Annals of Pure and Applied Logic 169 (2018) 1369–1418

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

Multicomponent proof-theoretic method for proving interpolation properties

Roman Kuznets [1]

Technische Universität Wien, Austria

ARTICLE INFO

Article history:
Available online 30 August 2018

MSC:
03B45
03C40
03F03
03F07

Keywords:
Craig interpolation
Lyndon interpolation
Structural proof theory
Hypersequent
Labelled sequent
Modal logic

ABSTRACT

Proof-theoretic method has been successfully used almost from the inception of interpolation properties to provide efficient constructive proofs thereof. Until recently, the method was limited to sequent calculi (and their notational variants), despite the richness of generalizations of sequent structures developed in structural proof theory in the meantime. In this paper, we provide a systematic and uniform account of the recent extension of this proof-theoretic method to hypersequents, nested sequents, and labelled sequents for normal modal logic. The method is presented in terms and notation easily adaptable to other similar formalisms, and interpolant transformations are stated for typical rule types rather than for individual rules.

1. Introduction

Interpolation was called "about *the last significant property of first-order logic* that has come to light"[2] (Van Benthem [5]). The property was first formulated and proved by Craig in [17,18] and was inspired by his post-publication review [16] of Beth's paper [6] on definability.

The *Craig interpolation property* for the logic of a given class of models, or *CIP* for short, states roughly that any logical consequence $A \vDash B$ can be supplied with an intermediary statement C, called an *interpolant*,

E-mail address: roman@logic.at.
URL: https://sites.google.com/site/kuznets/.

[1] The author was supported by the Austrian Science Fund (FWF) grants M 1770-N25, Y 544-N23, and S 11405-N23 (RiSE/SHiNE).

[2] Emphasis by Van Benthem.

https://doi.org/10.1016/j.apal.2018.08.007

that sits between A and B in terms of logical consequence, i.e., satisfies $A \vDash C$ and $C \vDash B$, and uses only the language elements common to A and B. This formulation is not entirely formal as one needs to specify which elements of the language need to be common. Alternatively, if the logic is defined syntactically rather than semantically, one can use $D \vdash E$ instead of $D \vDash E$. Since in most standard logical languages it is possible to define an implication that satisfies modus ponens and the deduction theorem, the usual formulation of the CIP uses $\vdash D \to E$ instead of $D \vdash E$.

Similar to decidability, the interpolation property is a desirable but not necessary property of logics. Similar to decidability, the CIP fails in some reasonably fundamental logics, e.g., the (predicate) intuitionistic logic of constant domains, CD (Mints et al. [51]). On the other hand, many less standard logics often possess the property. As with decidability, for logics that fail the Craig interpolation property, it can sometimes be weakened to a version that holds, while for other logics, it can be strengthened. We do not intend to discuss the complex and intricate hierarchy of interpolation properties, which becomes all the more interesting for weaker logics, where distinctions appear between various formulations of the CIP that are equivalent for classical propositional logic. Those interested are referred to the monograph by Gabbay and Maksimova [26], where the interpolation hierarchy is explored in relation to the corresponding hierarchy of algebraic amalgamation properties.

The only alternative variant of the CIP we will consider in this paper is the *Lyndon interpolation property*, or the *LIP* for short. It was introduced by Lyndon [43] shortly after Craig's original publication.

To explain the difference, we first need to clarify the notion of common language for logics discussed in this paper. As the logics we consider are predominantly monomodal logics based on classical propositional reasoning, with an occasional mention of intermediate logics, the natural definition of the common-language requirement on interpolants of A and B is that they only contain *atomic propositions* common to A and B.

Example 1.1. $P \wedge Q$ is an interpolant of $A = P \wedge Q \wedge S$ and $B = Q \vee \neg P$ for all the logics we consider because

- $\vdash P \wedge Q \wedge S \to P \wedge Q$,
- $\vdash P \wedge Q \to Q \vee \neg P$, and
- both atomic propositions P and Q occurring in the interpolant $P \wedge Q$ occur both in $P \wedge Q \wedge S$ and in $Q \vee \neg P$.

Lyndon suggested discounting $P \wedge Q$ from being an interpolant on the basis that P is not really playing the same role in A and B: P is present positively in A but negatively in B. Indeed, in the example above, it is clear that the transition from A to B has little to do with P and that Q would equally well play the role of a Craig interpolant. Unlike $P \wedge Q$, the formula Q is not only a Craig but also a Lyndon interpolant because the only atomic proposition occurring in Q, Q itself, occurs positively there, as well as in A and B.

The polarity of a subformula occurrence in a given formula is a standard notion related to its monotonicity/antimonotonicity with respect to logical consequence. In all the logics we consider, if any positive occurrence of B in $A(B)$ is replaced with a formula C such that $\vdash B \to C$, then $\vdash A(B) \to A(C)$. Conversely, if any negative occurrence of B in $A(B)$ is replaced with C such that $\vdash B \to C$, then $\vdash A(C) \to A(B)$. Practically, it means that each formula is a positive subformula of itself, the conjunction, disjunction, and modalities $\Box$ and $\Diamond$ do not change the polarity of subformula occurrences, e.g., a positive occurrence of B in $A(B)$ remains positive in $A(B) \wedge D$. The negation flips the polarity of all occurrences within, e.g., a negative occurrence of B in $A(B)$ becomes a positive occurrence in $\neg A(B)$. Finally, the implication preserves the polarity of subformula occurrences in its consequent and flips those in its antecedent, e.g., if B occurs positively in $A(B)$ and C occurs negatively in $D(C)$, both become negative occurrences in $A(B) \to D(C)$.