Contribution Title

First Author[1[0000−1111−2222−3333]], Second Author[2,3[1111−2222−3333−4444]], and Third Author[3[2222−−3333−4444−5555]]

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. Tableaux proofs are used in automated "proofers" as (TODO), whereas sequent calculus proofs are used in proof assistants as (TODO). This work aims to discuss the translation between them and provide an algorithm for translating first-order, predicate logic tableaux proofs into sequent calculus proofs. It begins with an overview of the definitions in both intuitionistic and classical logic. It then shows a translation process in classical logic, along with its properties. Finally, a potential extension towards translation in intuitionistic logic is explored. (TODO talk about book1, "extention")

Keywords: Tableux proof · sequent calculus · intuitionistic logic.

1 Introduction

1.1 Notation

In this work, sentences will implicitly refer to first-order predicate logic sentences; for intuitionistic logic, their meaning will come from Kripke's semantics [3] . The notation for structures and frames will be (heavily based on ?) / (as in) [1]. To make this document slightly more self-reliant, we will briefly explain:

1.2 Definitions

The definitions will be given for intuitionistic logic; To avoid redundancy, classical logic will be seen as intuitionistic logic with single-framed structures. [TODO cite?]

Definition 1. ***A structure of a language*** *consists of a domain and an assignment from the constant symbols of the language to the domain and from the predicate symbols of the language to predicates in the domain.*

They represent "possible worlds" [TODO cite] in a frame:

Definition 2. ***A Kripke Frame*** *of a Language* $\mathcal{L}$, $\mathcal{C} = (R, \{C(p)\}_{p \in R})$ *consists of a partially ordered set* R*, and an* $\mathcal{L}$*-structure* $C(p)$ *for all* p*'s in* R*. Furthermore, in a Kripke Frame: if* $p \leq q$ *, then* $C(q)$ *extends* $C(p)$*: all sentences that are true in* $C(p)$ *are true in* $C(q)$ *, the domain of* $C(p)$ *is included in the domain of* $C(q)$ *and the assignments in* $C(p)$ *are the same as in* $C(q)$

For simplicity, R will always be the set of sequences of integers, and $p \leq q$ if p is in q:

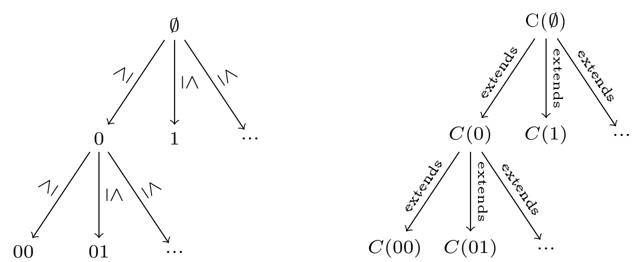

Fig. 1. R and a Kripke frame

Also, from now on, the constant elements of our language will be ordered $\{c_0, c_1, c_2...\}$

Definition 3. ***Forcing****. When a sentence* ϕ *of a language* $\mathcal{L}$ *is* ***forced*** *by a structure* $C(p)$ *of a frame* $\mathcal{C}$*, we denote:* $p \vDash_{\mathcal{C}} \phi$
Forcing is defined by induction: [1]

- $p \vDash_{\mathcal{C}} \phi \Leftrightarrow \phi$ *is true in* $C(p)$ *(if* ϕ *is an atomic sentence)*

- $p \vDash_{\mathcal{C}} (\phi \to \psi) \Leftrightarrow$ *for all* $q \geq p$*, if* $q \vDash_{\mathcal{C}} \phi$*, then* $q \vDash_{\mathcal{C}} \psi$
- $p \vDash_{\mathcal{C}} \neg\phi \Leftrightarrow$ *for all* $q \geq p$*,* q *does not force* ϕ

- $p \vDash_{\mathcal{C}} (\forall x)\phi(x) \Leftrightarrow$ *for all* $q \geq p$ *and* d *in* $\mathcal{L}_{C(q)}$*,* $q \vDash_{\mathcal{C}} \phi(d)$

- $p \vDash_{\mathcal{C}} (\exists x)\phi(x) \Leftrightarrow$ *exists a* d *in* $\mathcal{L}_{C(q)}$*, such that* $p \vDash_{\mathcal{C}} \phi(d)$

- $p \vDash_{\mathcal{C}} (\phi \wedge \psi) \Leftrightarrow p \vDash_{\mathcal{C}} \phi$ *and* $p \vDash_{\mathcal{C}} \psi$

- $p \vDash_{\mathcal{C}} (\phi \vee \psi) \Leftrightarrow p \vDash_{\mathcal{C}} \phi$ *or* $p \vDash_{\mathcal{C}} \psi$

[TODO extended language ?]

Definition 4. ***Truth*** *A sentence is Intuitionistically valid if it is forced in all structures of all Kripke frames of that the sentence's language.*

In classical logic, this definition simplifies to the one of forcing, and it's simplified again by the fact that p = q; in fact, we can define classical validity as: [1]

Definition 5. ***Truth*** *A sentence is classically valid if it is intuitionistically valid in all single-sentenced Kripke frames of that sentence's language.*

1.3 Classical tableaux

Considerations Here We first define a slightly different version of the destructive tableaux proof tree described by [1], where each node is a signed sentence / truth assertion . This will allow for a more implementation-oriented approach and the translation later on.

The correspondence of the destructive tableaux proof tree described in [1] to our new one is shown in figure [TODO].

Generally speaking, a node in the usual definition is replaced by a sequence of all nodes in the path that goes from the root to it. Afterwards, some nodes are removed from the newly formed tableaux by adjoining its son(s) and its parent. A node should be removed if its corresponding node in the original tableaux was not a leaf of the atomic tableaux that introduced it.

Classical tableaux The classical tableaux stands on pas mal de defintions, well try to justifiy them briefly:

Definition 6. ***A Signed sentence*** *is a forcing assertion inside of a tableaux proof. It looks like $T_q\phi$ or $F_p\phi$* ***A Signed sentence list*** *is a list forcing assertions inside of a tableau proof. We say that a list of forcing assertions $\{T_{p_1}\phi_1, T_{p_2}\phi_2, ...F_{p_i}\phi_i, F_{p_{i+1}}\phi_{i+1}...\}$ is "valid" if there exists a frame for witch $\mathcal{C}(p_1) \vDash \phi_1$ and $\mathcal{C}(p_2) \vDash \phi_2$ and ... $\mathcal{C}(p_i) \nvDash \phi_i$ and $\mathcal{C}(p_{i+1}) \nvDash \phi_{i+i}$*

A Signed sentence list will denote a existance assumption that may or may not be valid. We can infer other assumptions that are, by definition, consequences and necessities (?) of a given assumption. The function f, defined bellow, is one of the ways we can do that:

Definition 7. *The function f takes a signed sentence σ and a signed sentence list L and returns one or two signed sentence lists. It is defined as follows: (here we denote $l||l' = l_1, l_2..., l_{|l|}, l'_1, l'_2.., l'_{|l'|}$)*

- $f(T\neg\alpha, h||L) = \{L||h||F\alpha\}$.
- $f(F\neg\alpha, h||L) = \{L||h||T\alpha\}$.
- $f(T(\alpha \wedge \beta), h||L) = \{L||h||T\alpha||T\beta\}$.
- $f(F(\alpha \wedge \beta), h||L) = \{L||h||F\alpha, L||h||F\beta\}$.
- $f(T(\alpha \vee \beta), h||L) = \{L||h||T\alpha, L||h||T\beta\}$.
- $f(F(\alpha \vee \beta), h||L) = \{L||h||F\alpha||F\beta\}$.
- $f(T(\alpha \rightarrow \beta), h||L) = \{L||h||F\alpha, L||h||T\beta\}$.
- $f(F(\alpha \rightarrow \beta), h||L) = \{L||h||T\alpha||F\beta\}$.

- $f(T(\forall x)\phi(x), h||L) = \{L||h||T\phi(c_i)\}$ *for the first constant* c_i *for which* $\phi(x)$ *is not in* L.
- $f(F(\forall x)\phi(x), h||L) = \{L||h||F\phi(c_i)\}$ *for the first constant* c_i *not present in* $h||L$.
- $f(T(\exists x)\phi(x), h||L) = \{L||h||T\phi(c_i)\}$ *for the first constant* c_i *not present in* $h||L$.
- $f(F(\exists x)\phi(x), h||L) = \{L||h||F\phi(c_i)\}$ *for the first constant* c_i *for which* $\phi(x)$ *is not in* L.

Of course, this inference must be "well behaved":

Theorem 1. *Given a signed sentence list x, x is valid if and only if one of the sentence lists of f(x) is valid*

Proof. It is truth directly by the definition of forcing. Some special attention must be given to the choice of varibles and structures: When chosing [TODO]

Definition 8. ***The Tableux development of a sentence*** *is defined inductivelly:*

- *A tree with the single node* $\{F\phi\}$ *is a tableaux development of* ϕ.
- *If* τ *is a tableaux development of* ϕ, *then* $\hookleftarrow(\sigma, \tau)$ *is a tableaux development of* ϕ. *Where:*

$$\hookleftarrow(\sigma, \tau) = \tau \textit{ with } f(\sigma, l) \textit{added to all leaves } l \textit{ that contain } \sigma$$

Theorem 2. *"if there exists a frame that does not force* ϕ *then one of the leaves of* $\hookleftarrow(\sigma_1, \hookleftarrow(\sigma_2(.....(\hookleftarrow(\sigma_n, F\phi)...))))$ *is valid."*

Proof. The proof goes by induction:

The base case is true by definition. Next we assume $\tau' = \hookleftarrow(\sigma, \tau)$. There are two cases to consider:

- If there does not exist a frame that does not force ϕ, The theorem is valid for τ'.
- If there exists a frame that does not force ϕ:
  By the induction hypothesis, there exists a valid leaf σ in τ.
  - If σ is a leaf in τ, then the theorem is valid for τ'.
  - If σ is not a leaf in τ, then: One or two nodes were added to σ in τ' by the definition of $\hookleftarrow$. By the property [TODO] of f, one the added nodes are also valid. Consequently, the theorem is valid for τ'.

Theorem 3. *If all leaves of* $\hookleftarrow(\sigma_1, \hookleftarrow(\sigma_2(.....(\hookleftarrow(\sigma_n, F\phi)...))))$ *are contradictory, then* ϕ *is intuitionistically valid*

Proof. If ϕ is not intuitionistically valid, there exists a Kripke frame $\mathcal{C}$ and a world p such that $p \nVdash_{\mathcal{C}} \phi$. By the tableaux development process, at least one leaf remains valid. This contradicts the assumption that all leaves are contradictory. Hence, ϕ is intuitionistically valid.

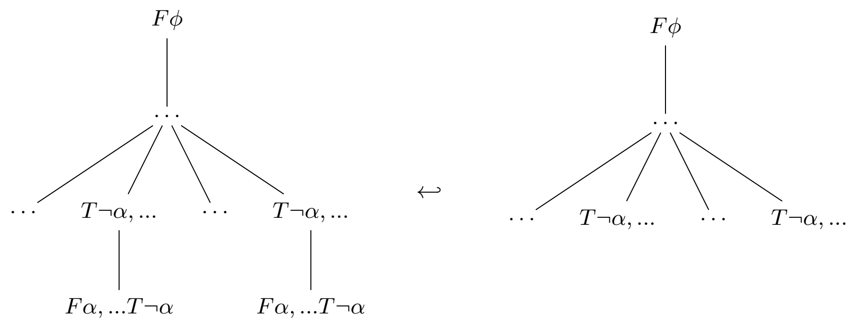

Fig. 2. Example of $\hookleftarrow (T\neg\alpha, \tau)$ and τ

Sequent calculus Here we use the multi-conclusion sequent calculus defined by [2]. We also use a Kripke semantics oriented approach to multi-conclusion sequent calculus, presented in [4] [TODO].

Definition 9. ***Sequent*** *A sequent is an expression of the form*

$$\Gamma \vdash \Delta$$

where $\Gamma = \{\Gamma_1, \Gamma_2, \Gamma_3, ...\}$ *and* $\Delta = \{\Delta_1, \Delta_2, \Delta_3 ...\}$ *are finite sets of formulas.* Γ *is called the antecedent, and* Δ *is called the succedent. [TODO cite]*

Γ represents multiple necessary hipotesis, while Δ represents multiple possible conclusions.

Definition 10. *A sequent is non-valid if there exists a frame that forces all sentences in* Γ *and does not force any sentence in* Δ*. A sequent is valid if not non-valid.*

Not a theorem neither a definition but maybe something:

Theorem 4. *Given a valid signed sentence list* $\{T_{p_1}\phi_1, T_{p_2}\phi_2, ...F_{p_i}\phi_i, F_{p_{i+1}}\phi_{i+1}...\}$ *and a frame w, if* $\{T_{q_1}\gamma_1, T_{q_2}\gamma_2, ...F_{s_1}\delta_1, F_{s_2}\delta_2...\}$ *is the list obtained by filtering the truth assertions of type T extended by w filtering the truth assertions of type F with frames extended by w , then* $\gamma_1, \gamma_2, ... \vdash \delta_1, \delta_2, ...$ *is not valid*

Proof. A sequence of signed lists $\{T_{p_1}\phi_1, T_{p_2}\phi_2, ...F_{p_i}\phi_i, F_{p_{i+1}}\phi_{i+1}...\}$ inside a tableaux proof signifies an assumption of existance of a frame C for witch $\mathcal{C}(p_1) \vDash \phi_1$ and $\mathcal{C}(p_2) \vDash \phi_2$ and ... $\mathcal{C}(p_i) \nvDash \phi_i$ and $\mathcal{C}(p_{i+1}) \nvDash \phi_{i+i}$ witch implicitly means, by rules [TODO] of the definition of forcing:

($\mathcal{C}(p) \vDash \phi_1$ for all $p \geq p_1$) and ($\mathcal{C}(p) \vDash \phi_2$ for all $p \geq p_2$) and ... ($\mathcal{C}(p) \nvDash \phi_i$ for all $p \leq p_i$) and ($\mathcal{C}(p) \nvDash \phi_{i+1}$ for all $p \leq p_{i+1}$)

take a structure w inside of $\mathcal{C}$ for witch we can infer: $w \vDash \gamma_1$ and $w \vDash \gamma_2$ and ... $w \nvDash \delta_1$ and $w \nvDash \delta_2$.... w is a counterexample proving the non-validity of $\gamma_1, \gamma_2, ... \vdash \delta_1, \delta_2, ...$

References

1. Nerode, A., Shore, R.A.: Logic for Applications. 2nd edn. Texts in Computer Science. Springer, New York, NY (1997). https://doi.org/10.1007/978-1-4612-0649-1
2. Dummett, M.: Elements of Intuitionism. Oxford Logic Guides. Clarendon Press (2000). https://books.google.fr/books?id=JVFzknbGBVAC
3. Kripke, S.A.: Semantical analysis of intuitionistic logic I. In: Studies in Logic and the Foundations of Mathematics. Elsevier (1965)
4. Svejdar, V.: On Sequent Calculi for Intuitionistic Propositional Logic. CMUC (2005). https://www.muni.cz/en/research/publications/1234567
5. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
6. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
7. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
8. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
9. LNCS Homepage, http://www.springer.coBm/lncs, last accessed 2023/10/25