Craig-Lyndon Interpolation for the Logic of Here and There with a Variation of Mints' Sequent System (Extended Abstract)

Christoph Wernhard

University of Potsdam, Germany

Abstract

We present a variation of Maehara's method to construct Craig-Lyndon interpolants for the three-valued propositional logic of here and there (HT), also known as Gödel's G_3, a superintuitionistic logic of importance in logic programming. Our method adapts a recent interpolation technique that operates on classically encoded logic programs to a variation of Mints' sequent system for HT. The approach is characterized by two stages: First, a preliminary interpolant is constructed, a formula that is an interpolant in some sense but not yet the desired HT formula. In the second stage, an actual HT interpolant is obtained from this preliminary interpolant. With the classical encoding, the preliminary interpolant is a classical Craig-Lyndon interpolant for classical encodings of the two input HT formulas. In the presented adaptation, the sequent system operates directly on HT formulas, and the preliminary interpolant is in a nonclassical logic that generalizes HT by an additional logic operator.

Keywords

Craig interpolation, Sequent systems, Maehara's method, Logic of here and there (HT), Gödel's logic G_3

1. Introduction

The propositional *logic of here and there (HT)*, also known as Gödel's G_3, is a superintuitionistic, or intermediate, logic, which plays an important role in logic programming. Central to this role is the observation, due to Lifschitz, Pearce and Valverde [1], that for answer set programs under the stable model semantics a particularly useful notion of program equivalence, *strong equivalence*, can be characterized as equivalence in HT. We consider here *Craig-Lyndon interpolation* for HT, aiming at applications in knowledge representation and program synthesis such as described, e.g., in [2, 3, 4]. As noted in [2], results by Maksimova [5, 6, 7] prove that HT has the Craig interpolation property, i.e., for two HT formulas such that one can be inferred from the other, a Craig interpolant exists. Baaz and Veith [8, Section 4.1.1] have shown that uniform interpolants for HT can be constructed as the disjunction over the formulas obtained from the given formula by substituting the atoms to eliminate in all possible ways with atoms to be retained and truth value constants for true and false.

Recently [4], Craig-Lyndon interpolation for HT was proven with a practically applicable construction of HT interpolants from clausal tableau proofs, which can be obtained from theorem provers for classical first-order logic. The underlying proof system, clausal tableaux, is for classical first-order logic and it is applied to HT formulas under a specific encoding into classical logic. This encoding is well established as a technique for mechanically proving strong equivalence of two given logic programs with theorem provers for classical logic. The core idea of the encoding is to represent an HT predicate by two classical predicates, one for the "here" world and the other for the "there" world.

Since [4] mainly addresses logic programming, HT interpolation is expressed in that paper as "LP-interpolation" on the level of first-order formulas that encode HT formulas [4, Theorem 11]. The transfer to interpolation directly on "logic programs", i.e., HT formulas in a certain normal form, is straightforward and made explicit in the Prolog implementation that accompanies the paper (predicate `p_interpolant/4`). Since any HT formula can be normalized into the special case of a "logic program", as outlined by Mints [9], the method of [4] is applicable w.l.o.g. to HT in full. Moreover, the method

CI-BD-SOQE 2026: Workshop on Craig Interpolation, Beth Definability, and Second-Order Quantifier Elimination, at FLoC 2026: The 9th Federated Logic Conference, July 24–25, 2026, Lisbon, Portugal

from [4] actually applies not just to propositional HT but also to a first-order generalization of HT. As outlined in [4] and also implemented, it can operate either directly on clausal first-order tableaux or on resolution proofs, which are converted to clausal tableaux.

With respect to practical application, the method of [4] provides exactly what is needed: powerful and highly optimized stock first-order provers can be utilized to compute HT interpolants. However, the method does not fit immediately into research threads that are centered around *sequent systems*. Hence, our objective here is the transfer of the method to the sequent system perspective, without involvement of a classical encoding.

With the method, an HT interpolant is constructed in two stages: first, a preliminary interpolant is constructed, a formula that is an interpolant in some sense but not yet the desired HT formula. In the second stage, an actual HT interpolant is obtained from this preliminary interpolant. In the "classical encoding variation" of the method, as described in [4], the preliminary interpolant is a classical Craig-Lyndon interpolant for classical encodings of the two input HT formulas. A postprocessing operation then yields the classical encoding of an HT formula that is an interpolant of the original HT inputs. Decoding the latter formula into an actual HT formula is then a trivial matter. In the "sequent system variation", described here, the input HT formulas are directly processed by a sequent system for HT, a modification of the propositional fragment of Mints' system for HT [9]. The sequent proof allows calculation of a preliminary interpolant in a logic that generalizes HT. The postprocessing operation then converts this to a proper HT interpolant.

The rest of this short paper is structured as follows: We specify the considered logics (Sect. 2), and present axioms and rules that extend Mints' system (Sect. 3). After describing Maehara's interpolation method (Sect. 4), we use it to define an interpolating sequent system that constructs for given HT formulas an interpolant in a logic that extends HT (Sect. 5). We then show, how this preliminary interpolant can be converted to the desired HT interpolant (Sect. 6). We conclude by noting issues for further investigation (Sect. 7). Examples of HT interpolation applied to synthesize strongly equivalent logic programs are provided in [4, Examples 13 and 16].

2. Logics

We consider the three-valued propositional logic of here and there (HT) and variations of this logic. Truth values are F (*false*), T (*true*), and NF (*not false*). Logic operators are the well-known $\bot, \top, \neg, \lor, \land, \rightarrow$ along with the unary operator nh („*not here*"). Their semantics is specified with the truth tables in Table 1. A *literal* is either an atom or the negation $\neg p$ of an atom p. The *polarity* of a subformula occurrence is *positive* (*negative*) if it is in the scope of an even (odd) number of occurrences of $\neg$, nh, and, with respect to its first argument, $\rightarrow$. The *polarity-aware vocabulary* $\mathsf{voc}^{\pm}(A)$ of a formula A is defined as the set of all pairs $\langle +, p\rangle$ where p is an atom with a positive occurrence in A and all pairs $\langle -, p\rangle$ where p is an atom with a negative occurrence in A. For example, $\mathsf{voc}^{\pm}((\mathsf{p} \land \neg \mathsf{q}) \rightarrow (\mathsf{nh}(\mathsf{r}) \lor \mathsf{s} \lor \neg s)) = \{\langle -, p\rangle, \langle +, q\rangle, \langle -, r\rangle, \langle +, s\rangle, \langle -, s\rangle\}$.

We quietly assume that semantic assertions with free formula parameters hold for "all" formulas, which includes any formulas that could be built with an extended truth-functionally complete set of operators. Our semantics allows the synonymous use of: (a) A entails B; (b) $A \models B$; (c) $A \rightarrow B$ is valid; (d) Under all assignments of the atoms in A or B to truth values from $\{\mathsf{F}, \mathsf{NF}, \mathsf{T}\}$ the truth value of $A \rightarrow B$ is T.[1] We use $A \leftrightarrow B$ to abbreviate $(A \rightarrow B) \land (B \rightarrow A)$ and synonymously use: (a) A and B are equivalent; (b) $A \equiv B$; (c) $A \leftrightarrow B$ is valid; (d) Under all assignments of the atoms in A or B to truth values from $\{\mathsf{F}, \mathsf{NF}, \mathsf{T}\}$ the truth value of $A \leftrightarrow B$ is T; (e) Under all assignments of the atoms in A or B to truth values from $\{\mathsf{F}, \mathsf{NF}, \mathsf{T}\}$ the truth value of A is the same as the truth value of B.

[1]Entailment is understood *comparative* [10]: $A \models B$ iff under all truth value assignments the value of A is less than or equal to that of B, where $\mathsf{F} \leq \mathsf{NF} \leq \mathsf{T}$. For HT, which has a classical deduction theorem, this is equivalent to *preservation of the designated value*: under all truth value assignments, if the value of A is T, then that of B is T.

Table 1
Truth tables for the considered logic operators. They extend the valuations for HT by the specification of $\mathsf{nh}(A)$.

$\bot$	$\top$
F	T

A	$\neg A$	$\mathsf{nh}(A)$
F	T	T
NF	F	T
T	F	F

A	B	$A \vee B$	$A \wedge B$	$A \to B$
F	F	F	F	T
F	NF	NF	F	T
F	T	T	F	T
NF	F	NF	F	F
NF	NF	NF	NF	T
NF	T	T	NF	T
T	F	T	F	F
T	NF	T	NF	NF
T	T	T	T	T

The formulas in which we are ultimately interested and for which we will develop our main result on Craig-Lyndon interpolation are those of HT, defined with the following grammar.

HT-formula: $\quad A, B := \mathit{Atom} \mid \bot \mid \top \mid \neg A \mid (A \vee B) \mid (A \wedge B) \mid (A \to B).$

We include the truth value constants $\top, \bot$, which does not extend expressiveness, as $\bot \equiv (A \wedge \neg A) \equiv \neg\top$ and $\top \equiv \neg\bot$. At least one of $\top$ or $\bot$ is required to take account of interpolation for formulas with disjoint vocabularies. Our construction of preliminary interpolants will involve a further formula class, without $\to$ but with nh and in a negation normal form:

NHNNF-formula: $\quad A, B := \mathit{Atom} \mid \neg \mathit{Atom} \mid \neg\neg \mathit{Atom} \mid \mathsf{nh}(\mathit{Atom}) \mid \bot \mid \top \mid (A \vee B) \mid (A \wedge B).$

Intuitively, $\mathsf{nh}(A)$ is true (i.e., T) if and only if A is false in the *here* world, independently of its value in the *there* world, and false (i.e., F) otherwise. We consider the nh operator because a variation of Maehara's method allows from proofs for HT-formulas the extraction of Craig-Lyndon interpolants that are NHNNF-formulas. In a postprocessing step these can be converted to HT-formulas. The nh operator has the following properties.

$$\neg A \to \mathsf{nh}(A) \equiv \top. \tag{1}$$
$$\mathsf{nh}(A) \vee A \equiv \top. \tag{2}$$
$$\mathsf{nh}(\neg A) \equiv \neg\neg A. \tag{3}$$
$$\mathsf{nh}(A \wedge B) \equiv \mathsf{nh}(A) \vee \mathsf{nh}(B). \tag{4}$$
$$\mathsf{nh}(A \vee B) \equiv \mathsf{nh}(A) \wedge \mathsf{nh}(B). \tag{5}$$
$$A \to B \equiv (\mathsf{nh}(A) \vee B) \wedge (\neg A \vee \neg\neg B). \tag{6}$$

3. Axioms and Rules

Our starting point is the propositional fragment of Mints' sequent system for HT [9]. We assume straightforward restrictions of its axioms and also consider some additional axioms and rules.

Restrictions of Mints' Axioms. We assume the two axioms of Mints' system for HT with the annotated restrictions.

Ax-1	$A, \Gamma \Rightarrow \Delta, A$	A a literal
Ax-2	$A, \neg A, \Gamma \Rightarrow \Delta$	A an atom

Axioms and Rules for Truth Value Constants. Axioms and rules for handling $\top$ and $\bot$ are straightforward and omitted in the presentation.

Axioms and Rules for nh. We have the following axioms and rule.

$$\begin{array}{lll} \text{Ax-nh-1} & \Gamma \Rightarrow \Delta, A, \mathsf{nh}(A) & A \text{ an atom} \\ \text{Ax-nh-2} & \neg A, \Gamma \Rightarrow \Delta, \mathsf{nh}(A) & A \text{ an atom} \end{array}$$

$$\neg\mathsf{nh}\Rightarrow \quad \frac{\Gamma \Rightarrow \mathsf{nh}(A), \Delta}{\neg\mathsf{nh}(A), \Gamma \Rightarrow \Delta}$$

In addition, we have rules for eliminating nh over negation, based on equivalence (3), and for moving nh inward, based on equivalences (4) and (5). The respective rules are straightforward and omitted from the presentation.

Soundness of Ax-nh-1 and Ax-nh-2 follows since the corresponding implications $B \rightarrow (C \lor A \lor \mathsf{nh}(A))$ and $(\neg A \land B) \rightarrow (C \lor \mathsf{nh}(A))$ are both valid. Soundness of $\neg\mathsf{nh}\Rightarrow$ follows from the equivalence $(\neg\mathsf{nh}(A) \land B) \rightarrow C \;\equiv\; B \rightarrow (\mathsf{nh}(A) \lor C)$.

A Third New Rule for Implication. Mints' system handles implication with two rules newly introduced in [9]. We introduce a third new rule, which is, like Mints' $\Rightarrow\rightarrow$, for implication in the succedent. We will use it in the interpolating variation of our sequent system in specific cases, while in other cases we will stay with Mints' version.

$$\Rightarrow\rightarrow* \quad \frac{\Gamma \Rightarrow \Delta, \mathsf{nh}(A), B \qquad \neg B, \Gamma \Rightarrow \Delta, \neg A}{\Gamma \Rightarrow \Delta, (A \rightarrow B)}$$

Soundness of the rule $\Rightarrow\rightarrow*$ follows from the equivalence

$$C \rightarrow (D \lor (A \rightarrow B)) \;\equiv\; (C \rightarrow (D \lor \mathsf{nh}(A) \lor B)) \land ((\neg B \land C) \rightarrow (D \lor \neg A)). \tag{7}$$

4. Interpolating Sequent Systems – Maehara's Method

Our method for Craig-Lyndon interpolation for HT operates by first computing for two given HT-formulas A, B a Craig-Lyndon interpolant C' that is a NHNNF-formula. This is done with *Maehara's method* [11, 12, 13, 14], a common technique for extracting an interpolant from a sequent proof. We briefly outline this technique in the context of classical propositional logic. As basis we take a sequent system of the G3 family [14, Sect. 3.5], where structural rules are absorbed. Interpolant construction is then specified through axioms and rules for a generalization of sequents, so-called *split-sequents*, which have the form

$$\Gamma^{\mathsf{L}}, \Gamma^{\mathsf{R}} \overset{H}{\Longrightarrow} \Delta^{\mathsf{L}}, \Delta^{\mathsf{R}}. \tag{8}$$

In a split-sequent each formula occurrence that is a member of the antecedent or of the succedent (antecedent and succedent are multisets) is labeled with a *provenance* $\in \{\mathsf{L}, \mathsf{R}\}$. We write the multiset of the members of the antecedent with provenance L (R) as Γ^{L} (Γ^{R}), and, analogously, the multiset of the members of the succedent with provenance L (R) as Δ^{L} (Δ^{R}). We write Γ for $\Gamma^{\mathsf{L}}, \Gamma^{\mathsf{R}}$ and Δ for $\Delta^{\mathsf{L}}, \Delta^{\mathsf{R}}$.

As usual for G3 systems, axioms and rules are suitable for bottom-up proof construction that starts with the given goal sequent as root and ends with axiom instances as leaves. The rules with split-sequents now specify, in addition, how the provenance labeling of formula occurrences is propagated upwards from the principal formula in the rule conclusion to the corresponding active formulas in the premises.

A split-sequent is decorated with a formula H, the *relative interpolant* of the sequent, such that any instance in a proof satisfies the following conditions.

$$\textstyle\bigwedge \Gamma^{\mathsf{L}} \models H \lor \bigvee \Delta^{\mathsf{L}}. \tag{I1}$$

$$\textstyle\bigwedge \Gamma^{\mathsf{R}} \land H \models \bigvee \Delta^{\mathsf{R}}. \tag{I2}$$

$$\textstyle\mathsf{voc}^{\pm}(H) \subseteq \mathsf{voc}^{\pm}(\bigwedge \Gamma^{\mathsf{L}} \land \neg \bigvee \Delta^{\mathsf{L}}) \cap \mathsf{voc}^{\pm}(\neg \bigwedge \Gamma^{\mathsf{R}} \lor \bigvee \Delta^{\mathsf{R}}) \tag{I3}$$

Observe that if $\Gamma = A^{\mathsf{L}}$ and $\Delta = B^{\mathsf{R}}$, then conditions (I1–I3) characterize H as a Craig-Lyndon interpolant of A and B. Thus, a Craig-Lyndon interpolant of given formulas A and B is obtained as formula H from a proof of the root sequent

$$A^{\mathsf{L}} \overset{H}{\Longrightarrow} B^{\mathsf{R}}. \tag{9}$$

Through the decoration with the relative interpolant H, the axioms and rules of the sequent system specify an inductive construction of the Craig-Lyndon interpolant, which proceeds downwards from leaves, axiom instances, to the root $A^{\mathsf{L}} \overset{H}{\Longrightarrow} B^{\mathsf{R}}$. Correctness of the interpolation construction then means to show that conditions (I1–I3) apply to the axioms and, for a rule, if they are assumed for the premises, then they also hold for the conclusion.

As an example rule consider the interpolating version of $\Rightarrow\wedge$ for classical logic (and also HT), where we have two cases, one for each possible provenance labeling of the principal formula.

$$\Rightarrow\wedge\mathsf{L} \qquad \frac{\Gamma \overset{H_1}{\Longrightarrow} \Delta, A^{\mathsf{L}} \qquad \Gamma \overset{H_2}{\Longrightarrow} \Delta, B^{\mathsf{L}}}{\Gamma \overset{H_1 \vee H_2}{\Longrightarrow} \Delta, (A \wedge B)^{\mathsf{L}}}$$

$$\Rightarrow\wedge\mathsf{R} \qquad \frac{\Gamma \overset{H_1}{\Longrightarrow} \Delta, A^{\mathsf{R}} \qquad \Gamma \overset{H_2}{\Longrightarrow} \Delta, B^{\mathsf{R}}}{\Gamma \overset{H_1 \wedge H_2}{\Longrightarrow} \Delta, (A \wedge B)^{\mathsf{R}}}$$

For both rules it is easy to see that syntactic condition (I3) transfers from the premises to the conclusion. To show for $\Rightarrow\wedge\mathsf{L}$ the transfer of the semantic condition (I1) we assume that it holds for the premises, i.e., $\bigwedge \Gamma^{\mathsf{L}} \models H_1 \vee \bigvee \Delta^{\mathsf{L}} \vee A$ and $\bigwedge \Gamma^{\mathsf{L}} \models H_2 \vee \bigvee \Delta^{\mathsf{L}} \vee B$. We can then conclude that it holds for the conclusion: $\bigwedge \Gamma^{\mathsf{L}} \models (H_1 \vee H_2) \vee \bigvee \Delta^{\mathsf{L}} \vee (A \wedge B)$. To show the transfer of the semantic condition (I2) we assume $\bigwedge \Gamma^{\mathsf{R}} \wedge H_1 \models \bigvee \Delta^{\mathsf{R}}$ and $\bigwedge \Gamma^{\mathsf{R}} \wedge H_2 \models \bigvee \Delta^{\mathsf{R}}$, and conclude $\bigwedge \Gamma^{\mathsf{R}} \wedge (H_1 \vee H_2) \models \bigvee \Delta^{\mathsf{R}}$.

5. An Interpolating Sequent System for HT with NHNNF Interpolants

We now go into detail of the first stage of our Craig-Lyndon interpolation method for HT, computing for two given HT-formulas A, B a Craig-Lyndon interpolant C' that is a NHNNF-formula. This is done with an interpolating sequent system based on a variation of the propositional fragment of Mints' system for HT from [9]. In a second stage, from the NHNNF-formula C' a HT-formula C is constructed that is a Craig-Lyndon interpolant of A, B. This two-stage proceeding adapts a technique from [4], where a Craig-Lyndon interpolant of two classical formulas that encode HT-formulas is postprocessed to a Craig-Lyndon interpolant that again encodes a HT-formula. The second stage will be presented in Sect. 6. The result of the first stage is expressed with the following theorem.

Theorem 1. *Let A, B be HT-formulas such that $A \models B$. Then there exists a NHNNF-formula C' such that*

1. *$A \models C'$ and $C' \models B$, and*
2. $\mathsf{voc}^{\pm}(C') \subseteq \mathsf{voc}^{\pm}(A) \cap \mathsf{voc}^{\pm}(B)$.

Moreover, such a NHNNF-formula C' can be effectively constructed from a proof of $A \models B$ in a variation of Mints' sequent system for HT.

We prove this theorem over the rest of this section by specifying an interpolating sequent system that constructs a suitable NHNNF-formula C' from given HT-formulas A and B.

5.1. An Assumption on the Form of the Second Argument

Definition 2. *We call a HT-formula* body-normalized *if no implication occurs in the antecedent of another implication.*

We assume that the second argument formula to Craig-Lyndon interpolation, is *body-normalized.* This assumption is w.l.o.g. as any HT-formula can be converted to an equivalent one that is body-normalized by rewriting with the following equivalences.

$$(A \to B) \to C \;\equiv\; (\neg A \to C) \land (A \lor \neg B \lor C) \land (B \to C). \tag{10}$$

$$(A \lor B) \to C \;\equiv\; (A \to C) \land (B \to C). \tag{11}$$

$$(A \land B) \to C \;\equiv\; (A \to C) \lor (B \to C). \tag{12}$$

$$\neg A \to B \;\equiv\; \neg\neg A \lor B. \tag{13}$$

5.2. Important Properties of Proofs in the System

Before presenting the interpolating system, we state properties of all split-sequents in proofs of a root sequent $A^{\mathsf{L}} \stackrel{C'}{\Longrightarrow} B^{\mathsf{R}}$, which underlie interpolation. This helps in the later presentation of the system details as it brings their crucial features to attention, although, of course, technically the stated properties are consequences of these details. Verifying these properties is not hard but somewhat tedious, typically by inductively proceeding downwards from leaves to the root, or upwards, from the root to the leaves. We sketch some of the proofs when presenting axioms and rules in the context of discussing their crucial features.

Lemma 3. *Consider a proof with root sequent $A^{\mathsf{L}} \stackrel{C'}{\Longrightarrow} B^{\mathsf{R}}$ in the interpolating system of Sect. 5.3, where A is a HT-formula and B is a body-normalized HT-formula. The following properties hold for all instances $\Gamma \stackrel{H}{\Longrightarrow} \Delta$ of split-sequents in the proof.*

(i) *H is a NHNNF-formula.*

(ii) *Conditions (I1)–(I3) (Sect. 4) are satisfied.*[2]

(iii) *All occurrences of* nh *have provenance* R.

(iv) *All occurrences of* nh *are as top-level operators of members of* Δ.

(v) *Argument formulas of* nh *have no occurrences of* $\to$ *and no occurrences of* nh.

(vi) *All members of Γ^{R} are negations, i.e., formulas with $\neg$ as top-level operator.*

Theorem 1 then follows immediately from Lemmas 3.i and 3.ii.

5.3. Axioms and Rules of the Interpolating System

Axioms. We need the following split-sequent variations of Ax-1 (Sect. 3).

Ax-1-LL	$A^{\mathsf{L}}, \Gamma \stackrel{\bot}{\Longrightarrow} \Delta, A^{\mathsf{L}}$	A a literal
Ax-1-LR	$A^{\mathsf{L}}, \Gamma \stackrel{A}{\Longrightarrow} \Delta, A^{\mathsf{R}}$	A a literal
Ax-1-RL	$(\neg A)^{\mathsf{R}}, \Gamma \stackrel{\neg\neg A}{\Longrightarrow} \Delta, (\neg A)^{\mathsf{L}}$	A an atom
Ax-1-RR	$(\neg A)^{\mathsf{R}}, \Gamma \stackrel{\top}{\Longrightarrow} \Delta, (\neg A)^{\mathsf{R}}$	A an atom

Axioms Ax-1-RL and Ax-1-RR apply only to instances of axiom Ax-1 where the principal formulas are *negative* literals. By Lemma 3.vi, instances where an atom with provenance R is a member of the antecedent are superfluous. For the provenance labeling of Ax-1-RL this restriction to negative literals is crucial since for a split-sequent $A^{\mathsf{R}}, \Gamma \stackrel{H}{\Longrightarrow} \Delta, A^{\mathsf{L}}$ in the case where A is an atom there actually is no formula H that satisfies the conditions (I1)–(I3), which characterize the relative interpolant. 'No formula' is quite broad here, even beyond the considered formula classes: there is no possible logic operator $op(A)$ that would map atom A depending on its truth value to another truth value such that $op(A)$ provides a relative interpolant H. For, e.g., Ax-1-RL conditions (I1) and (I2) instantiate to $\bigwedge \Gamma^{\mathsf{L}} \models \neg\neg A \lor \bigvee \Delta^{\mathsf{L}} \lor \neg A$, and $\neg A \land \bigwedge \Gamma^{\mathsf{R}} \land \neg\neg A \models \bigvee \Delta^{\mathsf{R}}$.

[2]Although these conditions were stated in Sect. 4 for classical logic, they apply here if logic operators are understood according to Table 1 and entailment is understood comparative (see footnote 1).

Of axiom Ax-2 we need the following split-sequent variations.

$$\begin{array}{lll} \text{Ax-2-LL} & A^{\mathsf{L}}, (\neg A)^{\mathsf{L}}, \Gamma \overset{\bot}{\Longrightarrow} \Delta & A \text{ an atom} \\ \text{Ax-2-LR} & A^{\mathsf{L}}, (\neg A)^{\mathsf{R}}, \Gamma \overset{A}{\Longrightarrow} \Delta & A \text{ an atom} \\ \text{Ax-2-LR}' & A^{\mathsf{L}}, (\neg A)^{\mathsf{R}}, \Gamma \overset{\neg\neg A}{\Longrightarrow} \Delta & A \text{ an atom} \end{array}$$

Instances of axiom Ax-2 with provenance A^{L} and $(\neg A)^{\mathsf{R}}$ have both A and also $\neg\neg A$ as interpolants. Thus either one of axioms Ax-2-LR or Ax-2-LR′ can be chosen alternatively. For, e.g., Ax-2-LR, conditions (I1) instantiate to $A \land \bigwedge \Gamma^{\mathsf{L}} \models A \lor \bigvee \Delta^{\mathsf{L}}$ and $\bigwedge \Gamma^{\mathsf{R}} \land \neg A \land A \models \bigvee \Delta^{\mathsf{R}}$. For Ax-2-LR′ they instantiate to $A \land \bigwedge \Gamma^{\mathsf{L}} \models \neg\neg A \lor \bigvee \Delta^{\mathsf{L}}$ and $\bigwedge \Gamma^{\mathsf{R}} \land \neg A \land \neg\neg A \models \bigvee \Delta^{\mathsf{R}}$. Axioms Ax-nh-1 and Ax-nh-2 (Sect. 3) are needed in the following split-sequent variations.

$$\begin{array}{lll} \text{Ax-nh-1-LR} & \Gamma \overset{\mathsf{nh}(A)}{\Longrightarrow} \Delta, A^{\mathsf{L}}, \mathsf{nh}(A)^{\mathsf{R}} & A \text{ an atom} \\ \text{Ax-nh-1-RR} & \Gamma \overset{\top}{\Longrightarrow} \Delta, A^{\mathsf{R}}, \mathsf{nh}(A)^{\mathsf{R}} & A \text{ an atom} \\ \text{Ax-nh-2-LR} & (\neg A)^{\mathsf{L}}, \Gamma \overset{\neg A}{\Longrightarrow} \Delta, \mathsf{nh}(A)^{\mathsf{R}} & A \text{ an atom} \\ \text{Ax-nh-2-RR} & (\neg A)^{\mathsf{R}}, \Gamma \overset{\top}{\Longrightarrow} \Delta, \mathsf{nh}(A)^{\mathsf{R}} & A \text{ an atom} \end{array}$$

That other potential variations of the axioms for nh are superfluous follows from Lemmas 3.iii and 3.iv. Instances of Ax-nh-1-LR introduce subformulas $\mathsf{nh}(A)$ into the constructed interpolant. Actually this is the only place where nh is introduced into the interpolant. For, e.g., Ax-nh-1-LR conditions (I1) and (I2) instantiate to $\bigwedge \Gamma^{\mathsf{L}} \models \mathsf{nh}(A) \lor \bigvee \Delta^{\mathsf{L}} \lor A$ and $\Gamma^{\mathsf{R}} \land \mathsf{nh}(A) \models \bigvee \Delta^{\mathsf{R}} \lor \mathsf{nh}(A)$.

Implication Rules. For provenance L of the principal formula, we use Mint's rules $\to\Rightarrow$ and $\Rightarrow\to$.

$$\to\Rightarrow\mathrm{L} \quad \frac{\neg A^{\mathsf{L}}, \Gamma \overset{H_1}{\Longrightarrow} \Delta \qquad \Gamma \overset{H_2}{\Longrightarrow} \Delta, A^{\mathsf{L}}, (\neg B)^{\mathsf{L}} \qquad B^{\mathsf{L}}, \Gamma \overset{H_3}{\Longrightarrow} \Delta}{(A \to B)^{\mathsf{L}}, \Gamma \overset{H_1 \lor H_2 \lor H_3}{\Longrightarrow} \Delta}$$

$$\Rightarrow\to\mathrm{L} \quad \frac{A^{\mathsf{L}}, \Gamma \overset{H_1}{\Longrightarrow} \Delta, B^{\mathsf{L}} \qquad (\neg B)^{\mathsf{L}}, \Gamma \overset{H_2}{\Longrightarrow} \Delta, (\neg A)^{\mathsf{L}}}{\Gamma \overset{H_1 \lor H_2}{\Longrightarrow} \Delta, (A \to B)^{\mathsf{L}}}$$

For provenance R of the principal formula, we note that a version of $\to\Rightarrow$, i.e., for implication with provenance R in the antecedent is by Lemma 3.vi, superfluous. For implication with provenance R in the succedent we use our new implication rule $\Rightarrow\to*$.[3]

$$\Rightarrow\to*\mathrm{R} \quad \frac{\Gamma \overset{H_1}{\Longrightarrow} \Delta, \mathsf{nh}(A)^{\mathsf{R}}, B^{\mathsf{R}} \qquad (\neg B)^{\mathsf{R}}, \Gamma \overset{H_2}{\Longrightarrow} \Delta, (\neg A)^{\mathsf{R}}}{\Gamma \overset{H_1 \land H_2}{\Longrightarrow} \Delta, (A \to B)^{\mathsf{R}}}$$

Rule $\Rightarrow\to*$R is the only rule that, read bottom-up, introduces nh into a sequent. The introduced occurrence has provenance R, which implies Lemma 3.iii. The occurrence is placed as a member of the succedent. Lemma 3.iv follows since there is no rule that would move a formula with nh as top-level operator from the succedent to an occurrence in the antecedent. The argument formula of the introduced occurrences of nh stems from the antecedent of an implication with provenance R. Since the given root succedent B^{R} is body-normalized, it follows that there is no implication occurring in this argument formula. This implies Lemma 3.v.

We observe that whenever our rules introduce a formula with provenance R into the antecedent of a premise, the formula is a negation. Since the root sequent has just a single formula with provenance L as antecedent, this implies Lemma 3.vi. The statements underlying the transfer of conditions (I1) and (I2) from the premises to the conclusion are for the implication rules as follows.

[3]It seems possible to use $\Rightarrow\to*$ instead of $\Rightarrow\to$ also for L-provenance, with appropriate modifications of other rules and axioms as well as Lemma 3. Our choice of $\Rightarrow\to$ for L-provenance tries to introduce nh not more than necessary.

$\rightarrow\Rightarrow$L, (I1)	If	$\neg A \wedge \bigwedge \Gamma^{\mathsf{L}} \models H_1 \vee \bigvee \Delta^{\mathsf{L}}$ and $\bigwedge \Gamma^{\mathsf{L}} \models H_2 \vee \bigvee \Delta^{\mathsf{L}} \vee A \vee \neg B$ and $B \wedge \bigwedge \Gamma^{\mathsf{L}} \models H_3 \vee \bigvee \Delta^{\mathsf{L}},$ then $(A \rightarrow B) \wedge \bigwedge \Gamma^{\mathsf{L}} \models (H_1 \vee H_2 \vee H_3) \vee \bigvee \Delta^{\mathsf{L}}.$
$\rightarrow\Rightarrow$L, (I2)	If	$\bigwedge \Gamma^{\mathsf{R}} \wedge H_1 \models \bigvee \Delta^{\mathsf{R}}$ and $\bigwedge \Gamma^{\mathsf{R}} \wedge H_2 \models \bigvee \Delta^{\mathsf{R}}$ and $\bigwedge \Gamma^{\mathsf{R}} \wedge H_3 \models \bigvee \Delta^{\mathsf{R}},$ then $\bigwedge \Gamma^{\mathsf{R}} \wedge (H_1 \vee H_2 \vee H_3) \models \bigvee \Delta^{\mathsf{R}}.$
$\Rightarrow\rightarrow$L, (I1)	If	$A \wedge \bigwedge \Gamma^{\mathsf{L}} \models (H_1 \vee \bigvee \Delta^{\mathsf{L}} \vee B)$ and $\neg B \wedge \bigwedge \Gamma^{\mathsf{L}} \models (H_2 \vee \bigvee \Delta^{\mathsf{L}} \vee \neg A),$ then $\bigwedge \Gamma^{\mathsf{L}} \models (H_1 \vee H_2) \vee \bigvee \Delta^{\mathsf{L}} \vee (A \rightarrow B).$
$\Rightarrow\rightarrow$L, (I2)	If	$\bigwedge \Gamma^{\mathsf{R}} \wedge H_1 \models \bigvee \Delta^{\mathsf{R}}$ and $\bigwedge \Gamma^{\mathsf{R}} \wedge H_2 \models \bigvee \Delta^{\mathsf{R}},$ then $\bigwedge \Gamma^{\mathsf{R}} \wedge (H_1 \vee H_2) \models \bigvee \Delta^{\mathsf{R}}.$
$\Rightarrow\rightarrow*$R, (I1)	If	$\bigwedge \Gamma^{\mathsf{L}} \models H_1 \vee \bigvee \Delta^{\mathsf{L}}$ and $\bigwedge \Gamma^{\mathsf{L}} \models H_2 \vee \bigvee \Delta^{\mathsf{L}},$ then $\bigwedge \Gamma^{\mathsf{L}} \models (H_1 \wedge H_2) \vee \bigvee \Delta^{\mathsf{L}}.$
$\Rightarrow\rightarrow*$R, (I2)	If	$\bigwedge \Gamma^{\mathsf{R}} \wedge H_1 \models \bigvee \Delta^{\mathsf{R}} \vee \mathsf{nh}(A) \vee B$ and $\neg B \wedge \bigwedge \Gamma^{\mathsf{R}} \wedge H_2 \models \bigvee \Delta^{\mathsf{R}} \vee \neg A,$ then $\bigwedge \Gamma^{\mathsf{R}} \wedge (H_1 \wedge H_2) \models \bigvee \Delta^{\mathsf{R}} \vee (A \rightarrow B).$

Further Axioms and Rules that are not Presented. The following axioms and rules also have to be included in our interpolating proof system. They are omitted in this presentation as they are straightforward.

- Axioms and rules for the truth value constants.
- Familiar rules for conjunction and disjunction. With the possible provenances of the principal formula these are 8 rules. Two of them are shown as examples in Sect. 4.
- 4 rules for $\neg\neg$.
- 12 rules for moving negation inward.[4]
- 1 rule for eliminating nh over negation, based on equivalence (3). By Lemma 3.iii and 3.iv, this is only needed for the principal formula in the succedent and with provenance R. To effect at bottom-up application a reduction of the number of logic operators, required to ensure termination, the rule incorporates elimination of the double negation, i.e., it is

$$\frac{(\neg A)^{\mathsf{R}}, \Gamma \overset{H}{\Longrightarrow} \Delta}{\Gamma \overset{H}{\Longrightarrow} \Delta, \mathsf{nh}(\neg A)^{\mathsf{R}}}$$

- 2 rules for moving nh inward, based on equivalences (4) and (5). By Lemma 3.iii and 3.iv, these are only needed for the principal formula in the succedent and with provenance R.
- By Lemma 3.iv the rule $\neg\mathsf{nh}\Rightarrow$ is superfluous.

5.4. Completeness

Completeness of our interpolating sequent system can be shown similarly as outlined by Mints [9]. Termination is ensured since, applied bottom-up, each of the rules either reduces the number of occurrences of binary logic operators or keeps that number but reduces the number of occurrences of unary operators. We note that whenever $\Gamma \overset{H}{\Longrightarrow} \Delta$ is a split-sequent that meets the conditions of Lemmas 3.iii–3.vi and Γ or Δ has a member that is not of the form A, $\neg A$, or $\mathsf{nh}(A)$, with A an atom, then some rule is applicable bottom-up to the split-sequent. For a leaf sequent $\Gamma \overset{H}{\Longrightarrow} \Delta$ to which no rule is applicable and which is no instance of an axiom we can construct a refuting model by assigning truth values to all atoms A as follows.

If $A \in \Gamma$, then $A := \mathsf{T}$,
else if $\mathsf{nh}(A) \in \Delta$, then $A := \mathsf{T}$,
else if $\neg A \in \Delta$, then $A := \mathsf{NF}$,
else $A := \mathsf{F}$.

[4] As noted in [15], there is a misprint in [9]: The correct premises of the $\Rightarrow\neg\vee$ rule are $\Gamma \Rightarrow \Delta, \neg A$ and $\Gamma \Rightarrow \Delta, \neg B$.

We obtain a truth value assignment under which $\bigwedge \Gamma$ has the value T, while $\bigvee \Delta$ has the value NF or the value F. Hence, the value of $\bigwedge \Gamma \rightarrow \bigvee \Delta$ must be F or NF but cannot be T.

Sine all our rules are invertible and sound, the formula $A \rightarrow B$ for the two interpolation arguments A and B is equivalent to the conjunction of $\bigwedge \Gamma \rightarrow \bigvee \Delta$ for all leaves of the proof tree that are not closed by an axiom. Thus, if one of these conjuncts has the value F or NF, it follows that the value of $A \rightarrow B$ can only be F or NF but not T.

6. Craig-Lyndon Interpolation for HT

We now turn our attention to the second stage of our Craig-Lyndon interpolation method for HT, the conversion of the interpolating NHNNF-formula formula C' obtained according to Theorem 1 to a Craig-Lyndon interpolant C that is a HT-formula. This conversion is based on the following theorem.

Theorem 4. *Let A be a HT-formula and let C' be a NHNNF-formula such that $A \models C'$. Then there exists a HT-formula C such that $A \models C$, $C \models C'$ and $\mathsf{voc}^{\pm}(C) \subseteq \mathsf{voc}^{\pm}(C')$. Moreover, such a HT-formula C can be effectively constructed from C'.*

Proof. We show the construction of a suitable HT-formula C from the NHNNF-formula C'. First C' is converted to an equivalent CNF, which can be done with the familiar conversions for classical propositional logic. Let $D = \mathsf{nh}(E_1) \lor \ldots \lor \mathsf{nh}(E_m) \lor F$, with $m \geq 0$, where F has no occurrences of nh, be an arbitrary clause of this CNF. It holds that $A \models D$. Hence $\neg\neg A \models \neg\neg D$. Hence there is a proof P of $\neg\neg A \models \neg\neg D$ in Mints' system extended with axiom Ax-nh-2 and rule $\neg\mathsf{nh}\Rightarrow$ defined in Sect. 3 (completeness of this system for the considered implication formulas follows with the model construction of Sect. 5.4). Note that axiom Ax-nh-1 is not required: since the root sequent is $\neg\neg A \Rightarrow \neg\neg D$, if no rule is applicable to a sequent $\Gamma \Rightarrow \Delta$, then each member of Γ, Δ is either of the form $\neg p$ or of the form $\mathsf{nh}(p)$, with p an atom. Let D' be D with all formulas of the form $\mathsf{nh}(E_i)$ replaced by $\neg E_i$. We modify the proof P to a proof P' of $\neg\neg A \models \neg\neg D'$ in Mints' original system as follows: (1) Throughout the proof replace all formulas of the form $\mathsf{nh}(E_i)$ with $\neg E_i$. (2) Readjust rule labels: Leaves that were an instance of Ax-nh-2 are now instances of Ax-1; applications of rule $\neg\mathsf{nh}\Rightarrow$ become applications of Mints' rule $\neg\neg\Rightarrow$. Proof P' thus justifies $\neg\neg A \models \neg\neg D'$. Since $A \models \neg\neg A$, it follows that $A \models \neg\neg D'$. Thus, $A \models D \land \neg\neg D'$. The entailed conjunction $D \land \neg\neg D'$ is

$$(\mathsf{nh}(E_1) \lor \ldots \lor \mathsf{nh}(E_m) \lor F) \land \neg\neg(\neg E_1 \lor \ldots \lor \neg E_m \lor F), \tag{14}$$

which is equivalent to

$$(\mathsf{nh}(E_1) \lor \ldots \lor \mathsf{nh}(E_m) \lor F) \land (\neg E_1 \lor \ldots \lor \neg E_m \lor \neg\neg F), \tag{15}$$

which is equivalent to the HT-formula

$$E_1 \land \ldots \land E_m \rightarrow F. \tag{16}$$

Hence, the conjunction of the implications $E_1 \land \ldots \land E_{m_j} \rightarrow F_j$ for all clauses $D_j = \mathsf{nh}(E_1) \lor \ldots \lor \mathsf{nh}(E_{m_j}) \lor F_j$ of the CNF of C' provides the HT-formula C as claimed in the theorem statement. ☐

Although the proof of Theorem 4 refers to both formulas A and C', the *construction* of C is actually just from C', independently from A. The sequent proofs P and P' referenced in the proof of the theorem are for the purpose of justifying properties of C but not used in the actual construction of C.

Theorem 4 is in essence an adaptation of the final step in the proof of Theorem 11 in [4]. In this step, an LP-interpolant H that encodes a logic program is obtained from a Craig-Lyndon interpolant H' for classical formulas that encode logic programs as $H := H' \land \mathsf{rename}_{0 \mapsto 1}(H')$.

We are now ready to state the overall result, Craig-Lyndon interpolation for HT on the basis of Mints' sequent system.

Theorem 5. *Let A, B be HT-formulas such that $A \models B$. Then there exists a HT-formula C such that*

1. $A \models C$ *and* $C \models B$, *and*
2. $\mathsf{voc}^{\pm}(C) \subseteq \mathsf{voc}^{\pm}(A) \cap \mathsf{voc}^{\pm}(B)$.

Moreover, such a HT-formula C can be effectively constructed from a proof of $A \models B$ in a variation of Mints' sequent system for HT.

Proof. Follows from Theorems 1 and 4. $\square$

7. Conclusion

We have introduced a method to construct Craig-Lyndon interpolants for HT, which proceeds in two stages, extraction of a preliminary interpolant in an extended logic with Maehara's method, followed by a conversion to an interpolant in HT. The method is an adaptation of an encoding-based interpolation method for HT implemented by automated first-order provers [4]. Our sequent calculus reformulation aims at linking to research threads in proof theory and supporting potential applications where the focus are the proofs themselves, for example as explanations, in contrast to the interpolating formulas.

A recent observation [16] relates to the focus on proofs: Maehara's method can be refined to *proof relevant interpolation*, a way of cut introduction, or proof splitting. As our preliminary interpolant is obtained with Maehara's method, this refinement should transfer to our first stage. Incorporation and apparent necessity of the second stage, the conversion to HT, has to be investigated.

For HT, the distinction between *positive* and *negative* polarity underlying Craig-Lyndon interpolation suggests refinement. If a HT formula is normalized to a "logic program" (a conjunction of implications $A \rightarrow B$, with A a conjunction of literals and B a disjunction of literals), then *four* roles may be distinguished for an atom occurrence: positively in A, negated in A, positively in B, and negated in B. It seems that roles of atom occurrences in interpolants can be ensured to be in certain relationships with the roles in the interpolated formulas, similarly as discussed for Beth definability in [4, Sect. 3.4].

Two steps in our interpolation method for HT involve formula transformations that seem not to be polynomially restricted: Lemma 3 requires the second interpolation input to be *body-normalized*, and the postprocessing according to Theorem 4 involves transformation to CNF. The actual impact of this is not clear. Logic programs are typically already in body-normalized form. Possibly there are more sensible transformations such as a relaxed body-normalized condition and an adaptation of the Tseitin encoding. For methods avoiding non-polynomial formula transformations, the question arises whether effort has just been passed to proving, with proof steps now performing the work of the transformation.

Acknowledgments

The author thanks David Pearce for the conversations that led to this work. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 457292495.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] V. Lifschitz, D. Pearce, A. Valverde, Strongly equivalent logic programs, ACM Trans. Comp. Log. 2 (2001) 526–541. doi:10.1145/383779.383783.

[2] D. M. Gabbay, D. Pearce, A. Valverde, Interpolable formulas in equilibrium logic and answer set programming, JAIR 42 (2011) 917–943.

[3] D. Pearce, A. Valverde, Synonymous theories and knowledge representations in answer set programming, J. Comput. Syst. Sci. 78 (2012) 86–104. doi:10.1016/J.JCSS.2011.02.013.

[4] J. Heuer, C. Wernhard, Synthesizing strongly equivalent logic programs: Beth definability for answer set programs via Craig interpolation in first-order logic, in: C. Benzmüller, M. Heule, R. Schmidt (Eds.), IJCAR 2024, volume 14739 of *LNCS (LNAI)*, Springer, 2024, pp. 172–193. doi:10.1007/978-3-031-63498-7_11.

[5] L. L. Maksimova, Craig's theorem in superintuitionistic logics and amalgamable varieties of pseudo-Boolean algebras, Algebra and Logic 16 (1977) 427–455. doi:10.1007/BF01670006.

[6] L. L. Maksimova, Interpolation properties of superintuitionistic logics, Studia Logica 38 (1979) 419–428. doi:10.1007/BF00370479.

[7] D. M. Gabbay, L. L. Maksimova, Interpolation and Definability: Modal and Intuitionistic Logics, Oxford Univ. Press, 2005. doi:10.1093/acprof:oso/9780198511748.001.0001.

[8] M. Baaz, H. Veith, Interpolation in fuzzy logic, Arch. Math. Logic 38 (1999) 461–489. doi:10.1007/s001530050151.

[9] G. Mints, Cut-free formulations for a quantified logic of here and there, Ann. Pure Appl. Log. 162 (2010) 237–242. doi:10.1016/J.APAL.2010.09.009.

[10] K. Bendová, Interpolation and three-valued logics, Reports Math. Log. 39 (2005) 127–131. URL: https://rml.tcs.uj.edu.pl/rml-39/bendova.pdf.

[11] S. Maehara, On the interpolation theorem of Craig (Japanese), Sugaku 12 (1960) 235–237.

[12] G. Takeuti, Proof Theory, second ed., North-Holland, 1987.

[13] R. M. Smullyan, First-Order Logic, Springer, 1968. Also republished with corrections by Dover publications, 1995.

[14] A. S. Troelstra, H. Schwichtenberg, Basic Proof Theory, second ed., Cambridge University Press, 2000.

[15] J. Otten, T. Schaub, Implementing the first-order logic of here and there, in: M. Gebser, D. Inclezan, F. Ricca, M. Carro, M. Truszczynski (Eds.), ICLP 2025, volume 439 of *EPTCS*, 2026, pp. 453–468. doi:10.4204/EPTCS.439.31.

[16] A. Saurin, Interpolation as cut-introduction: On the computational content of Craig-Lyndon interpolation, in: M. Fernández (Ed.), FSCD 2025, LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, pp. 32:1–32:21. doi:10.4230/LIPICS.FSCD.2025.32.