theta :\Uparrow F}{\vdash \Theta :\Downarrow\, !F}\ !$$

Identity, Reaction, and Decide rules

$$\frac{}{\vdash \Theta : A_p^{\perp} \Downarrow A_p}\ I_1 \qquad \frac{}{\vdash \Theta, A_p^{\perp} :\Downarrow A_p}\ I_2 \qquad \frac{\vdash \Theta : \Gamma, S \Uparrow L}{\vdash \Theta : \Gamma \Uparrow L, S}\ R\Uparrow$$

$$\frac{\vdash \Theta : \Gamma \Downarrow P}{\vdash \Theta : \Gamma, P \Uparrow}\ D_1 \qquad \frac{\vdash \Theta, P : \Gamma \Downarrow P}{\vdash \Theta, P : \Gamma \Uparrow}\ D_2 \qquad \frac{\vdash \Theta : \Gamma \Uparrow N}{\vdash \Theta : \Gamma \Downarrow N}\ R\Downarrow$$

Figure 4: The focused proof system LLF for classical linear logic. Here, L is a list of formulas, A_p is a positive literal, S is positive or a literal, P is positive and N is negative.

In this work, we will use both: LLF for classical proofs and its intuitionistic version ILLF, presented in Figure 5.

Negative Phase

$$\frac{\Theta : \Gamma, F, G \longrightarrow C}{\Theta : \Gamma, F \otimes G \longrightarrow C}\ \otimes_L \qquad \frac{\Theta : \Gamma, F \longrightarrow G}{\Theta : \Gamma \longrightarrow F \multimap G}\ \multimap_R \qquad \frac{\Theta : \Gamma \longrightarrow C}{\Theta : \Gamma, 1 \longrightarrow C}\ 1_L \qquad \frac{\Theta : \Gamma \longrightarrow}{\Theta : \Gamma \longrightarrow \bot}\ \bot_R$$

$$\frac{}{\Theta : \Gamma \longrightarrow \top}\ \top_R \qquad \frac{}{\Theta : \Gamma, 0 \longrightarrow C}\ 0_L \qquad \frac{\Theta, F : \Gamma \longrightarrow C}{\Theta : \Gamma, !F \longrightarrow C}\ !_L$$

$$\frac{\Theta : \Gamma \longrightarrow F \quad \Theta : \Gamma \longrightarrow G}{\Theta : \Gamma \longrightarrow F \,\&\, G}\ \&_R \qquad \frac{\Theta : \Gamma, F \longrightarrow C \quad \Theta : \Gamma, H \longrightarrow C}{\Theta : \Gamma, F \oplus H \longrightarrow C}\ \oplus_L$$

Positive Phase

$$\frac{\Theta : \Gamma_1 - F \rightarrow \quad \Theta : \Gamma_2 - G \rightarrow}{\Theta : \Gamma_1, \Gamma_2 - F \otimes G \rightarrow}\ \otimes_R \qquad \frac{\Theta : \Gamma_1 - F \rightarrow \quad \Theta : \Gamma_2 \xrightarrow{G} C}{\Theta : \Gamma_1, \Gamma_2 \xrightarrow{F \multimap G} C}\ \multimap_L \qquad \frac{\Theta : \Gamma - F_i \rightarrow}{\Theta : \Gamma - F_1 \oplus F_2 \rightarrow}\ \oplus_{R_i}$$

$$\frac{\Theta : \Gamma \xrightarrow{F_i} C}{\Theta : \Gamma \xrightarrow{F_1 \& F_2} C}\ \&_{L_i} \qquad \frac{}{\Theta : -_1 \rightarrow}\ 1_R \qquad \frac{}{\Theta : \xrightarrow{\bot}}\ \bot_L \qquad \frac{\Theta : \longrightarrow F}{\Theta : -_{!F} \rightarrow}\ !_R$$

$$\frac{}{\Theta : \Gamma - A \rightarrow}\ I_R \text{ given } A \in (\Theta \cup \Gamma) \text{ and } \Gamma \subseteq \{A\}$$

Structural Rules

$$\frac{\Theta, N_a : \Gamma \xrightarrow{N_a} C}{\Theta, N_a : \Gamma \longrightarrow C}\ D_{L1} \qquad \frac{\Theta : \Gamma \xrightarrow{N_a} C}{\Theta : \Gamma, N_a \longrightarrow C}\ D_{L2} \qquad \frac{\Theta : \Gamma - P_a \rightarrow}{\Theta : \Gamma \longrightarrow P_a}\ D_R$$

$$\frac{\Theta : \Gamma, P_a \longrightarrow C}{\Theta : \Gamma \xrightarrow{P_a} C}\ R_L \qquad \frac{\Theta : \Gamma \longrightarrow N}{\Theta : \Gamma - N \rightarrow}\ R_R$$

Figure 5: System ILLF: a focused proof system for ILL. Here, A is an atomic formula; N_a is a negative non atomic formula; P_a is a positive or atomic formula; N is a negative formula.

B Multiplicative translation of Kleene's list

1. $\vdash A \multimap A$ (identity)
2. $A \multimap B, B \multimap C \vdash A \multimap C$ (transitivity of implication)
3. $A \multimap (B \multimap C) \vdash B \multimap (A \multimap C)$ (exchange of premises)
4. $A \multimap (B \multimap C) \vdash A \otimes B \multimap C$ (uncurrying)
5. $A \otimes B \multimap C \vdash A \multimap (B \multimap C)$ (currying)
6. $A \multimap B \vdash (B \multimap C) \multimap (A \multimap C)$ (precomposing maps)
7. $A \multimap B \vdash (C \multimap A) \multimap (C \multimap B)$ (post-composing maps)
8. $A \multimap B \vdash A \otimes C \multimap B \otimes C$ (tensor is a bifunctor)
9. $A \multimap B \vdash C \otimes A \multimap C \otimes B$ (tensor is a bifunctor)
10. $A^\perp \nvdash A \multimap B$
11. $A \nvdash A^\perp \multimap B$
12. $B \nvdash A \multimap B$ but (projection is non-linear)
13. $A \multimap B \vdash B^\perp \multimap A^\perp$ (linear negation is contravariant)
14. $A \multimap B^\perp \vdash B^\perp \multimap A^\perp$
15. $A \multimap B, B \multimap A \vdash A \multimap\!\!\circ\, B$
16. $A \multimap\!\!\circ\, B \nvdash A \multimap B$ (cannot throw away $B \multimap A$)
17. $A \multimap\!\!\circ\, B \nvdash B \multimap A$ (cannot throw away $A \multimap B$)
18. $A \multimap\!\!\circ\, B, A \nvdash B$
19. $A \multimap\!\!\circ\, B, B \nvdash A$
20. $\vdash A \multimap\!\!\circ\, A$
21. $A \multimap\!\!\circ\, B \vdash B \multimap\!\!\circ\, A$
22. $A \multimap\!\!\circ\, B, B \multimap\!\!\circ\, C \vdash A \multimap\!\!\circ\, C$
23. $A \multimap (B \multimap C), A^{\perp\perp}, B^{\perp\perp} \vdash C^{\perp\perp}$
24. $(A \multimap B)^{\perp\perp} \vdash A^{\perp\perp} \multimap B^{\perp\perp}$ (double negation is a functor)
25. $(A \multimap B)^{\perp\perp}, (B \multimap C)^{\perp\perp} \vdash (A \multimap C)^{\perp\perp}$
26. $\nvdash (A \otimes B)^{\perp\perp} \multimap\!\!\circ\, A^{\perp\perp} \otimes B^{\perp\perp}$
27. $\nvdash (A \multimap\!\!\circ\, B)^{\perp\perp} \multimap\!\!\circ\, (A \multimap B)^{\perp\perp} \otimes (B \multimap A)^{\perp\perp}$
28. $A \multimap\!\!\circ\, B \vdash (A \multimap C) \multimap\!\!\circ\, (B \multimap C)$
29. $A \multimap\!\!\circ\, B \vdash (C \multimap A) \multimap\!\!\circ\, (C \multimap B)$
30. $A \multimap\!\!\circ\, B \vdash (A \otimes C) \multimap\!\!\circ\, (B \otimes C)$
31. $A \multimap\!\!\circ\, B \vdash (C \otimes A) \multimap\!\!\circ\, (C \otimes B)$
32. $A \multimap\!\!\circ\, B \vdash A^\perp \multimap\!\!\circ\, B^\perp$
33. $\vdash ((A \otimes B) \otimes C) \multimap\!\!\circ\, (A \otimes (B \otimes C))$.
34. $\vdash A \otimes B \multimap\!\!\circ\, B \otimes A$
35. $\nvdash A \otimes A \multimap\!\!\circ\, A$ ($\otimes$ is not idempotent)
36. $A \nvdash (A \multimap B) \multimap\!\!\circ\, B$
37. $B \nvdash (A \multimap B) \multimap\!\!\circ\, B$
38. $A^\perp \nvdash (A \multimap B) \multimap\!\!\circ\, A^\perp$
39. $B^\perp \nvdash (A \multimap B) \multimap\!\!\circ\, A^\perp$
40. $B \nvdash (A \otimes B) \multimap\!\!\circ\, A$
41. $B^\perp \nvdash ((!A \otimes B) \multimap B) \otimes (B \multimap (A \otimes B))$
42. $\vdash A \multimap A^{\perp\perp}$
43. $\vdash A^{\perp\perp\perp} \multimap\!\!\circ\, A^\perp$
44. $\vdash (A \otimes A^\perp)^\perp$
45. $\nvdash (A \multimap\!\!\circ\, A^\perp)^\perp$
46. $\nvdash (((A \multimap 0) \multimap 0) \multimap A)^{\perp\perp}$
47. $\nvdash A \otimes (B \otimes B^\perp) \multimap\!\!\circ\, (B \otimes B^\perp)$
48. $\vdash (A \multimap B) \multimap (A \otimes B^\perp)^\perp$
49. $\vdash (A \multimap B^\perp) \multimap\!\!\circ\, (A \otimes B)^\perp$
50. $\vdash (A \otimes B)^\perp \multimap\!\!\circ\, (A^{\perp\perp} \multimap B^\perp)$
51. $B^{\perp\perp} \multimap B \vdash (A^{\perp\perp} \multimap B) \multimap\!\!\circ\, (A \multimap B)$
52. $B^{\perp\perp} \multimap B \vdash (A \multimap B) \multimap\!\!\circ\, (A \otimes B^\perp)^\perp$
53. $\vdash (A^{\perp\perp} \multimap B) \multimap (A \otimes B^\perp)^\perp$
54. $\vdash A \otimes B \multimap (A \multimap B^\perp)^\perp$
55. $\vdash A \otimes B^\perp \multimap (A \multimap B)^\perp$
56. $\vdash (A^{\perp\perp} \otimes B) \multimap (A \multimap B^\perp)^\perp$
57. $\nvdash (A^{\perp\perp} \otimes B^\perp) \multimap\!\!\circ\, (A \multimap B)^\perp$
58. $\nvdash (A \multimap B)^\perp \multimap\!\!\circ\, (A \otimes B^\perp)^{\perp\perp}$
59. $\nvdash (A \multimap B)^{\perp\perp} \multimap\!\!\circ\, (A \otimes B^\perp)^\perp$
60. $\vdash (A \otimes B^\perp)^\perp \multimap\!\!\circ\, (A \multimap B^{\perp\perp})$
61. $\vdash (A \multimap B^{\perp\perp}) \multimap\!\!\circ\, (A^{\perp\perp} \multimap B^{\perp\perp})$