universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

"Automated Deduction for Intuitionistic Logic via Embedding into Classical Logic"

verfasst von / submitted by

Alexander Pluska, BSc

angestrebter akademischer Grad / in partial fulfillment of the requirements for the degree of

Master of Science (MSc)

Wien, 2022 / Vienna, 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 821
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Mathematik
Betreut von / Supervisor:	Univ.-Prof. Matthias Aschenbrenner, PhD

Abstract

The famous double negation translation [16, 18] establishes an embedding from classical into intuitionistic logic. Curiously, the reverse direction has not been covered in literature. We present an effective embedding from intuitionistic into classical logic, both in the propositional and first-order case, as well as an effective embedding of intuitionistic propositional logic into quantified boolean formulas.

Furthermore, we implement a system that takes intuitionistic first-order problems in the tptp file format as an input and performs our transformation. This allows the use of classical theorem provers for checking intuitionistic validity. We benchmark our implementation using the Vampire theorem prover [21] on the ILTP problem set [4].

Finally, we discuss how the generated classical proofs of the transformed problem can be translated back into intuitionistic proofs of the original problem. All in all, this establishes a novel approach to theorem proving for intuitionistic logic and provides a first proof of concept.

Acknowledgements

I want to thank Matthias for accommodating my particular interests by agreeing to supervise a thesis which lies outside his main area of research and bearing with the organizational difficulties.

I sincerely appreciate Florian's guidance, who graciously agreed to co-supervise a thesis at a different university. He has provided me with a fascinating topic as well as plenty of his time, expertise and thorough feedback.

Finally, I am incredibly grateful to my family, in particular my parents Stefania and Cezary, who have instilled in me the joy of learning, provided all the love and help I could ask for and invariably supported me along my journey.

Contents

Chapter 1

Introduction

Constructive mathematics refers to a flavour of mathematics in which the existence of an object can only be established by explicit construction, as opposed to classical mathematics where existence can be shown implicitly, e.g. by assuming non-existence and deriving a contradiction. The formalism usually associated with constructive mathematics is intuitionistic logic, which essentially differentiates itself from classical logic by the fact that the law of excluded middle $A \vee \neg A$ and the double negation shift $\forall x \neg\neg P(x) \to \neg\neg\forall x P(x)$ are not valid. Besides philosophical considerations, most prominently advocated by Brouwer [11] and Bishop [10], there is a particular motivation for studying constructive mathematics from the perspective of computer science in that proofs in intuitionistic logic directly correspond to computer programs — as expressed in the Curry-Howard correspondence [19].

The interest in intuitionistic logic has led to the development of a number of automated theorem proving systems for propositional as well as for predicate logic and a collection of benchmark problems (see e.g. the ILTP library website [4]). However, progress in automated theorem proving for intuitionistic logic has been slow, whereas reasoners for classical logic have made tremendous progress, see e.g. the TPTP [1] and SAT [3] competitions. This difference can be partially explained by fundamental differences between the logics. Foremost, determining intuitionistic validity is computationally harder, i.e. in the propositional case intuitionistic validity is `PSPACE`-complete [25] whereas classical validity is `coNP`-complete [12]. A further advantage of classical logic is the existence of calculi that are particularly suitable for automation such as superposition [8], which rely on the existence of convenient normal forms such as CNF, and the duality between validity and satisfiability, i.e. to show the validity of a formula it suffices to show the unsatisfiability of the negated formula. While some (albeit more complex) normal forms also exist for intuitionistic logic, crucially the duality between validity and satisfiability of the negation does not hold. Therefore, most dedicated intuitionistic theorem provers [23, 26] use the naive inverse method, i.e. direct search for a cut-free proof by applying the rules from some proof calculus inversely, enhanced by search strategies such as focussing and polarization. This approach generally leads to a much more complex search and is therefore difficult to implement efficiently. Finally, we add that in contrast to intuitionistic provers a tremendous amount of work has been put into optimizing provers for classical logic, in particular for the propositional case, i.e. SAT-solvers.

With this work we want to propose a novel approach for intuitionistic theorem proving that leverages the progress in classical theorem proving, that is for each formula φ:

- Give a formula $\varphi^\#$ that is classically valid if and only if φ is intuitionistically valid.
- Use a state-of-the-art classical prover to establish the validity/invalidity of $\varphi^\#$.
- Transform the generated proof/counter-model of $\varphi^\#$ to one of φ.

The most challenging part of this approach is giving the translation of φ to $\varphi^\#$. Interestingly, the reverse direction, the famous double-negation translation, has long been established and goes back to Glivenko [16] in the propositional case, and to Gödel [18] and Gentzen [15] in the first-order case. In the propositional case, it is particularly simple: φ is classically valid if and only if $\neg\neg\varphi$ is intuitionistically valid. Intuitively, the translation collapses for each subformula ψ of φ the truth values of ψ and $\neg\neg\psi$, which are classically but not intuitionistically equivalent. This gives us a first idea why the reverse direction is perhaps more difficult: We need to expand the truth values of ψ and $\neg\neg\psi$, i.e. if they both occur in φ, we must have a way to (classically) assign different truth values to their respective counterparts in $\varphi^\#$. In particular, this necessitates the introduction of new propositional variables in our procedure, which already marks a big difference to the double negation translation.

While establishing the translation of formulas we also perform an effective translation of counter-models, i.e. for each intuitionistic counter-model of φ we effectively construct a classical counter-model of $\varphi^\#$ and vice versa. We note that the existence of counter-models is a key notion that forms a proper dual to validity — whereas the satisfiability of the negation is not necessary for invalidity (in contrast to classical logic, where it is a proper dual to validity). Transforming and reducing counter-models to a normal form is also what ultimately enables our translation. Apart from a translation of counter-models in the final chapter, we also explicitly describe how to equivalently transform classical proofs of $\varphi^\#$ to intuitionistic proofs of φ.

As a final contribution, we implement our translation in the Rust programming language. Our implementation transforms a first-order problem in the tptp format [6] and outputs the translated problem in the same format. The code is published on GitHub [2]. We then benchmark our implementation using the ILTP problem set [4], i.e. we translate all problems in the set and then run the Vampire theorem prover [22] on the translated problems. Our approach performs on par with existing approaches for intuitionistic theorem proving, but comes short of the state-of-the art. As there is still a lot of room for optimization, this is a hopeful first sign.

Chapter 2

Overview

The main content is organized in chapters 3 to 6. Chapter 3 collects prerequisite definitions and serves to fix our notation and recall some important facts. Chapter 4 is the meat of this thesis. In particular, it features a complete description of our embedding of intuitionistic into classical logic as well as proving its correctness. In chapter 5 we present the implementation of the embedding and the results of our benchmark. Finally, in chapter 6 we explicitly describe how to translate the generated proofs and counter-models which in principle directly follows from the results of section 4. We shall now for each chapter give an overview of our main results and highlight the key arguments.

2.1 Embedding intuitionistic into classical logic

Recall that our goal in this chapter is to give an effective translation procedure that, for a given formula φ, yields a formula $\varphi^{\#}$, such that φ is intuitionistically valid if and only if $\varphi^{\#}$ is classically valid. We start with the propositional case. The key arguments are the same as in the first-order case, while it is technically simpler and less cluttered.

Before our main transformation we employ a preprocessing step akin to the Tseytin transformation [28], which is a popular pre-processing step in classical automated reasoning: It gives an equisatisfiable sentence (over an extended set of propositions) in conjunctive normal form. Eliminating all implications, however, is not possible in intuitionistic logic since $A \to B$ is not equivalent to $\neg A \lor B$. Still, we propose a similar transformation. A notable feature of our transformation is that all non-classical content is encapsulated in formulas of type $(A \to B) \to C$, i.e. if there are no such formulas in the transformed formula then the classical validity of the transformed formula immediately implies intuitionistic validity and no further processing is needed.

Theorem 2.1.1. For every propositional formula φ there effectively are an atom P and a set of $\mathcal{S}$ of formulas (over an extended set of propositions) of one of the forms

$$A \to (B \land C), (A \land B) \to C, A \to (B \lor C), (A \lor B) \to C, (A \to B) \to C,$$

for atomic A, B, C, such that φ is intuitionistically valid if and only if $\bigwedge \mathcal{S} \to P$ is intuitionistically valid. The time complexity of the transformation is linear in the input size.

The main transformation then proceeds in three steps: 1) We encode as a first-order sentence that the considered formula holds for every Kripke frame. Since Kripke frames for propositional logic over a fixed alphabet of propositional variables form a first-order theory, this step is rather straightforward. 2) Next we eliminate some quantifiers via Herbrandization. The only quantifiers eliminated occur in formulas derived from formulas of type $(A \to B) \to C$. 3) We can then argue that if there is a counter-model for the resulting formula, then there is a counter-model with a certain Kripke frame of bounded size completely determined by the number of formulas of type $(A \to B) \to C$. This allows us to eliminate the remaining quantifiers by simply enumerating the worlds in that Kripke frame.

Summarizing, we obtain the following result:

Theorem 2.1.2. Let $\mathcal{S}$ be as in Theorem 2.1.1 and $\mathcal{F}_{\to} \subseteq \mathcal{S}$ denote the subset of formulas of the form $(A \to B) \to C$ and Λ denote the set of sequences without repetition over $\mathcal{F}_{\to}$. For each atom A and $k \in \Lambda$ consider a new atom A^k. Obtain $\mathcal{S}^{\#}$ by including the following formulas:

- $A^k \to A^{k\psi}$ for each atom A occurring in $\mathcal{S}$, $k \in \Lambda$ and $\psi \in \mathcal{F}_{\to}$ not occurring in k.
- $A^k \to (B^k \circ C^k)$ for each $\circ \in \{\wedge, \vee\}$, $A \to (B \circ C) \in \mathcal{S}$, $k \in \Lambda$.
- $(A^k \circ B^k) \to C^k$ for each $\circ \in \{\wedge, \vee\}$, $A \to (B \circ C) \in \mathcal{S}$, $k \in \Lambda$.
- $(A^{k\psi} \to B^{k\psi}) \to C^k$ for $\psi = (A \to B) \to C \in \mathcal{S}$, $k \in \Lambda$ if ψ does not occur in k.

Then, $\bigwedge S \to P$ is intuitionistically valid iff $\bigwedge S^{\#} \to P^{\epsilon}$ is classically valid, where ϵ denotes the empty sequence. The size of $\mathcal{S}^{\#}$ is in $\mathcal{O}(|\mathcal{S}| \cdot 2^{|\mathcal{F}_{\to}| \cdot \log(|\mathcal{F}_{\to}|)})$. Furthermore, there is an effective procedure for translating counter-models between the sentences.

Instead of explicitly enumerating the worlds of a counter-example of bounded size (as in the reduction stated in the above theorem), we can also do this enumeration implicitly with a quantified boolean formula (QBF). We can show that this QBF is satisfiable iff there exists a counter-model for our original formula:

Theorem 2.1.3. Let $\mathcal{S}$ be as in Theorem **??** and $\mathcal{F}_{\to} \subseteq \mathcal{S}$ denote the subset of formulas of the form $(A \to B) \to C$. There is an effective procedure that produces a QBF φ^Q of size $\mathcal{O}(|\mathcal{S}| \cdot |\mathcal{F}_{\to}| + |\mathcal{F}_{\to}|^3)$ with $2 \cdot |\mathcal{F}_{\to}| - 1$ quantifier alternations such that φ is intuitionistically valid if and only if φ^Q is not satisfiable. For fixed $N \in \mathbb{N}$, deciding intuitionistic validity for formulas $\bigwedge \mathcal{S} \to P$ where $\mathcal{S}$ is as above and $|\mathcal{F}_{\to}| = N$ is therefore in Σ_{2N-1}.

We then move to the first-order case. Here the construction is a bit more involved, but the underlying approach is the same. Again, we first perform a Tseytin-like transformation. Here the non-classical content is encapsulated by formulas of type $(A \to B) \to C$ and $\forall x A \to B$. The details of this translation are given in Section 4.1.2. We then proceed similarly to the propositional case. Note however that one difficulty arises from the fact that our (classical) domain will now contain on the one hand worlds in the Kripke frame, but on the other hand also proper domain elements. We resolve this apparent conflict by introducing a special binary predicate E, inspired by [20], encoding which domain elements exists at which world. The main transformation then again proceeds in three steps: Step 1) and 2) are analogous to the propositional case. The real difference arises in step 3): While we can also establish the existence of a canonic counter-model whose frame only depends on formulas of type $(A \to B) \to C$ and $\forall x A \to B$, the size of this counter-model is countably infinite in general. This is not surprising, since certain intuitionistically invalid formulas like the double negation shift $\forall x \neg\neg A(x) \to \neg\neg\forall x A(x)$ don't have finite counter-models. Therefore, we are not able to eliminate the $\forall$-quantifiers associated with the Kripke semantics. However, since our translation targets first-order logic, we believe that the introduction of these quantifiers is acceptable.

Summarizing, we obtain the following result:

Theorem 2.1.4. There exists a linear-time procedure that gives for every first-order formula φ a formula $\varphi^{\#}$ such that φ is intuitionistically valid if and only if $\varphi^{\#}$ is classically valid. The size of $\varphi^{\#}$ is in linear in the size of φ, however, for each n-ary relation symbol in φ there is a corresponding $n+1$-ary relation symbol in $\varphi^{\#}$, and φ contains a new binary predicate E as well as a number of new function symbols. Furthermore, there is an effective translation between intuitionistic counter-models of φ and classical counter-models of $\varphi^{\#}$.

A more detailed account of Theorem 2.1.4 is given in Theorem 4.3.1.

2.2 Implementation

In this section, we describe our system that implements the translation from the previous section. The rust source code can be found online [2]. We then elaborate our benchmarking process, which utilizes the ILTP problem set [4] and the Vampire theorem prover [22].

Our implementation performs similar to existing dedicated provers for intuitionistic logic, but falls short of the state-of-the-art. Since there is still a lot of possible optimizations left for our translation, this is a hopeful first sign.

2.3 Translation of counter-models and proofs

In the final section, we give an explicit mechanism how to transform the proofs and counter-models for translated formula $\varphi^{\#}$ into intuitionistic proofs/counter-models of the

	AGT	ALG	COM	CSR	GEO	GRA	GRP	HAL
Proven	5	170	0	0	58	3	4	2
Disproven	0	0	0	0	0	0	0	
Timeout	47	29	3	29	107	74	1	16

	KRS	LCL	MGT	MSC	NLP	NUM	PLA	PUZ
Proven	0	0	30	1	9	1	9	2
Disproven	0	0	0	0	12	0	0	0
Timeout	5	9	128	3	69	1	237	82

	SET	SWC	SWV	SYN	TOP	GEJ	GPJ	SYJ
Proven	0	5	27	1	64	65	54	1
Disproven	6	1	297	423	154	108	298	2

Figure 2.1: Performance on problem sections of the ILTP set.

original formula φ. The existence of these transformations is clear from the methodology of chapter 4, however, they are hidden in the proofs and here we make them explicit. The two main results are as follows:

Theorem 2.3.1. There exists an effective procedure that converts any classical counter-model for a translated formula $\varphi^{\#}$ into an intuitionistic counter-model for the original formula φ.

Theorem 2.3.2. There exists an effective procedure that converts any classical proof of a translated formula $\varphi^{\#}$ into a intuitionistic proof of the original formula φ.

Given the arguments of Chapter 4 the translation of counter-models is more or less just a compilation of proofs. On the other hand, the explicit translation of proofs does require some work, since to that point all arguments were semantic and this requires a syntactic viewpoint.

Chapter 3

Preliminaries

First, let us recapitulate the most important definitions and some facts about them. This will serve to fix our notation as well.

3.1 Syntax

3.1.1 Propositional logic

Let us fix some countably infinite set of propositional variables $A, B, C \dots$.

Definition 3.1.1. A *formula* and *subformula* is defined inductively via the following rules:

- every propositional variable as well as a special symbol $\bot$ is a formula, such a formula is called *atomic* and $A, B, C, \dots$ and $\bot$ are called *atoms*.
- if φ and ψ are formulas then $\varphi \circ \psi$ is a formula for $\circ \in \{\wedge, \vee, \rightarrow\}$.
- every formula is a subformula of itself.
- for a formula $\chi = \varphi \circ \psi$ every subformula of φ and ψ is also a subformula of χ.

We write $\neg\varphi$ for $\varphi \rightarrow \bot$.

3.1.2 Predicate logic

The analogon for predicate logic is much more involved as we need to define terms.

Definition 3.1.2. A *signature* consists of

- a finite set S_f of function symbols $f_1, \dots, f_n$
- a finite set S_R of relation symbols $R_1, \dots, R_m$
- a function ar: $S_f \cup S_R \rightarrow \mathbb{N}$ assigning to each symbol its arity (possibly 0).

Usually the signature will be left implicit. If explicitly stated we denote it by

$$\{f_1/\text{ar}(f_1), \dots, f_n/\text{ar}(f_n), R_1/\text{ar}(R_1). \dots, R_m/\text{ar}(R_m)\}.$$

Fix a countably infinite collection of free variables $a, b, c \dots$ and bound variables $x, y, z \dots$. Note that the following definitions could be simplified by not distinguishing between bound and free variables. However, this has some other drawbacks and in particular when defining a sequent calculus for first-order logic having disjoint sets of bound and free variables is convenient.

Definition 3.1.3. A *semiterm* is defined inductively via the following rules

- each variable is a term
- if $t_1, \dots, t_n$ are terms and f a n-ary function symbol then $f(t_1, \dots, t_n)$ is a term.

A *term* is a semiterm in which each occurring variable is a free variable. Every term is a semiterm. Analogously to the definition of subformulas we can define *subsemiterms* and *subterms*.

Definition 3.1.4. A *Substitution* is a function σ from free variables to semiterms. For a semiterm t we define $t\sigma$ as follows:

- $x\sigma = x$ if x is bound.
- $a\sigma = \sigma(a)$ if a is free.
- $f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$.

Definition 3.1.5. A *formula* is defined inductively via the following rules

- $\bot$ is a formula.
- if $t_1, \dots, t_n$ are terms and R and an n-ary relation symbol then $R(t_1, \dots, t_n)$ is a formula.
- if s, t are terms then $s = t$ is a formula.

Formulas formed by the above rules are called *atomic.*

- If φ and ψ are formulas then $\varphi \circ \psi$ is a formula for $\circ \in \{\land, \lor, \to\}$.
- If φ is a formula and a is a free variable occurring in φ and x is a bound variable not occurring in φ then $Qx(\varphi[x/a])$ is a formula for $Q \in \{\exists, \forall\}$ where the *substitution* $\varphi\sigma$ is defined inductively via
  - $\bot\sigma = \bot$

  - $R(t_1, \ldots, t_n)\sigma = R(t_1\sigma, \ldots, t_n\sigma)$
  - $(s = t)\sigma = (s\sigma = t\sigma)$
  - $(\varphi \circ \psi)\sigma = \varphi\sigma \circ \psi\sigma$ for $\circ \in \{\wedge, \vee, \rightarrow\}$.
  - $(Qy\varphi)\sigma = Qy(\varphi\sigma)$ for $Q \in \{\exists, \forall\}$.

- A formula containing no free variables is a *sentence*.

3.2 Semantics

Classical and intuitionistic semantics give (different) meaning to formulas.

3.2.1 Propositional Logic

Classical Semantics

Definition 3.2.1. A *valuation* v is a function that assigns to each propositional variable A a *truth value* $v(A) \in \{0, 1\}$. We define a model relation $\models$ between valuations v and formulas:

- $v \not\models \bot$
- $v \models A$ iff $v(A) = 1$ for each propositional variable A.
- $v \models \varphi \wedge \psi$ iff $v \models \varphi$ and $v \models \psi$.
- $v \models \varphi \vee \psi$ iff $v \models \varphi$ or $v \models \psi$.
- $v \models \varphi \rightarrow \psi$ iff $v \not\models \varphi$ or $v \models \psi$.

v is a *model* for φ if $v \models \varphi$. φ is *satisfiable* if there exists a valuation which is a model. It is *valid* if every valuation is a model, and we write $\models \varphi$. We denote the set of valid formulas with CPC. We can furthermore define an entailment relation between sets of clauses, that is $\mathcal{S} \models_C \mathcal{T}$ if for every valuation v with $v \models \bigwedge S$ we have $v \models \bigvee \mathcal{T}$.

One notable property of classical logic is that a formula φ is valid iff its negation, i.e. $\neg\varphi$ is not satisfiable. The same does not hold true for intuitionistic logic, as we shall see.

Intuitionistic Semantics

Definition 3.2.2. A *Kripke structure* $\mathcal{K} = (W, (v_w)_{w \in W})$ consists of a partially ordered set W of nodes an a family of valuations $(v_w)_{w \in W}$ such that for $u \leq w$ we have $v_u(A) \leq v_u(A)$ for every propositional variable A, this is called the *persistency condition.* In this case, we define a model relation between worlds u and formulas. It is slightly different in the case of implications, i.e. have

- $u \not\models \bot$
- $u \models A$ iff $v_u(A) = 1$ for each propositional variable A.
- $u \models \varphi \wedge \psi$ iff $v \models \varphi$ and $v \models \psi$.
- $u \models \varphi \vee \psi$ iff $v \models \varphi$ or $v \models \psi$.
- $u \models \varphi \to \psi$ if for all $w \geq u$ we have $w \not\models \varphi$ or $w \models \psi$.

We say that φ is satisfied at a node u if $u \models \varphi$. If a formula is satisfied at every node, then K is a *model* for φ, and we write $K \models \varphi$. φ is satisfiable if there exists a Kripke structure, which is a model. It is *valid* if every Kripke structure is a model and we write $\models \varphi$. We denote the set of valid formulas with IPC. As before, we define an entailment relation $\models_I$ between sets of formulas.

There are many classical tautologies which are not intuitionistically valid

Example 3.2.3. Consider the famous law of excluded middle $A \vee \neg A$. As a counter model consider a Kripke structure with two nodes $u \leq w$ and $v_u(A) = 0, v_w(A) = 1$. Then $u \not\models A$ and $u \not\models \neg A$, i.e. $u \not\models A \vee \neg A$.

The following is a famous and important result due to Glivenko [17]

Theorem 3.2.4 (Glivenko's theorem)**.** A propositional formula φ is classically valid iff $\neg\neg\varphi$ is intuitionistically valid.

Note however that this difference between classical and intuitionistic logic is not reflected in satisfiability, only in validity:

Lemma 3.2.5. A propositional formula φ is intuitionistically satisfiable if and only if it is classically satisfiable.

Proof. Every classical model can be viewed as a Kripke model with a single node. Therefore, classical satisfiability trivially implies intuitionistic satisfiability. On the other hand, in every Kripke structure that is a model every node is a classical model so intuitionistic satisfiability implies intuitionistic satisfiability. □

In classical logic, a proven strategy for showing that a formula valid is to consider its negation and show that it is not satisfiable. Due to the previous lemma, this is insufficient for intuitionistic logic. There are formulas like $A \vee \neg A$ which are not an intuitionistic tautology, while their negation is not satisfiable either. This serves to motivate the consideration of equivalidity in the following chapters. In the context of classical logic it can be fully explained via equisatisfiability and is therefore neglected, but for intuitionistic logic it is more fundamental.

Finally, let us note the complexities of these various decision problems, which play a fundamental role in all of theoretical computer science.

Theorem 3.2.6. Satisfiability for classical and intuitionistic propositional logic is NP-complete.

For classical logic this is the famous Cook-Levin theorem first published in Cook's seminal paper [12]. Of course, we have seen that the intuitionistic case is the same decision problem.

Corollary 3.2.7. Validity for classical logic is co-NP-complete

This is clear, since validity is just the complementary problem to satisfiability of the negated formula. However, in intuitionistic logic unfortunately validity is really harder than satisfiability as first shown by Statman [25].

Theorem 3.2.8. Validity for intuitionistic logic in PSPACE-complete.

One thing to note is that in contrast to predicate logic, all these problems are decidable.

3.2.2 Predicate Logic

We now give account of the more involved semantics of first-order logic

Classical Semantics

Definition 3.2.9. Let Σ be a signature. A Σ-structure $\mathcal{M}$ consists of a non-empty set M, the domain of $\mathcal{M}$, and a function I that assigns

- to each n-ary function symbol f a n-ary function $f^I : M^n \to M$.
- to each n-ary relation symbol R a n-ary relation $R^I \subseteq M^n$.

A variable assignment v is a function that assigns to each free variable an element $m \in M$. For each free variable a and $m \in M$ we define

$$v[m/a](b) = \begin{cases} m & \text{if } b = a \\ v(b) & \text{otherwise} \end{cases}$$

Then terms are interpreted as follows:

- $a^{I,v} = v(a)$ for each free variable a.
- $f(t_1, \ldots, t_n)^{I,v} = f^I(t_1^{I,v}, \ldots, t_n^{I,v})$.

We define a model relation between pairs $\mathcal{M}, v$ and formulas φ as follows

- $\mathcal{M}, v \not\models \bot$.
- $\mathcal{M}, v \models R(t_1, \ldots, t_n)$ iff $(t_1^{I,v}, \ldots, t_n^{I,v}) \in R^I$.

- $\mathcal{M}, v \models s = t$ iff $s^{I,v} = t^{I,v}$.
- $\mathcal{M}, v \models \varphi \land \psi$ iff $\mathcal{M}, v \models \varphi$ and $\mathcal{M}, v \models \psi$.
- $\mathcal{M}, v \models \varphi \lor \psi$ iff $\mathcal{M}, v \models \varphi$ or $\mathcal{M}, v \models \psi$.
- $\mathcal{M}, v \models \varphi \to \psi$ iff $\mathcal{M}, v \not\models \varphi$ or $\mathcal{M}, v \models \psi$.
- $\mathcal{M}, v \models \exists x \varphi$ iff there exists $m \in M$ such that $\mathcal{M}, v[m/a] \models \varphi[a/x]$ where a is a free variable that does not occur in φ.
- $\mathcal{M}, v \models \forall x \varphi$ iff for all $m \in M$ we have $\mathcal{M}, v[m/a] \models \varphi[a/x]$ where a is a free variable that does not occur in φ.

φ is satisfiable if for some $\mathcal{M}, v$ we have $\mathcal{M}, v \models \varphi$. It is valid if for all $\mathcal{M}, v$ we have $\mathcal{M}, v \models \varphi$. We write $\mathcal{M} \models \varphi$ if for every v we have $\mathcal{M}, v \models \varphi$. We denote the set of valid formulas with CQC. As before define an entailment relation $\models_C$ between sets of formulas.

Intuitionistic Semantics

Definition 3.2.10. A Σ-*Kripke structure* $\mathcal{K}$ is a partially ordered set W and a family of Σ-structures $(\mathcal{M}_w)_{w \in W}$ such that for $u \leq w$

- $M_u \subseteq M_w$.
- $f^{I_w}|_{M_u} = f^{I_u}$.[1]
- $R^{I_w}|_{M_u} = R^{I_u}$.

Then we define a model relation between worlds $u \in W$ and variable assignments v (targeting M_u) and formulas φ as follows:

- $u, v \not\models \bot$
- $u, v \models R(t_1, \ldots, t_n)$ iff $(t_1^{I_u,v}, \ldots, t_n^{I_u,v}) \in R^{I_u}$.
- $u, v \models s = t$ iff $s^{I_u,v} = t^{I_u,v}$.
- $u, v \models \varphi \land \psi$ iff $u, v \models \varphi$ and $u, v \models \psi$.
- $u, v \models \varphi \lor \psi$ iff $u, v \models \varphi$ or $u, v \models \psi$.
- $u, v \models \varphi \to \psi$ iff for every $w \geq u$ we have $w, v \not\models \varphi$ or $w, v \models \psi$.
- $u, v \models \exists x \varphi$ iff there exists $m \in M_u$ such that $u, v[m/a] \models \varphi[a/x]$ where a is a free variable that does not occur in φ.

[1]Here $f|_M$ denotes the restriction of f to M.

- $u, v \models \forall x \varphi$ iff for every $w \geq u$ and $m \in M_w$ we have $w, v[m/a] \models \varphi[a/x]$ where a is a free variable that does not occur in φ.

We say that $\mathcal{K}$ satisfies φ if $u, v \models \varphi$ holds for every world u and variable assignment v and write $\mathcal{K} \models \varphi$. As always, a formula is satisfiable if it is satisfied by some Kripke structure, and it is valid if it is satisfied in every Kripke structure. We denote the set of valid formulas with IQC. As before, define an entailment relation $\models_I$ between sets of formulas.

In addition to the propositional tautologies, there are now sentences involving quantifiers, which are classically valid but not intuitionistically. Consider for instance the classical tautology φ

$$\neg \forall x A(x) \rightarrow \exists x \neg A(x)$$

We exhibit an intuitionistic counter model. Consider the Kripke structure with two nodes $u \leq w$ with $M_u = \{0\}, M_w = \{0, 1\}$ and $A^{I_u} = A^{I_w} = \{0\}$. Then $u, v \not\models \varphi$: First note that $w, v \not\models \forall x A(x)$ and $u, v \not\models \forall x A(x)$ for any valuation v since $w, v[1/a] \not\models A(a)$, i.e. $u, v \models \neg \forall x A(x)$ and also $w, v \models \neg \forall x A(x)$. Furthermore, $u, v[0/a] \not\models \neg A(a)$ and $w, v[0/a] \not\models \neg A(a)$ so $u, v \not\models \exists x \neg A(x)$ for any valuation v (note however that this is not true at w!). Therefore, $u, v \not\models \varphi$.

Intuitively, the problem is that the existential quantifier at u does not "see" the elements at w, i.e. it must "choose" between the elements at u. As in the propositional case, there are well-known negative translations of classical logic into intuitionistic logic going back to Gödel [18] and Gentzen [15].

Definition 3.2.11. For a predicate formula φ define inductively φ^N as follows:

- $\varphi^N = \varphi$ if φ is atomic
- $(\varphi \wedge \psi)^N = \varphi^N \wedge \psi^N$
- $(\varphi \vee \psi)^N = \neg(\neg \varphi^N \wedge \neg \varphi^N)$
- $(\varphi \rightarrow \psi)^N = \varphi^N \rightarrow \psi^N$
- $(\forall x \varphi)^N = \forall x \varphi^N$
- $(\exists x \varphi)^N = \neg \forall x \neg \varphi^N$

Theorem 3.2.12. A Σ-formula φ is classically valid if and only if φ^N is intuitionistically valid.

But again this is not reflected in satisfiability, i.e. with a completely analogous argument to 3.2.5 we obtain

Lemma 3.2.13. Any Σ-formula φ is classically satisfiable if and only if it is intuitionistically satisfiable.

3.3 Skolemization and Herbrandization

An important step in the embedding will be the elimination of quantifiers via Herbrandization. In this process, we introduce fresh variables and add additional function symbols to the signature. A fresh variable is any variable that does not occur in any of the considered formulas. Whenever we add a function symbol, we implicitly extend the signature by some not previously contained symbol.

Definition 3.3.1. For formulas φ we define the Skolemization φ^S_Z and Herbrandization φ^H_Z with respect to Z by simultaneous induction as follows:

- $A^S_Z = A^H_Z = A$ for each atomic A.
- $(\varphi \circ \psi)^X_Z = \varphi^X_Z \circ \psi^X_Z$ for $\circ \in \{\wedge, \vee\}$, $X \in \{S, H\}$.
- $(\varphi \to \psi)^S_Z = \varphi^H_Z \to \psi^S_Z$
  $(\varphi \to \psi)^H_Z = \varphi^S_Z \to \psi^H_Z$.
- $(\forall x \varphi)^S_Z = \forall x(\varphi[a/x]^S_{Z\cup\{a\}}[x/a])$ where a is a new free variable
  $(\forall x \varphi)^H_Z = \varphi[s(z_1, \ldots, z_n)/x]^H_Z$ where s is a new function, $\{z_1 \ldots z_n\} = Z$.
- $(\exists x \varphi)^S_Z = \varphi[s(z_1, \ldots, z_n)/x]^S_Z$ where s is a new function, $\{z_1 \ldots z_n\} = Z$
  $(\exists x \varphi)^H_Z = \exists x(\varphi[a/x]^H_{Z\cup\{a\}}[x/a])$ where a is a new free variable.

Let $\varphi^S = (\exists x_1 \ldots \exists x_n \varphi[x_1/a_1 \ldots x_n/a_n])^S_\emptyset$ and $\varphi^H = (\forall x_1 \ldots \forall x_n \varphi[x_1/a_1 \ldots x_n/a_n])^H_\emptyset$ where $a_1, \ldots, a_n$ are the free variables occurring in φ.

Theorem 3.3.2. For every formula φ

- φ and φ^S are classically equisatisfiable.
- φ and φ^H are classically equivalid.

which follows from the following two Lemmata:

Lemma 3.3.3. Let χ be a formula and let $\{z_1 \ldots z_n\} = Z$ contain all free variables in χ, then for every structure $\mathcal{M} = (M, I)$ there exists $\mathcal{M}_\chi = (M, I_\chi)$ such that $p^{I_\chi} = p^I$ for all function and relation symbols p occurring in χ and for all variable assignments v we have

- if $M, v \models \chi$ then $\mathcal{M}_\chi, v \models \chi^S_T$.
- if $\mathcal{M}, v \not\models \chi$ then $\mathcal{M}_\chi, v \not\models \chi^H_T$.

Proof. We proceed by simultaneous induction on the formula height.

For atomic χ the claims are clear.

Otherwise, there are 3 cases:

1. $\chi = \varphi \circ \psi$ for $\circ \in \{\wedge, \vee, \to\}$. By induction hypothesis there exist $\mathcal{M}_{\varphi,v} = (M, I_{\varphi,v})$ and $\mathcal{M}_{\psi,v} = (M, I_{\psi,v})$ for φ, ψ as stated in the lemma. Then we can define $I_{\chi,v}$ as follows:

For every symbol p that occurs in φ^S_Z or φ^H_Z but not ψ^S_Z, ψ^H_Z, we set $p^{I_{\chi,v}} = p^{I_{\varphi,v}}$. For every symbol p that occurs in ψ^S_Z or ψ^H_Z but not φ^S_Z, φ^H_Z, we set $p^{I_{\chi,v}} = p^{I_{\psi,v}}$. Otherwise, p already occurs in φ so have $p^{I_{\chi,v}} = p^I$.

Let $\circ = \to$. Suppose $\mathcal{M}, v \models \chi$. Then $\mathcal{M}, v \not\models \varphi$ or $\mathcal{M} \models \psi$ and therefore $\mathcal{M}_\chi \not\models \varphi^S_Z$ or $\mathcal{M}_\chi \models \psi^H_Z$, i.e. $\mathcal{M}_\chi \models \chi^S_Z$. On the other hand, suppose $\mathcal{M}, v \not\models \chi$. Then $\mathcal{M}, v \models \varphi$ and $\mathcal{M}, v \not\models \psi$ and therefore $\mathcal{M}_\chi, v \models \varphi^S_Z$ and $\mathcal{M}_\chi, v \not\models \psi^S_Z$, i.e. $\mathcal{M}_\chi \not\models \chi^H_Z$. Analogous arguments work for $\circ \in \{\wedge, \vee\}$.

2. $\chi = \forall x\varphi$. Choose $s^{I_\chi} : M^n \to M$ such that $s^{I_\chi}(x_1, \dots, x_n) = m$ if there exists $m \in M$ such that $\mathcal{M}, v[x_1/z_1 \dots x_n/z_n, m/a] \not\models \varphi[a/x]$ for all v and arbitrary otherwise. Since Z contains all free variables occurring in χ this is a well-defined function.

By induction hypothesis there exists a structure $\mathcal{M}_{\varphi[a/x]}$ as is the Lemma. Let $p^{I_\chi} = p^{I_{\varphi[a/x]}}$ for all symbols occurring in χ and

$$p^{I_\chi}(x_1 \dots x_{i-1}, x_{i+1} \dots x_m) = p^{I_{\varphi[a/x]}}(x_1 \dots x_{i-1}, s^{I_\chi}(x_1 \dots x_n), x_{i+1} \dots x_m)$$

for all symbols p occurring in $\varphi[a/x]^S_{Z \cup a}$ or $\varphi[a/x]^H_{Z \cup a}$ but not in χ where x_i is the argument corresponding to the free variable a.

Suppose $\mathcal{M}, v \models \chi$. Then, for all $m \in M$ we have that $\mathcal{M}, v[m/a] \models \varphi[a/x]$ and therefore $\mathcal{M}_\chi, v[m/a] \models \varphi[a/x]^S_{Z \cup \{a\}}$ and thus $\mathcal{M}_\chi, v \models \forall x\varphi([a/x]^S_{Z \cup \{a\}}[x/a])$, i.e. $\mathcal{M}_\chi, v \models \chi^S_Z$. On the other hand, suppose $\mathcal{M}, v \not\models \chi$. Then there exists $m \in M$ such that $\mathcal{M}, v[m/a] \not\models \varphi[a/x]$ and therefore $\mathcal{M}_\chi, v[m/a] \not\models \varphi[a/x]^H_{Z \cup \{a\}}$ and so by definition $\mathcal{M}_\chi, v \not\models \varphi[s(z_1, \dots z_n)/x]^H_Z$, i.e. $\mathcal{M}_\chi, v \not\models (\forall x\varphi)^H_Z$.

3. $\chi = \exists x\varphi$. The argument runs dually to 2. Choose $s^{I_\chi} : M^n \to M$ such that $s^{I_\chi}(x_1, \dots, x_n) = m$ if there exists $m \in M$ such that for all v we have $\mathcal{M}, v[x_1/z_1 \dots x_n/z_n, m/a] \models \varphi[a/x]$ and arbitrary otherwise. Since Z contains all free variables occurring in χ this is a well-defined function.

By induction hypothesis there exists a structure $\mathcal{M}_{\varphi[a/x]}$ as is the Lemma. Let $p^{I_\chi} = p^{I_{\varphi[a/x]}}$ for all symbols occurring in χ and

$$p^{I_\chi}(x_1 \dots x_{i-1}, x_{i+1} \dots x_m) = p^{I_{\varphi[a/x]}}(x_1 \dots x_{i-1}, s^{I_\chi}(x_1 \dots x_n), x_{i+1} \dots x_m)$$

for all symbols p occurring in $\varphi[a/x]^S_{Z \cup a}$ or $\varphi[a/x]^H_{Z \cup a}$ but not in χ where x_i is the argument corresponding to the free variable a.

Then as in 2 from $\mathcal{M}, v \models \chi$ follows $\mathcal{M}_\chi, v \models \chi^S_Z$ and from $\mathcal{M}, v \not\models \chi$ follows $\mathcal{M}_\chi, v \not\models \chi^H_Z$. □

Lemma 3.3.4. For every structure $\mathcal{M}$ and $\{z_1 \dots z_n\} = Z$ that contains all free variables in χ and variable assignment v

- if $\mathcal{M}, v \models \varphi^S_Z$ then $\mathcal{M}, v \models \varphi$
- if $\mathcal{M}, v \not\models \varphi^H_Z$ then $\mathcal{M}, v \not\models \varphi$.

Proof. Again, we proceed by simultaneous induction on the formula height.

For atoms the claims are clear. We distinguish 5 cases.

1. $\chi = \varphi \to \psi$. Suppose $\mathcal{M}, v \models \chi^S_Z$, i.e. $\mathcal{M}, v \not\models \varphi^H_Z$ or $\mathcal{M}, v \models \psi^S_Z$. By induction hypothesis, $\mathcal{M}, v \not\models \varphi$ or $\mathcal{M}, v \models \psi$, i.e. $\mathcal{M}, v \models \chi$. On the other hand, suppose $\mathcal{M}, v \not\models \chi^H_Z$, i.e. $\mathcal{M}, v \models \varphi^S_T$ and $\mathcal{M}, v \not\models \varphi^H_Z$. Again, by induction hypothesis $\mathcal{M}, v \models \varphi$ and $\mathcal{M}, v \not\models \varphi$, so $\mathcal{M}, v \not\models \chi$.

2. + 3. Conjunctions and Disjunctions are dealt with analogously.

4. $\chi = \forall x \varphi$. Suppose $\mathcal{M}, v \models \chi^S_Z$, i.e. for all $m \in M$ we have $\mathcal{M}, v[m/a] \models \chi[a/x]^S_{Z\cup\{a\}}$ and, by induction hypothesis, $\mathcal{M}, v[m/a] \models \chi[a/x]$. Then it follows, that $\mathcal{M}, v \models \varphi$. On the other hand, suppose $\mathcal{M}, v \not\models \chi^H_Z$, i.e. there exists $m \in M$ such that $\mathcal{M}, v[m/a] \not\models \chi[a/x]^H_{Z\cup\{a\}}$, then by induction hypothesis $\mathcal{M}, v[m/a] \not\models \chi[a/x]$, i.e. $\mathcal{M}, v \not\models \forall x \chi$.

5. $\chi = \exists x \varphi$. The argument runs dually to 4. □

3.4 Proof theory

Logical truth can also be approached via syntax rather than semantics. This gives rise to proof theory, which will be especially important in the final chapter when we establish proof translations.

Definition 3.4.1. Pairs of multisets of formulas $A = \{A_1, \dots, A_n\}, B = \{B_1, \dots, B_m\}$ form a *Sequent* $A \Rightarrow B$. Usually, we will write $A_1, \dots, A_n \Rightarrow B_1, \dots, B_m$, note however that A and B are **not** to be interpreted as sequences but as multisets.

We define sequent calculi as in [27, p.77]. Have following inferences

$$\frac{}{P, \Gamma \Rightarrow \Delta, P}\ \text{Ax } (P \text{ atomic}) \qquad \frac{}{\bot, \Gamma \Rightarrow \Delta}\ \text{L}\bot$$

$$\frac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\ \text{L}\wedge \qquad \frac{\Gamma \Rightarrow \Delta, A \qquad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}\ \text{R}\wedge$$

$$\frac{A, \Gamma \Rightarrow \Delta \qquad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}\ \text{L}\vee \qquad \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B}\ \text{R}\vee$$

$$\frac{\Gamma \Rightarrow \Delta, A \qquad B, \Gamma \Rightarrow \Delta}{A \to B, \Gamma \Rightarrow \Delta}\ \text{L}\to \qquad \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \to B}\ \text{R}\to$$

$$\frac{A[t/x], \Gamma \Rightarrow \Delta}{\forall x A, \Gamma \Rightarrow \Delta}\ \text{L}\forall \qquad \frac{\Gamma \Rightarrow \Delta, A[a/x]}{\Gamma \Rightarrow \Delta, \forall x A}\ \text{R}\forall$$

$$\frac{A[a/x], \Gamma \Rightarrow \Delta}{\exists x A, \Gamma \Rightarrow \Delta}\ \text{L}\exists \qquad \frac{\Gamma \Rightarrow \Delta, A[t/x]}{\Gamma \Rightarrow \Delta, \exists x A}\ \text{R}\exists$$

where in R∀ and L∃ a is a free variable not occurring in the sequent otherwise.

The above inferences make up the calculus for classical logic called **Gc**. It corresponds to **G3c**$^=$ from [27]. Similarly, we can define a calculus **Gi** for intuitionistic logic corresponding to **m-G3i**$^=$:

$$\dfrac{}{P, \Gamma \Rightarrow \Delta, P}\text{ Ax } (P \text{ atomic}) \qquad \dfrac{}{\bot, \Gamma \Rightarrow \Delta}\text{ L}\bot$$

$$\dfrac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\text{ L}\wedge \qquad \dfrac{\Gamma \Rightarrow \Delta, A \qquad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}\text{ R}\wedge$$

$$\dfrac{A, \Gamma \Rightarrow \Delta \qquad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}\text{ L}\vee \qquad \dfrac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B}\text{ R}\vee$$

$$\dfrac{A \to B, \Gamma \Rightarrow \Delta, A \qquad B, \Gamma \Rightarrow \Delta}{A \to B, \Gamma \Rightarrow \Delta}\text{ L}\to \qquad \dfrac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow \Delta, A \to B}\text{ R}\to$$

$$\dfrac{A[t/x], \Gamma \Rightarrow \Delta}{\forall x A, \Gamma \Rightarrow \Delta}\text{ L}\forall \qquad \dfrac{\Gamma \Rightarrow A[a/x]}{\Gamma \Rightarrow \forall x A}\text{ R}\forall$$

$$\dfrac{A[a/x], \Gamma \Rightarrow \Delta}{\exists x A, \Gamma \Rightarrow \Delta}\text{ L}\exists \qquad \dfrac{\Gamma \Rightarrow \Delta, A[t/x]}{\Gamma \Rightarrow \Delta, \exists x A}\text{ R}\exists$$

The differences are found in L→, R→ and R∀. Note how in particular the rules for R→ and R∀ allow only a single formula in the succedent.

Definition 3.4.2. A *Derivation* is a rooted finite labelled tree, in which each leaf is labelled with one of the two axiom rules and each other label is derived from the labels of the successor nodes in accordance to the rules above. All formulas not in Γ or Δ are the *active* formulas of the rule. The label of the root is called *head*. A formula that is the head of a derivation is said to be *derivable*.

The following is one of the fundamental results of first-order Logic:

Theorem 3.4.3 (Completeness theorem)**.** A sequent $\Phi \Rightarrow \varphi$ is derivable if and only if $\Phi \models \varphi$.

A further important property of the above calculi is the subformula property. While it is typically used to prove soundness of propositional logic, it is also a useful tool for proving many other results.

Theorem 3.4.4 (Subformula Property)**.** Let φ be a formula that occurs in a derivation of $\Delta \Rightarrow \Gamma$. Then φ is a subformula of some formula in $\Delta \cup \Gamma$.

A direct consequence of this result is the soundness of propositional logic:

Theorem 3.4.5 (Soundness of propositional logic)**.**
The empty sequent is not derivable in **Gc** and **Gi**.

Example 3.4.6. The following is a **Gc** derivation of the law of excluded middle.

$$\dfrac{\dfrac{\dfrac{}{A \Rightarrow \bot, A}\,\text{Ax}}{\Rightarrow A, A \to \bot}\,\text{R}{\to}}{\Rightarrow A \lor (A \to \bot)}\,\text{R}{\lor}$$

Note how this derivation is invalid in **Gi** since R$\to$ is not applied correctly. However it is possible to prove its double negation in **Gi**:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(A \lor (A \to \bot)) \to \bot, A \Rightarrow A, (A \to \bot), \bot}\,\text{Ax}}{(A \lor (A \to \bot)) \to \bot, A \Rightarrow A \lor (A \to \bot), \bot}\,\text{R}{\lor} \qquad \dfrac{}{\bot \Rightarrow}\,\text{L}\bot}{(A \lor (A \to \bot)) \to \bot, A \Rightarrow \bot}\,\text{L}{\to}}{(A \lor (A \to \bot)) \to \bot \Rightarrow A, A \to \bot}\,\text{R}{\to}}{(A \lor (A \to \bot)) \to \bot \Rightarrow A \lor (A \to \bot)}\,\text{R}{\lor} \qquad \dfrac{}{\bot \Rightarrow}\,\text{L}\bot}{(A \lor (A \to \bot)) \to \bot \Rightarrow}\,\text{L}{\to}$$

Chapter 4

Embedding intuitionistic into classical logic

The goal of this section is to give a complete embedding of intuitionistic into classical logic. We shall start with the simpler case of propositional formulas. The process will be divided into three steps. The first is a first-order encoding of intuitionistic semantics, which gives an obvious but unsatisfactory embedding of propositional intuitionistic logic into classical logic. We then discuss how some of the quantifiers can be eliminated via Herbrandization. As a third part, we give a certain normal form that then finally allows us to realize that all quantifiers can be indeed eliminated from the translated formula, and we end up with a propositional formula. As a quick aside, we also give an embedding form IPC to QBF.

For predicate formulas, the process is a bit more involved in that we also need to respect the constructive content of quantifiers. For this, we use a special existence predicate E inspired by the work in [20]. This allows us to utilize a quite similar process as in the first section, even though it is a bit more involved.

4.1 Propositional logic

As mentioned before, we proceed by first giving an embedding into classical predicate logic, transforming that embedding and finally arguing that all the quantifiers can be eliminated, by showing that if a counter-model exists than there exists a bounded one.

4.1.1 IPC to CQC

The most obvious approach to embedding intuitionistic into classical logic is to express intuitionistic semantics, i.e. Kripke frames, as a first-order theory. For every propositional variable A consider a unary predicate A of the same name where $A(u)$ expresses that A is true at some world u. We can then naively encode formulas as follows:

Definition 4.1.1. Let φ be a propositional formula. Define φ^u inductively as follows:

- $A^u = A(u)$ for every propositional variable A.
- $\bot^u = \bot$.
- $(\varphi \circ \psi)^u = \varphi^u \circ \psi^u$ for $\circ \in \{\wedge, \vee\}$.
- $(\varphi \to \psi)^u = \forall w(u \preceq w \to \varphi^w \to \psi^w)$ where w is some new variable.

Let $K(\varphi)$ encode the theory of Kripke structures, i.e.

$$K(\varphi) = \text{PartialOrder}(\preceq) \wedge \forall u \forall w(u \preceq w \to \text{Persistent}(u, w))$$

with e.g.

$$\begin{aligned}\text{PartialOrder}(\preceq) &= \forall u(u \preceq u) \wedge \forall u \forall w(u \preceq w \to w \preceq u \to u = w) \wedge \\ &\quad \forall u \forall v \forall w(u \preceq v \to v \preceq w \to u \preceq w) \\ \text{Persistent}(u, w) &= \bigwedge \{A(u) \to A(w) \mid \ A \text{ is a propositional variable that occurs in } \varphi\}\end{aligned}$$

Then define

$$\varphi^C = K(\varphi) \to \varphi^u$$

where u is some free variable.

As all we have done is modelling Kripke frames as a first-order theory, we directly obtain:

Lemma 4.1.2. φ is intuitionistically valid if and only if φ^C is classically valid.

Example 4.1.3. Consider the law of excluded middle

$$\varphi = (A \to \bot) \vee A,$$

which is not intuitionistically valid. Then

$$\begin{aligned}\varphi^C &= K(\varphi) \to ((A \to \bot) \vee A)^u \\ &= K(\varphi) \to (A \to \bot)^u \vee A^u \\ &= K(\varphi) \to (\forall u'(u \preceq u' \to A^{u'} \to \bot^{u'}) \vee A(u)) \\ &= K(\varphi) \to (\forall u'(u \preceq u' \to A(u') \to \bot) \vee A(u))\end{aligned}$$

and indeed φ^C admits a counter model with domain $\{u, u'\}$, where $u \preceq u'$, $A(u) = 0, A(u') = 1$.

Note how this counter model corresponds exactly to the intuitionistic counter model presented in example 3.2.3. We can make this correspondence explicit.

Definition 4.1.4. Let $\mathcal{K} = (W, (v_w)_{v \in W})$ be a propositional Kripke structure. Define $\mathcal{M}(\mathcal{K}) = (M, I)$ as follows:

- As a domain take $M = W$.
- Have $u \in A^I$ iff $u, v_u \models A$.
- Interpret $\preceq$ as the partial order on W.

By definition, we then have

Lemma 4.1.5. $\mathcal{K} \models \varphi$ iff $\mathcal{M}(\mathcal{K}) \models \varphi^C$.

Similarly, for every Σ-structure $\mathcal{M}$ with $\mathcal{M} \models K(\varphi)$ we could construct a Kripke Frame $\mathcal{K}(\mathcal{M})$ such that $\mathcal{M} \models \varphi^C$ iff $\mathcal{K}(\mathcal{M}) \models \varphi$.

4.1.2 Normal form transformation

Of course, transforming a propositional formula into a predicate formula is somewhat unsatisfying. As a first step towards obtaining a propositional formula, we want to apply Herbrandization. Now, trying to apply it to arbitrary formulas presents us with quite a leap in complexity, leading to nested expressions that will make arguments more complicated. We therefore propose a Tseytin-like transformation that produces a formula of a more manageable syntactic form. In fact, our translation is often presented as the first step in the Tseytin transformation, after which the individual formulas are then converted to CNF. Hence, the propositional case of our translation is well known, but we are not aware of any work that uses just this step as a stand-alone. A similar translation for intuitionistic propositional logic was also proposed in [25].

Definition 4.1.6. Let φ be some formula. For each subformula ψ of φ, consider some new n-ary relation symbol P_ψ, where n is the number of variables occurring in ψ that is not quantified within ψ. We denote those variables with $\vec{z}_\psi$ in some arbitrary but fixed order. We inductively define clause sets $\mathcal{S}^+(\varphi)$ and $\mathcal{S}^-(\varphi)$ as follows:

- For every atomic formula A:
  $\mathcal{S}^+(A) = \{P_A(\vec{z}_A) \to A\}$
  $\mathcal{S}^-(A) = \{A \to P_A(\vec{z}_A)\}$
- For every conjunction or disjunction $\varphi \circ \psi$ with $\circ \in \{\wedge, \vee\}$:
  $\mathcal{S}^+(\varphi \circ \psi) = \{P_{\varphi\circ\psi}(\vec{z}_{\varphi\circ\psi}) \to P_\varphi(\vec{z}_\varphi) \circ P_\psi(\vec{z}_\psi)\} \cup \mathcal{S}^+(\varphi) \cup \mathcal{S}^+(\psi)$
  $\mathcal{S}^-(\varphi \circ \psi) = \{P_\varphi(\vec{z}_\varphi) \circ P_\psi(\vec{z}_\psi) \to P_{\varphi\circ\psi}(\vec{z}_{\varphi\circ\psi})\} \cup \mathcal{S}^-(\varphi) \cup \mathcal{S}^-(\psi)$
- For every implication $\varphi \to \psi$:
  $\mathcal{S}^+(\varphi \to \psi) = \{(P_{\varphi\to\psi}(\vec{z}_{\varphi\to\psi}) \wedge P_\varphi(\vec{z}_\varphi)) \to P_\psi(\vec{z}_\psi)\} \cup \mathcal{S}^-(\varphi) \cup \mathcal{S}^+(\psi)$
  $\mathcal{S}^-(\varphi \to \psi) = \{(P_\varphi(\vec{z}_\varphi) \to P_\psi(\vec{z}_\psi)) \to P_{\varphi\to\psi}(\vec{z}_{\varphi\to\psi})\} \cup \mathcal{S}^+(\varphi) \cup \mathcal{S}^-(\psi)$
- For quantified formulas $Qx\varphi(x)$ with $Q \in \{\forall, \exists\}$:
  $\mathcal{S}^+(Qx\varphi(x)) = \{P_{Qx\varphi}(\vec{z}_{Qx\varphi}) \to QxP_\varphi(\vec{z}_\varphi)\} \cup \{\forall x\psi \mid \psi \in \mathcal{S}^+(\varphi)\}$
  $\mathcal{S}^-(Qx\varphi(x)) = \{QxP_\varphi(\vec{z}_\varphi) \to P_{Qx\varphi}(\vec{z}_{Qx\varphi})\} \cup \{\forall x\psi \mid \psi \in \mathcal{S}^-(\varphi)\}$

The following lemma can be directly obtained by induction on the formula height:

Lemma 4.1.7. $\bigwedge S^+(\varphi) \wedge P_\varphi(\vec{z}_\varphi) \to \varphi$ and $\bigwedge S^-(\varphi) \wedge \varphi \to P_\varphi(\vec{z}_\varphi)$ are valid both classically and intuitionistically.

It is straight-forward to extend structures to the additional symbols such that the formulas $S^+(\varphi) \cup \mathcal{S}^-(\varphi)$ are satisfied.

Definition 4.1.8. For every classical and intuitionistic structure $\mathcal{M}$ define a structure $\mathcal{S}(\mathcal{M}, \varphi)$ that agrees with $\mathcal{M}$ on everything except the interpretation(s) of atoms of the form P_ψ. By slight abuse of notation, we denote with $\vec{z}_\psi$ elements of the domain instead of variables. In the classical case, we set

$$\begin{aligned}
P^I_A(\vec{z}_A) &:\Leftrightarrow A^I(\vec{z}_A)\\
P^I_{\varphi\wedge\psi}(\vec{z}_{\varphi\wedge\psi}) &:\Leftrightarrow P^I_\varphi(\vec{z}_\varphi) \wedge P^I_\psi(\vec{z}_\psi)\\
P^I_{\varphi\vee\psi}(\vec{z}_{\varphi\vee\psi}) &:\Leftrightarrow P^I_\varphi(\vec{z}_\varphi) \vee P^I_\psi((\vec{z}_\psi))\\
P^I_{\varphi\to\psi}(\vec{z}_{\varphi\to\psi}) &:\Leftrightarrow (\neg P^I_\varphi(\vec{z}_\varphi)) \vee P^I_\psi(\vec{z}_\psi)\\
P^I_{\forall x\varphi}(\vec{z}_{\forall x\varphi}) &:\Leftrightarrow \{\vec{z} \mid\ P^I_\varphi(\vec{z}_\varphi) \text{ for all } x \in M\}\\
P^I_{\exists x\varphi}(\vec{z}_{\exists x\varphi}) &:\Leftrightarrow \{\vec{z} \mid\ P^I_\varphi(\vec{z}_\varphi) \text{ for some } x \in M\},
\end{aligned}$$

and for intuitionistic logic and each world u, we set

$$\begin{aligned}
P^{I_u}_A(\vec{z}_A) &:\Leftrightarrow A^{I_u}(\vec{z}_A)\\
P^{I_u}_{\varphi\wedge\psi}(\vec{z}_{\varphi\wedge\psi}) &:\Leftrightarrow P^{I_u}_\varphi(\vec{z}_\varphi) \wedge P^{I_u}_\psi(\vec{z}_\psi)\\
P^{I_u}_{\varphi\vee\psi}(\vec{z}_{\varphi\vee\psi}) &:\Leftrightarrow P^{I_u}_\varphi(\vec{z}_\varphi) \vee P^{I_u}_\psi((\vec{z}_\psi))\\
P^{I_u}_{\varphi\to\psi}(\vec{z}_{\varphi\to\psi}) &:\Leftrightarrow (\neg P^{I_w}_\varphi(\vec{z}_\varphi)) \vee P^{I_w}_\psi(\vec{z}_\psi) \text{ for all } w \geq u\\
P^{I_u}_{\forall x\varphi}(\vec{z}_{\forall x\varphi}) &:\Leftrightarrow P^{I_w}_\varphi(\vec{z}_\varphi) \text{ for all } w \geq u,\ x \in M_w\\
P^{I_u}_{\exists x\varphi}(\vec{z}_{\exists x\varphi}) &:\Leftrightarrow P^{I_u}_\varphi(\vec{z}_\varphi) \text{ for some } x \in M_u.
\end{aligned}$$

From this definition, we directly obtain the following Lemmata by induction on the formula height.

Lemma 4.1.9. For every formula φ, structure $\mathcal{M}$, we have $\mathcal{S}(\mathcal{M}, \varphi) \models \mathcal{S}^+(\varphi) \cup S^-(\varphi)$.

Lemma 4.1.10. $\mathcal{M} \models \varphi$ if and only if $S(\mathcal{M}, \varphi) \models P_\varphi(\vec{z}_\varphi)$.

Then from the previous three Lemmata we get:

Corollary 4.1.11. In both intuitionistic and classical logic,

- φ is satisfiable iff $\bigwedge S^+(\varphi) \wedge P_\varphi(\vec{z}_\varphi)$ is, and
- φ is valid iff $\bigwedge \mathcal{S}^-(\varphi) \to P_\varphi(\vec{z}_\varphi)$ is.

Example 4.1.12. Consider $\varphi = \neg\forall x A(x) \to \exists x \neg A(x)$. Then

$$\begin{aligned}
\mathcal{S}^+(\varphi) =& \{(P_\varphi \wedge P_{\neg\forall x A(x)}) \to P_{\exists x \neg A(x)}, (P_{\forall x A(x)} \to \bot) \to P_{\neg\forall x A(x)}\} \\
& \cup \{\forall x P_{A(x)}(x) \to P_{\forall x A(x)}, \forall x(P_{A(x)}(x) \to A(x))\} \\
& \cup \{P_{\exists x \neg A(x)} \to \exists x P_{\neg A(x)}(x), \forall x(P_{\neg A(x)}(x) \to P_{A(x)}(x) \to \bot)\} \\
& \cup \{\forall x(A(x) \to P_{A(a)}(x))\} \\
\mathcal{S}^-(\varphi) =& \{(P_{\neg\forall x A(x)} \to P_{\exists x \neg A(x)}) \to P_\varphi, (P_{\neg\forall x A(x)} \wedge P_{\forall x A(x)}) \to \bot\} \\
& \cup \{P_{\forall x A(x)} \to \forall x P_{A(x)}(x), \forall x(A(x) \to P_{A(x)}(x))\} \\
& \cup \{\exists x P_{\neg A(x)}(x) \to P_{\exists x \neg A(x)}, \forall x((P_{A(x)}(x) \to \bot) \to P_{\neg A(x)}(x))\} \\
& \cup \{\forall x(P_{A(a)}(x) \to A(x))\}
\end{aligned}$$

4.1.3 IPC to simplified CQC

We are now ready to combine the previously presented reductions. Let φ be some propositional formula. As a first step we perform the normalization from the last section, i.e. instead of φ we consider

$$\varphi' = \bigwedge \mathcal{S}^-(\varphi) \to P_\varphi.$$

Recall that since φ is quantifier-free, each formula $\psi \in \mathcal{S}^-(\varphi)$ is of one of the forms

$$A \to (B \wedge C), (A \wedge B) \to C, A \to (B \vee C), (A \vee B) \to C, (A \to B) \to C,$$

where we can treat $A \to B$ as a special case $A \to (B \wedge B)$. We then apply the transformation to CQC, leading to a formula

$$\varphi'' = K(\varphi') \to \bigwedge \mathcal{S} \to P_\varphi(u),$$

where u is some new free variable and each $\psi \in \mathcal{S}$ is of one of the forms

$$\begin{gathered}
\forall k(u \preceq k \to A(k) \to (B(k) \wedge C(k))) \qquad \forall k(u \preceq k \to (A(k) \wedge (B(k)) \to C(k))) \\
\forall k(u \preceq k \to A(k) \to (B(k) \vee C(k))) \qquad \forall k(u \preceq k \to (A(k) \vee B(k)) \to C(k)) \\
\forall k(u \preceq k \to (\forall l(k \preceq l \to A(l) \to B(l))) \to C(k)),
\end{gathered}$$

with each $P(x)$ possibly being $\bot$. Applying Herbrandization, we obtain a formula

$$\varphi''' = K(\varphi') \to \bigwedge \mathcal{S}' \to P_\varphi(s),$$

where s is some new constant term and each $\psi \in \mathcal{S}'$ is of one of the forms

$$\begin{gathered}
\forall k(s \preceq k \to A(k) \to (B(k) \wedge C(k))) \qquad \forall k(s \preceq k \to (A(k) \wedge (B(k)) \to C(k))) \\
\forall k(s \preceq k \to A(k) \to (B(k) \vee C(k))) \qquad \forall k(s \preceq k \to (A(k) \vee (B(k)) \to C(k))) \\
\forall k(s \preceq k \to (k \preceq f_\psi(k) \to A(f_\psi(k)) \to B(f_\psi(k))) \to C(k)),
\end{gathered}$$

and f_ψ is a new function symbol for each introduced formula of that form. Now if (M, I) is a counter-model, then we have a counter-model (M', I') where $M' = \{m \in M | s \preceq m\}$ and $f_\psi^{I'}(u) = f_\psi^I(u)$ iff $u \preceq f_\psi^I(u)$, and $f_\psi^{I'}(u) = u$, else. That is, where s is the least element and the f_ψ increase their argument. Therefore, instead of φ''' we may consider a formula

$$\varphi^\circ = \forall k(s \preceq k) \to \bigwedge\{\forall k(k \preceq f_\psi(k)) \mid \psi \in \mathcal{F}_\to\} \to K(\varphi') \to \bigwedge \mathcal{S}^\circ \to P_\varphi(s)$$

where each formula $\psi \in \mathcal{S}^\circ$ is of one of the forms

$$\begin{gathered}\forall k(A(k) \to (B(k) \land C(k))) \qquad \forall k((A(k) \land B(k)) \to C(k)) \\ \forall k(A(k) \to (B(k) \lor C(k))) \qquad \forall k((A(k) \lor B_\psi(k)) \to C(k)) \\ \forall k((A(f_\psi(k)) \to B(f_\psi(k))) \to C(k)),\end{gathered}$$

and $\mathcal{F}_\to$ denotes the set of formulas of the last kind.

4.1.4 IPC to CPC

We now argue that if a counter-model exists for φ°, then a bounded counter-model exists, allowing us to return to the propositional case.

Definition 4.1.13. Suppose we are given a counter-model $\mathcal{M} = (M, I)$ for φ°. We say $\psi \in \mathcal{F}_\to$ is *fulfilled* at $u \in M$ iff $A_\psi^I(u) \to B_\psi^I(u)$ is false or $C_\psi^I(u)$ is true. For $\psi \in \mathcal{F}_\to$ define

$$g_\psi : M \to M, u \mapsto \begin{cases} u, & \text{if } \psi \text{ is fulfilled in } u, \\ f_\psi^I(u), & \text{else.} \end{cases}$$

Define a model $\mathcal{M}_T = (M_T, I_T)$ as follows:

- M_T is the set of sequences without repetition on $\mathcal{F}_\to$.
- Interpret $\preceq$ as the prefix-order.
- We set

$$f_\psi^{I_T}(\psi_1 \dots \psi_n) = \begin{cases} \psi_1 \dots \psi_n, & \text{if } \psi \text{ occurs in } \psi_1 \dots \psi_n, \\ \psi_1 \dots \psi_n \psi, & \text{else.} \end{cases}$$

- For propositional variables P, we set

$$P^{I_T}(\psi_1 \dots \psi_n) = P^I(g_{\psi_n}(\dots(g_{\psi_1}(s^I))\dots)).$$

Lemma 4.1.14. Let $\mathcal{M} = (M, I)$ be a counter-model to φ°.

1. ψ is fulfilled at $f_\psi^I(u)$ for all $u \in M, \psi \in \mathcal{F}_\to$.
2. If ψ is fulfilled at some $u \in M$ then ψ is fulfilled at all $w \succeq u$.

Proof. 1. If $C^I(u)$ is true then we are done due to persistency. Otherwise $C^I(u)$ is false. Because $(A^I(f^I_\psi(u)) \to B^I(f^I_\psi(u))) \to C^I(u)$ holds, then $A^I(f^I_\psi(u)) \to B^I(f^I_\psi(u))$ must be false.

2. Let w be some element with $w \geq u$. If $C^I(w)$ is true, we are done. Otherwise $C^I(w)$ is false. Due to persistency $C^I(u)$ is also false. Because ψ is fulfilled at u, $A^I(u) \to B^I(u)$ must be false, i.e. $A^I(u)$ is true and $B^I(u)$ is false. Then $A^I(w)$ and $A^I(f^I_\psi(w))$ are also true due to persistency. Because $(A^I(f^I_\psi(w)) \to B^I(f^I_\psi(w))) \to C^I(w)$ holds, we now get that $B^I(f^I(w))$ must be false. But then due to persistency so is $B^I(w)$ and we have that ψ is fulfilled at w. $\square$

Essentially, what this tells us is that for every u at which ψ is fulfilled, in particular for each u in the image of f_ψ, we may add the assumption that $f_\psi(w) = w$ for all $w \geq u$ without changing the validity of ψ. With this lemma in hand, showing that (M_T, I_T) is a counter-model to φ° is a straightforward check of definitions.

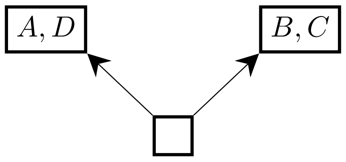

Figure 4.1: Counter-model $\bigwedge S \to E$. (Example 4.1.16.)

Corollary 4.1.15. If $\mathcal{M}$ is a counter-model to φ° then so is $\mathcal{M}_T$.

Example 4.1.16. Consider

$$\mathcal{S} = \{(A \to B) \to C, (B \to A) \to D, (A \wedge B) \to \bot, (C \vee D) \to E\}$$

and $\varphi = \bigwedge S \to E$ which has the Kripke counter-model depicted in 4.1 where at each node the true propositions are indicated.

There is a corresponding classical counter-model $\mathcal{M} = (M, I)$ to φ° with $M = \{u, u_l, u_r\}$, $A^I(l) = D^I(l) = B^I(r) = C^I(r) = 1$ and 0 else. Denoting $\psi_1 := (A \to B) \to C$ and $\psi_2 := (B \to A) \to D$, this corresponds to a transformed counter-model $\mathcal{M}_T = (M_T, I_T)$ of φ° and interpretation as presented in 4.2.

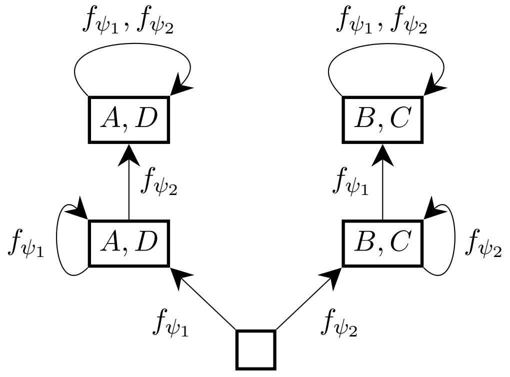

Figure 4.2: $\mathcal{M}_T$

Remark 4.1.17. It is well known that Kripke frames which are trees are complete with regard to intuitionistic logic. Our contribution is giving an explicit geometry that depends only on the structure of the considered formula.

So we know that φ° is valid if and only if it is valid for structures that have as a domain the sequences without repetition over $\mathcal{F}_\rightarrow$, ordered by the prefix-relation and

$$s^I = \epsilon, f^I_\psi(x) = \begin{cases} x, & \text{if } \psi \text{ occurs in } x, \\ x\psi, & \text{otherwise.} \end{cases}$$

Now we can replace every $\forall$-quantifier by enumerating all the ground terms, and then replace all distinct ground instances of relations by new propositional variables yielding a propositional formula. This gives us the following theorem.

Theorem 4.1.18. Let $\mathcal{S}$ be a set of formulas of the form

$$(A \wedge B) \rightarrow C, (A \vee B) \rightarrow C, A \rightarrow (B \wedge C), A \rightarrow (B \vee C), (A \rightarrow B) \rightarrow C$$

and $\mathcal{F}_\rightarrow \subseteq \mathcal{S}$ denote the subset of formulas of the form $(A \rightarrow B) \rightarrow C$ and Λ denote the set of sequences without repetition over $\mathcal{F}_\rightarrow$. For each atom A and $k \in \Lambda$ consider a new atom A^k. Obtain $\mathcal{S}^\#$ by including the following formulas:

- $A^k \rightarrow A^{k\psi}$ for each atom A occurring in $\mathcal{S}$, $k \in \Lambda$ and $\psi \in \mathcal{F}_\rightarrow$ not occurring in k.
- $A^k \rightarrow (B^k \circ C^k)$ for each $\circ \in \{\wedge, \vee\}$, $A \rightarrow (B \circ C) \in \mathcal{S}$, $k \in \Lambda$.
- $(A^k \circ B^k) \rightarrow C^k$ for each $\circ \in \{\wedge, \vee\}$, $A \rightarrow (B \circ C) \in \mathcal{S}$, $k \in \Lambda$.
- $(A^{k\psi} \rightarrow B^{k\psi}) \rightarrow C^k$ for $\psi = (A \rightarrow B) \rightarrow C \in \mathcal{S}$, $k \in \Lambda$ if ψ does not occurr in k.

Then, $\bigwedge S \rightarrow P$ is intuitionistically valid iff $\bigwedge S^\# \rightarrow P^\epsilon$ is classically valid, where ϵ denotes the empty sequence. The size of $\mathcal{S}^\#$ is in $\mathcal{O}(|\mathcal{S}| \cdot 2^{|\mathcal{F}_\rightarrow| \cdot \log(|\mathcal{F}_\rightarrow|)})$. Furthermore, there is an effective procedure for translating counter-models between the sentences.

Example 4.1.19. Let us consider the law of excluded middle $\varphi = A \vee \neg A$. Then we get

$$\mathcal{S}^-(\varphi) = \{(P_A \vee P_{\neg A}) \rightarrow P_\varphi, A \rightarrow P_A, (A \rightarrow \bot) \rightarrow P_{\neg A}\},$$

and we know that φ is valid iff $\bigwedge \mathcal{S}^-(\varphi) \rightarrow P_\varphi$ is valid. Applying the above it is intuitionistically valid iff $\bigwedge \mathcal{S}^\# \rightarrow P^\emptyset_\varphi$ is classically valid, where, setting $\psi = (A \rightarrow \bot) \rightarrow P_{\neg A}$,

$$\begin{aligned}\mathcal{S}^\# =& \{P^\epsilon_\varphi \rightarrow P^\psi_\varphi, P^\epsilon_A \rightarrow P^\psi_A, P^\epsilon_{\neg A} \rightarrow P^\psi_{\neg A}, A^\epsilon \rightarrow A^\psi, (P^\epsilon_A \vee P^\epsilon_{\neg A}) \rightarrow P^\epsilon_\varphi, (P^\psi_A \vee P^\psi_{\neg A}) \rightarrow P^\psi_\varphi\} \cup \\ & \{A^\epsilon \rightarrow P^\epsilon_A, P^\psi_A \rightarrow P^\psi_A, (A^\psi \rightarrow \bot) \rightarrow P^\epsilon_{\neg A}, (A^\psi \rightarrow \bot) \rightarrow P^\psi_{\neg A}\}.\end{aligned}$$

We now note that $P^\epsilon_\varphi = P^\epsilon_A = P^\epsilon_{\neg A} = A^\epsilon = 0$, $P^\psi_\varphi = P^\psi_A = P^\psi_{\neg A} = A^\psi = 1$ defines a counter-model.

4.2 IPC to QBF

It is known that a polynomial time translation between QBF and IPC must exist, since both problems are PSPACE complete [14, 25]. However, giving an explicit translation from IPC to QBF still promises to be useful, allowing to leverage the progress in QBF solving. The translation from the last section will serve as starting point for developing the translation in this section. We do not define the semantics of QBF and refer the reader to standard sources e.g. [9].

Fix some set $\mathcal{S}$ of formulas of the form

$$A \to (B \wedge C), (A \wedge B) \to C, A \to (B \vee C), (A \vee B) \to C, (A \to B) \to C$$

and

$$\varphi = \bigwedge \mathcal{S} \to P,$$

and denote with $\mathcal{F}_{\to}$ the set of formulas of the form $(A \to B) \to C$. We will now formulate a QBF that expresses that φ has an intuitionistic counter-model. The idea how to get a polynomially-sized formula is: Instead of expressing validity at every node of $\mathcal{M}_T$ — which was done in the CPC translation — we express that on each path in $\mathcal{M}_T$ the nodes satisfy all conditions for $\mathcal{M}_T$ being a counter-model. The universally quantified variables F_ψ in the next definition handle the branching, i.e. express which path is considered.

Definition 4.2.1. For every propositional variable X occurring in $\mathcal{S}$ and non-negative integer n consider a new atom X^n and for every formula $\psi \in \mathcal{F}_{\to}$ consider a new atom F^n_ψ. Let $\vec{X}^n$ range over all propositional variables X^n and $\vec{F}^n$ range over all F^n_ψ. Define

- Valid(n), which encodes that $\vec{F}^n_\psi$ represents a valid next element of a sequence without repetition, i.e. if we interpret the $\vec{F}^i$ as bit-vectors, then $\vec{F}^n$ has exactly one bit set to 1, indicating the formula at the n-th position of the sequence, and it is at a position that is 0 in $\sum\{\vec{F}^i \mid i < n\}$.
- Persistent(n), which encodes that the persistency condition holds.
- Sat$_{\mathcal{S}}(n)$, encoding that the formulas in $\mathcal{S} \setminus \mathcal{F}_{\to}$ hold at the n-th world of the path.
- Sat$_{\mathcal{F}_{\to}}(n)$, encoding that the formula in $\mathcal{F}_{\to}$ that is represented by $\vec{F}^{n-1}$ holds.

Have $k = |\mathcal{F}_{\to}|$ and define

$$\varphi^Q_i = \exists \vec{X}^i \forall \vec{F}^i \Big(\text{Persistent}(i) \wedge \text{Sat}_{\mathcal{S}}(i) \wedge \text{Sat}_{\mathcal{F}_{\to}}(i) \wedge \Big(\text{Valid}(i) \to \varphi^Q_{i+1}\Big)\Big)$$

for $0 < i < k$, as well as the special cases

$$\varphi^Q_k = \exists \vec{X}^k \left(\text{Persistent}(k) \wedge \text{Sat}_{\mathcal{S}}(k) \wedge \text{Sat}_{\mathcal{F}_{\to}}(k)\right)$$

for leafs, and

$$\varphi^Q = \varphi^Q_0 = \exists \vec{X}^0 \forall \vec{F}^0 \left(\neg P^0 \wedge \mathrm{Sat}_{\mathcal{S}}(0) \wedge (\mathrm{Valid}(0) \to \varphi^Q_1) \right)$$

for the root. Example encodings of the above formulas are:

$$\begin{aligned}
\mathrm{Valid}(n) &= \left(\bigvee \{ \mathcal{F}^n_\psi \mid \psi \in \mathcal{F}_\to \} \right) \wedge \left(\bigwedge \{ \neg (F^n_{\psi_1} \wedge F^n_{\psi_1}) \mid \psi_1 \neq \psi_2 \in \mathcal{F}_\to \} \right) \wedge \\
&\quad \left(\bigwedge \left\{ \bigwedge \{ F^i_\psi \to \neg F^n_\psi \mid i < n \} \mid \psi \in \mathcal{F}_\to \right\} \right) \\
\mathrm{Persistent}(n) &= \bigwedge \{ X^{n-1} \to X^n \mid X \text{ prop. variable with } X^{n-1} \in \vec{X}^{n-1}, X^n \in \vec{X}^n \} \\
\mathrm{Sat}_{\mathcal{S}}(n) &= \bigwedge \{ (A^n \circ B^n) \to C^n \mid (A \circ B) \to C \in \mathcal{S} \setminus \mathcal{F}_\to \} \wedge \\
&\quad \bigwedge \{ A^n \to (B^n \circ C^n) \mid A \to (B \circ C) \in \mathcal{S} \setminus \mathcal{F}_\to \} \\
\mathrm{Sat}_{\mathcal{F}_\to}(n) &= \bigwedge \{ F^{n-1}_\psi \to (A^n \to B^n) \to C^{n-1} \mid \psi = (A \to B) \to C \in \mathcal{F}_\to \}
\end{aligned}$$

Example 4.2.2. With the previous encoding for the double negation elimination

$$\varphi = ((A \to \bot) \to \bot) \to A$$

we have

$$\varphi^Q = \exists A^0 \forall F^0 \left(\neg A^0 \wedge \left(F^0 \to \exists A^1 \left((A^0 \to A^1) \wedge (F_0 \to (A^1 \to \bot) \to \bot) \right) \right) \right),$$

which is a satisfiable QBF since $A^0 = 0, A^1 = 1$ satisfies it for any F^0.

Lemma 4.2.3. φ is not intuitionistically valid iff φ^Q is a satisfiable QBF.

Proof. Suppose φ is not intuitionistically valid. Then, $\bigwedge \mathcal{S} \to P$ has an intuitionistic counter-model. Then, by the previous section there exists a classical counter-model $\mathcal{M}$ for $\bigwedge \mathcal{S}^\# \to P^\epsilon$. We now define a QBF model iteratively. For each atom A interpret A^0 such as $\mathcal{M}$ interprets A^ϵ. Suppose we are given interpretations of all atoms A^i for $i < n$ and a sequence with no repetitions $\psi_1 \dots \psi_{n-1}$ over $\mathcal{F}_\to$ such that ψ_i is exactly the $\psi \in \mathcal{F}_\to$ for which F^i_ψ is true and A^i is interpreted as $A^{\psi_1 \cdots \psi_i}$ in $\mathcal{M}$. Let the F^n_ψ be arbitrarily interpreted (since they are $\forall$-quantified). If not exactly one of the F^n_ψ is interpreted as true, then valid(n) fails and the remaining propositions can be chosen arbitrarily. If $F^i_{\psi_n} = 1$ for some $i < n$, then valid(n) also fails and the remaining propositions can be chosen arbitrarily. So we may assume that $\psi_1 \dots \psi_n$ is a sequence with no repetitions. Interpret the atoms A^n as $\mathcal{M}$ interprets $A^{\psi_1 \cdots \psi_n}$. Continue this construction until $n = |\mathcal{F}_\to|$. Then from $\mathcal{M}$ being a counter-example to $\mathcal{S}^\# \to P$ it directly follows that this interpretation satisfies φ^Q.

On the other hand, suppose φ^Q is satisfiable. We construct a counter-example to $\varphi^\#$. Again, we proceed iteratively. Interpret A^ϵ such as A^0 is interpreted in some satisfying interpretation of φ^Q. Suppose we are given a sequence $\psi_1 \dots \psi_{n-1}$ such that for $i < n$

having $A^i = A^{\psi_0 \cdots \psi_{n-1}}$ is part of a satisfying interpretation of φ^Q, in which F^i_ψ is chosen true iff $\psi = \psi_i$. Let $\psi_n \in \mathcal{F}_\rightarrow$. Consider some interpretation of the A^n that are part of a satisfying assignment where F^n_ψ is true iff $\psi = \psi_n$ and all variables quantified above are chosen as before. Have $A^{\psi_0 \cdots \psi_n} = A^n$ for each propositional variable A. From the definitions it directly follows that construction yields a counter-model for $\varphi^\#$. □

A simple counting argument shows that for each $n < |\mathcal{F}_\rightarrow|$ we have $|\mathrm{Valid}(n)| \in \mathcal{O}(|\mathcal{F}_\rightarrow|^2)$, $|\mathrm{Persistent}(n)|, |\mathrm{Sat}_\mathcal{S}(n)| \in \mathcal{O}(|\varphi|)$ and $|\mathrm{Sat}_{\mathcal{F}_\rightarrow}(n)| \in \mathcal{O}(|\mathcal{F}_\rightarrow|)$ so overall we get:

Lemma 4.2.4. The size of φ^Q is in $\mathcal{O}(|\varphi| \cdot |\mathcal{F}_\rightarrow| + |\mathcal{F}_\rightarrow|^3)$.

Note that the last set of universally quantified variables can be avoided because there is only one assignment that has the chance to falsify φ^Q, namely the assignment that assigns 1 to the single F^n_ψ such that $F^i_\psi = 0$ for all $i < n$. Hence, in our final formula we can replace every F^n_ψ with $\bigwedge\{\neg F^i_\psi \mid i < n\}$ and remove that quantification over F^n_ψ. With that we get the following:

Corollary 4.2.5. Let N-`IntInvalid` be the problem of deciding if a formula $\bigwedge \mathcal{S} \rightarrow P$ as above with $|\mathcal{F}_\rightarrow| \leq N$ is not intuitionistically valid. Then N-`IntInvalid` is in Σ^P_{2N-1}. The dual problem N-`IntValid` is in Π^P_{2N-1}.

4.3 CQC to IQC

We now give an analogous transformation for first order logic. Our trick of lifting the sentence to first-order logic to encode the Kripke semantics no longer works in such a straightforward manner, since we already are in first-order logic. However, we do it nonetheless! The domain of our classical model will feature on the one hand elements representing worlds in a Kripke frame and on the other hand the domain-elements from the Kripke model. We reconcile these notions by introducing a special binary predicate E, first considered in [7] and expanded onto in [20], where $E(x, u)$ encodes that x is an element of the domain of the world u. We fix some signature Σ for this section. To encode that some n-ary relation A holds at a world u we replace each n-ary relation symbol A with a $n+1$-ary relation symbol $A^\#$, interpreting $A^\#(\vec{x}, u)$ as "$A(\vec{x})$ holds at u". Furthermore we write $\vec{E}(\vec{x}, u)$ for $\bigwedge\{E(x_i, u) \mid x_i \in \vec{x}\}$.

First of all recall, that due to Lemma 4.1.11 it is sufficient to consider sentences of the form $\bigwedge \mathcal{S} \rightarrow P$ where all formulas in $\mathcal{S}$ are of one of the forms

$$\forall\vec{z}(A(\vec{a}) \rightarrow (B(\vec{b}) \wedge C(\vec{c}))) \qquad \forall\vec{z}((A(\vec{a}) \wedge B(\vec{b})) \rightarrow C(\vec{c})) \qquad \forall\vec{z}(A(\vec{a}) \rightarrow (B(\vec{b}) \vee C(\vec{c})))$$
$$\forall\vec{z}((A(\vec{a}) \vee B(\vec{b})) \rightarrow C(\vec{c})) \qquad \forall\vec{z}((A(\vec{a}) \rightarrow B(\vec{b})) \rightarrow C(\vec{c})) \qquad \forall\vec{z}(\forall x A(\vec{a}) \rightarrow B(\vec{b}))$$
$$\forall\vec{z}(A(\vec{a}) \rightarrow \forall x B(\vec{b})) \qquad \forall\vec{z}(\exists x A(\vec{a}) \rightarrow B(\vec{b})) \qquad \forall\vec{z}(A(\vec{a}) \rightarrow \exists x B(b)).$$

The complete result that we will show is as follows:

Theorem 4.3.1. Let $\mathcal{S}$ be as above and $\mathcal{F}_{\rightarrow} \subseteq \mathcal{S}$ contain the formulas of the form $\forall\vec{t}((A(\vec{z}_A) \rightarrow B(\vec{z}_B)) \rightarrow C(\vec{z}_C))$ and $\mathcal{F}_{\forall} \subseteq \mathcal{S}$ the formulas of the form $\forall\vec{z}(\forall x A(\vec{z}_A) \rightarrow B(\vec{z}_B))$. Let $\mathcal{F} = \mathcal{F}_{\rightarrow} \cup \mathcal{F}_{\forall}$. For every n-ary atomic relation R define a new $n+1$-ary relation $R^{\#}$. For every $\psi \in \mathcal{F}$ define a new function symbol f_ψ. Consider a new binary relation symbol E and define $\vec{E}(\vec{x}, u) := \bigwedge\{E(x,u) \mid x \in \vec{x}\}$. Obtain $\mathcal{S}^{\#}$ by including formulas as follows:

- $\forall\vec{z}\forall u(\vec{E}(\vec{z},u) \rightarrow (A^{\#}(\vec{a},u) \rightarrow (B^{\#}(\vec{b},u) \circ C^{\#}(\vec{c},u))))$
  for each $\circ \in \{\wedge, \vee\}, \psi = \forall\vec{z}(A(\vec{a}) \rightarrow (B(\vec{b}) \circ C(\vec{c}))) \in \mathcal{S}$
- $\forall\vec{z}\forall u(\vec{E}(\vec{z},u) \rightarrow (A^{\#}(\vec{a},u) \circ B^{\#}(\vec{b},u)) \rightarrow C^{\#}(\vec{c},u))$
  for each $\circ \in \{\wedge, \vee\}, \psi = \forall\vec{z}((A(\vec{a}) \circ B(\vec{b})) \rightarrow C(\vec{c})) \in \mathcal{S}$
- $\forall\vec{z}\forall u(\vec{E}(\vec{z},u) \rightarrow (A^{\#}(\vec{a}, f_\psi(\vec{z},u)) \rightarrow B^{\#}(\vec{b}, f_\psi(\vec{z},u))) \rightarrow C^{\#}(\vec{c},u))$
  for each $\psi = \forall\vec{z}((A(\vec{a}) \rightarrow B(\vec{b})) \rightarrow C(\vec{c})) \in \mathcal{S}$
- $\forall\vec{z}\forall u(\vec{E}(\vec{z},u) \rightarrow \forall x(E(x, f_\psi(\vec{z},u)) \rightarrow A^{\#}(\vec{a}, f_\psi(\vec{z},u))) \rightarrow B^{\#}(\vec{b},u))$
  for each $\psi = \forall\vec{z}(\forall x A(\vec{a}) \rightarrow B(\vec{t}_B)) \in \mathcal{S}$.
- $\forall\vec{z}\forall u(\vec{E}(\vec{z},u) \rightarrow A^{\#}(\vec{a},u) \rightarrow \forall x(E(x,u) \rightarrow B^{\#}(\vec{b},u)))$
  for each $\forall\vec{t}(A(\vec{a}) \rightarrow \forall x B(\vec{b})) \in \mathcal{S}$.
- $\forall\vec{z}\forall u(\vec{E}(\vec{z},u) \rightarrow \exists x(E(x,u) \wedge A^{\#}(\vec{z},u)) \rightarrow B^{\#}(\vec{b}))$
  for each $\forall\vec{z}(\exists x A(\vec{a}) \rightarrow B(\vec{b})) \in \mathcal{S}$
- $\forall\vec{z}\forall u(\vec{E}(\vec{z},u) \rightarrow A^{\#}(\vec{b}) \rightarrow \exists x(E(x,u) \wedge B^{\#}(\vec{b},u)))$
  for each $\forall\vec{z}(A(\vec{a}) \rightarrow \exists x B(\vec{b})) \in \mathcal{S}$

Then $\bigwedge S \rightarrow P$ is intuitionistically iff $\exists x E(x,s) \rightarrow K \rightarrow \bigwedge \mathcal{S}^{\#} \rightarrow P^{\#}(s)$ is classically valid, where s is a new constant symbol, Σ is the original signature and

$$\begin{aligned}K := &\bigwedge\{\forall\vec{z}\forall x\forall u((E(x,u) \rightarrow E(x, f_\psi(\vec{z},u))) \mid \psi \in \mathcal{F}\} \wedge \\ &\forall u\left(\exists x E(x,u) \rightarrow \bigwedge\{\forall\vec{z}(\vec{E}(\vec{z},u) \rightarrow E(f(\vec{z}),u)) \mid f \text{ is a function symbol in } \Sigma\}\right) \wedge \\ &\bigwedge\{\forall\vec{z}_1\forall\vec{z}_2\forall u(A^{\#}(\vec{z}_1,u) \rightarrow A^{\#}(\vec{z}_1, f_\psi(\vec{z}_2,u))) \mid \psi \in \mathcal{F},\ A \text{ is a relation symbol in } \Sigma\}.\end{aligned}$$

The size of the obtained formula is linear in the input. However $|\mathcal{F}| + 1$ new function symbols and the new binary predicate E were introduced and each n-ary relation symbol has been extended to a $n+1$-ary relation symbol.

Example 4.3.2. Consider the classical tautology $\varphi := \neg\forall x\neg A(x) \rightarrow \exists x A(x)$. Setting

$\psi := \forall x \neg A(x)$ we have

$$\begin{aligned} S^{\#}(\varphi) =& \{\forall u((P^{\#}_{\neg\psi}(f_\varphi(u)) \to P^{\#}_{\exists x A(x)}(f_\varphi(u))) \to P^{\#}_{\varphi}(u))\} \cup \\ & \{\forall u(\exists x(E(x,u) \land A^{\#}(x,u)) \to P^{\#}_{\exists x A(x)}(u)), \forall u((P^{\#}_{\neg\psi}(u) \land P^{\#}_{\psi}(u)) \to \bot)\} \cup \\ & \{\forall u(\forall x(E^{\#}(x, f_\psi(u)) \to P^{\#}_{\neg A(x)}(x, f_\psi(u))) \to P^{\#}_{\psi}(u))\} \cup \\ & \{\forall x \forall u(E(x,u) \to (A^{\#}(x, f_{\neg A(x)(u)}) \to \bot) \to P^{\#}_{\neg A(x)}(x,u))\} \end{aligned}$$

And as one might expect we can indeed define a counter-model for

$$\exists x E(x,s) \to K \to \bigwedge S^{\#} \to P^{\#}(s),$$

namely take as a domain $M = \{m, v, w\}$ where u, w represent worlds, and m representing a single domain element. Have $f^I_\varphi(x) = f^I_\psi(x) = f^I_{\neg A(x)}(x) = w$ for all $x \in M$, $E^I(x,y)$ if and only if $x = m$ and $y \in \{u, v\}$ and finally $A^{\#}(x,y)$ iff $x = m$ and $y = w$.

It is not possible to completely eliminate worlds from the domain as in the propositional case, since counter-models can possibly be infinite (as we will see in more detail later).

4.3.1 Encoding Kripke Semantics

As in the propositional case we must encode the Kripke Semantics, this time not only the relation $\preceq$, but also E has to be axiomatized. For that we use the following predicates:

- PartialOrder($\preceq$), encoding that $\preceq$ is a partial order.
- DomainSubset(u, w), encoding that the domain at u is a subset of that at w.
- World(u), encoding that u represents a world.
- DomainClosed(u), which encodes that at u the domain is closed under functions, e.g. contains all interpretations of constants.
- Persistent(u, w), which encodes that persistency is satisfied between u and w, i.e. for all predicates A and domain elements x with $A(x)$, at u we have $A(x)$ at w.

Then, analogously to the propositional case, we define:

$$\begin{aligned} K(\varphi) =& \text{PartialOrder}(\preceq) \land \forall u \forall w(u \preceq w \to \text{DomainSubset}(u,w)) \land \\ & \forall u(\text{World(u)} \to \text{DomainClosed}(u)) \land \forall u \forall w(u \preceq w \to \text{Persistent}(u,w)). \end{aligned}$$

Example encodings of these formulas are as follows:

$$\begin{aligned}
\text{PartialOrder}(\preceq) &= \forall u(u \preceq u) \wedge \forall u \forall w(u \preceq w \to w \preceq u \to u = w) \wedge \\
&\quad \forall u \forall v \forall w(u \preceq v \to v \preceq w \to u \preceq w) \\
\text{DomainSubset}(u, w) &= \forall z(E(z, u) \to E(z, w)) \\
\text{World}(u) &= \exists x E(x, u) \\
\text{DomainClosed}(u) &= \bigwedge \{\forall \vec{z}(\vec{E}(\vec{z}, u) \to E(f(\vec{z}), u)) \mid f \in \Sigma \text{ is a function symbol}\} \\
\text{Pesistent}(u, w) &= \bigwedge \{\forall \vec{z}(\vec{A}^{\#}(\vec{z}, u) \to A^{\#}(\vec{z}, w)) \mid A \in \Sigma \text{ is a relation symbol}\}
\end{aligned}$$

Using E, we then obtain $\mathcal{S}'$ analogously to the predicate case by including formulas:

- $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, u) \to (B^{\#}(\vec{b}, u) \circ C^{\#}(\vec{c}, u))))$
  for each $\circ \in \{\wedge, \vee\}, \psi = \forall \vec{z}(A(\vec{a}) \to (B(\vec{b}) \circ C(\vec{c}))) \in \mathcal{S}$
- $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, u) \circ B^{\#}(\vec{b}, u)) \to C^{\#}(\vec{c}, u))$
  for each $\circ \in \{\wedge, \vee\}, \psi = \forall \vec{z}((A(\vec{a}) \circ B(\vec{b})) \to C(\vec{c})) \in \mathcal{S}$
- $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to \forall w(u \preceq w \to A^{\#}(\vec{a}, w) \to B^{\#}(\vec{b}, w)) \to C^{\#}(\vec{c}, u))$
  for each $\psi = \forall \vec{z}((A(\vec{a}) \to B(\vec{b})) \to C(\vec{c})) \in \mathcal{S}$
- $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to \forall w(u \preceq w \to \forall x(E(x, w) \to A^{\#}(\vec{a}, w))) \to B^{\#}(\vec{b}, u))$
  for each $\forall \vec{z}(\forall x A(\vec{a}) \to B(\vec{t}_B)) \in \mathcal{S}$.
- $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to A^{\#}(\vec{a}, u) \to \forall x(E(x, u) \to B^{\#}(\vec{b}, u)))$
  for each $\forall \vec{t}(A(\vec{a}) \to \forall x B(\vec{b})) \in \mathcal{S}$.
- $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to \exists x(E(x, u) \wedge A^{\#}(\vec{z}, u)) \to B^{\#}(\vec{b}, u))$
  for each $\forall \vec{z}(\exists x A(\vec{a}) \to B(\vec{b})) \in \mathcal{S}$
- $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to A^{\#}(\vec{b}, u) \to \exists x(E(x, u) \wedge B^{\#}(\vec{b}, u)))$
  for each $\forall \vec{z}(A(\vec{a}) \to \exists x B(\vec{b})) \in \mathcal{S}$

Then $\varphi = \bigwedge \mathcal{S} \to P$ is intuitionistically valid if and only if

$$\varphi' = \text{World}(s) \to K(\varphi) \to \bigwedge S' \to P^{\#}(s)$$

is classically valid, where s is a new constant symbol. This was obvious in the propositional case but here it is more nuanced.

Proof. We proceed by translation of counter-examples. Suppose first we have a counter-example $\mathcal{M} = (M, I)$ to φ'. As a Kripke frame $(W, \leq)$ take all $W = \{m \in M \mid \text{World}^I(m)\}$ let $\leq$ be $\preceq^I$ restricted to W. Then let $M_u = \{m \in M | E(m, u)\}$ and let f^{I_u} be f^I restricted to M_u and $A^{I_u}(\vec{x}) :\Leftrightarrow A^{\#}(\vec{x}, u)$. It is then a straightforward check of definitions that this defines a Kripke counter-model to φ.

The other direction is a bit more involved. Suppose we have a Kripke counter-model to φ with frame $(W, \preceq)$ and family of Σ-structures $(M_w, I_w)_{w \in W}$. In particular, since it is a counter-model there exists $w_0 \in W$ with $w_0 \not\models \varphi$. Let $W_0 = \{w \in W \mid w \geq w_0\}$ and define an equivalence relation $\sim$ on $\{(x, u) \mid u \in W_0, x \in M_u\}$ via $(x, u) \sim (y, w)$ iff $x = y$ and there exists $v \in W_0$ comparable with both u, w such that $x \in v$ and denote the equivalence class of (x, u) with $[x, u]$. Let $M = W_0 \cup \{[x, u] \mid u \in W_0, x \in M_u\}$. Now have

- $s^I = w_0$.
- $E^I(m, w)$ iff $w \in W_0$ and $m \sim [x, w]$ for some $x \in M_w$.
- $f^I(m_1 \dots m_n) = \begin{cases} f^{I_u}(x_1 \dots x_n), & \text{if there are } u \in W_0, x_i \in M_u \text{ with } m_i \sim [x_i, u] \text{ for all } i, \\ w_0, & \text{otherwise.} \end{cases}$
- $A^{\#^I}(m_1 \dots m_n, u) \Leftrightarrow \begin{cases} A^{I_u}(x_1 \dots x_n), & \text{if } u \in W_0 \text{ and } \exists x_i \in M_u \text{ w. } m_i \sim [x_i, u] \text{ for all } i, \\ \top, & \text{otherwise.} \end{cases}$

One easily verifies that these are well-defined. Let us now briefly check that this indeed defines a counter-model. First of all $P^{\#^I}(s^I)$ is false since $P^{I_{w_0}}$ is. Clearly World(w_0) holds and $K(\varphi)$ is also easily verified. All that remains to show is $M, I \models \bigwedge \mathcal{S}'$. Consider e.g. the case where $\psi \in \mathcal{S}$ is of the form

$$\forall \vec{z}((A(\vec{a}) \to B(\vec{b})) \to C(\vec{c}))$$

We have to show that

$$M, I \models \forall \vec{z} \forall u(\vec{E}(t, u) \to \forall w(u \preceq w \to A^{\#}(\vec{a}, w) \to B^{\#}(\vec{b}, w)) \to C^{\#}(\vec{c}, u))$$

holds. Suppose towards contradiction that it does not, i.e. there are $u, \vec{z} \in M$ such that for all $w \in M$

$$\vec{E}^I(\vec{z}, u) \to (u \preceq w \to A^{\#^I}(\vec{a}, w) \to B^{\#^I}(\vec{b}, w)) \to C^{\#^I}(\vec{c}, u)$$

is false. For that $C^{\#^I}(\vec{c}, u)$ must be false and in particular $u \in W_0$. Futhermore $\vec{E}^I(\vec{z}, u)$ must be true and we can write $\vec{z} = [z_1, u] \dots [z_n, u]$ for some $z_i \in W_u$. Note that the above is false in particular for $w \in M_0$ with $u \preceq w$. But this would imply

$$((A^{I_u}(\vec{a}) \to B^{I_u}(\vec{b})) \to C^{I_u}(\vec{c}))[\vec{z}/z_1 \dots z_n]$$

in our original Kripke counter-model. The other cases are analogous. $\square$

4.3.2 Reducing the encoding

As in the predicate case, we now show the existence of a canonical counter-model (in case such a counter-model exists). Denote the set of formulas $\psi \in \mathcal{S}$ of the form $\forall \vec{z}((A(\vec{a}) \to B(\vec{b})) \to C(\vec{c}))$ with $\mathcal{F}_\to$ and of the form $\forall \vec{z}(\forall x A(\vec{a}) \to B(\vec{b}))$ with $\mathcal{F}_\forall$. Let $\mathcal{F} = \mathcal{F}_\to \cup \mathcal{F}_\forall$. For each $\psi \in \mathcal{F}$ consider a new function symbol f_ψ. As before we may assume that s is a least element and we can avoid the quantification over w by considering the formulas

- $\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (A^\#(\vec{a}, f_\psi(\vec{z}, u)) \to B^\#(\vec{b}, f_\psi(\vec{z}, u))) \to C^\#(\vec{c}, u))$ instead of $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to \forall w(u \preceq w \to A^\#(\vec{a}, w) \to B^\#(\vec{b}, w)) \to C^\#(\vec{c}, u))$,
- $\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (\forall x(E(x, f_\psi(\vec{z}, u)) \to A^\#(\vec{a}, f_\psi(\vec{z}, u)))) \to B^\#(\vec{b}, u))$ instead of $\forall \vec{z} \forall u(s \preceq u \to \vec{E}(\vec{z}, u) \to \forall w(u \preceq w \to \forall x(E(x, w) \to A^\#(\vec{a}, w))) \to B^\#(\vec{b}, u))$,

and by removing removing the $s \preceq u$ for the remaining formulas in $\mathcal{S}'$. We denote the resulting set of formulas by $\mathcal{S}^\#$. Then, the previous formulas are equivalid to

$$\varphi^\circ = \forall u(s \preceq u) \to \bigwedge \{\forall \vec{t} \forall u(u \preceq f(\vec{z}, u)) \mid \psi \in \mathcal{F}\} \to K(\varphi) \to \bigwedge \mathcal{S}^\# \to P^\#(s).$$

As in the previous section we can define a tree counter-model for this.

Definition 4.3.3. Suppose we are given a counter-model $\mathcal{M} = (M, I)$ for φ°. Define $\mathcal{M}_T = (M_T, I_T)$ as follows:

- M_T are the sequences on $\{(\vec{z}, \psi) \mid \psi \in \mathcal{F}, \vec{z} \in M^n\}$.
- We interpret $\preceq$ as the prefix-order.
- We set
$$f_\psi^{I_T}(\vec{z}, (\vec{z}_1, \psi_1) \dots (\vec{z}_n, \psi_n)) = (\vec{z}_1, \psi_1) \dots (\vec{z}_n, \psi_n)(\vec{z}, \psi).$$
- For relation symbols $R^\#$, we set
$$R^{\#^{I_T}}(\vec{z}, (\vec{z}_1, \psi_1) \dots (\vec{z}_n, \psi_n)) = R^{\#^I}(\vec{z}, f_{\psi_n}(\vec{z}_n, \dots (f_{\psi_1}(\vec{z}_1, s^I)) \dots)).$$

Lemma 4.3.4. If $\mathcal{M}$ is a counter-model to φ° then so is $\mathcal{M}_T$.

Even if we don't have the finiteness property of the propositional case, we have still managed to reduce the complexity of possible counter-models because we know that considering worlds of the form

$$f_{\psi_1}(\vec{z}_1, f_{\psi_2}(\vec{z}_2, \dots f_{\psi_n}(\vec{z}_n, s) \dots))$$

is sufficient. In particular, this allows us to completely eliminate $\preceq$ from the formula, that is we remove the PartialOrder($\preceq$) from $K(\varphi)$ and adjust DomainSubset(u, w) and Persistent(u, w), e.g. remove quantification over w and the precondition $u \preceq w$ from $K(\varphi)$ and replace DomainSubset(u, w) with

$$\bigwedge\{\forall\vec{z}\forall u\forall x(E(x, u) \to E(x, f_\psi(\vec{z}, u))) \mid \psi \in \mathcal{F}_\to\}$$

and Persistent(u, w) with

$$\bigwedge\{\forall\vec{z}_1\forall\vec{z}_2\forall u(A(\vec{z}_1, u) \to A(\vec{z}_1, f_\psi(\vec{z}_2, u))) \mid \psi \in \mathcal{F}_\to, A \in \Sigma \text{ is a relation symbol}\}.$$

We set

$$\varphi^\# = K \to \bigwedge S^\# \to P^\#.$$

Then, we obtain the following result:

Theorem 4.3.5. φ is intuitionistically valid iff $\varphi^\#$ is classically valid.

The complete translation is summarised in Theorem 4.3.1 (stated earlier). Let us understand why a further reduction as in the propositional case is impossible. In this case this is not such an issue, because as we started in predicate logic it is not so troublesome that we introduce some additional predicates, however it is still interesting to understand why it is required.

Definition 4.3.6. Suppose we are given a counter-model $\mathcal{M} = (M, I)$ for φ°. For $\psi = \forall\vec{z}\psi' \in \mathcal{F}_\to$ we say it is *fulfilled* at $u \in M$ with $\vec{m} \in M^n$ iff $(A^I_\psi(\vec{a}, u) \to B^I_\psi(\vec{b}, u))[\vec{m}/\vec{z}]$ is false or $C^I_\psi(\vec{c}, u)[\vec{m}/\vec{z}]$ is true. For $\psi = \forall\vec{z}\psi' \in \mathcal{F}_\forall$ we say it is *fulfilled* at $u \in M$ with $\vec{m} \in M^n$ iff there is $m \in M$ with $E^I(m, u)$ such that $A^{I[m/x]}_\psi(\vec{a})[\vec{m}/\vec{z}]$ is false or $B^I_\psi(\vec{b})[\vec{m}/\vec{z}]$ is true.

We get an analogous result to Lemma 4.1.14:

Lemma 4.3.7. Let $\mathcal{M} = (M, I)$ be a counter-model to φ°.

1. ψ is fulfilled at $f^I_\psi(\vec{a}, u)$ with $\vec{a}$ for all $u \in M, \psi \in \mathcal{F}$, $\vec{a} \in M^n$.
2. If $\psi \in \mathcal{F}_\to$ is fulfilled at $u \in M$ with $\vec{a}$ then ψ is fulfilled at all $w \geq u$ with $\vec{a}$.

However part 2 of Lemma 4.3.7 fails for $\psi \in \mathcal{F}_\forall$, because even if for all x with $E(x, v)$, we have $A(x, v)$, there could be some term y with $\neg E(y, v)$ and w with $v \preceq w$ such that $E(y, w)$ and $\neg A(y, w)$. This is why we must consider sequences that have repetitions:

Example 4.3.8. Consider $\varphi = \neg(\neg\forall x A(x) \vee \neg\forall x B(x))$. There is a counter-model as shown in Figure 4.3, where we indicate at each world the elements that exist at it. For black numbers n, both $A(n)$ and $B(n)$ hold at that world, for red ones only $A(n)$, for blue ones only $B(n)$. The arrows indicate the order. $\neg\forall x A(x)$ is not fulfilled at worlds in the left column and $\neg\forall x B(x)$ is not fulfilled at worlds in the right column. In particular applying $f_{\neg\forall x A(x)}$ we must always reach the right column and applying $f_{\neg\forall x B(x)}$ we must always reach the left column. Therefore applying $f_{\neg\forall x A(x)} \circ f_{\neg\forall x B(x)}$ we cannot remain stationary.

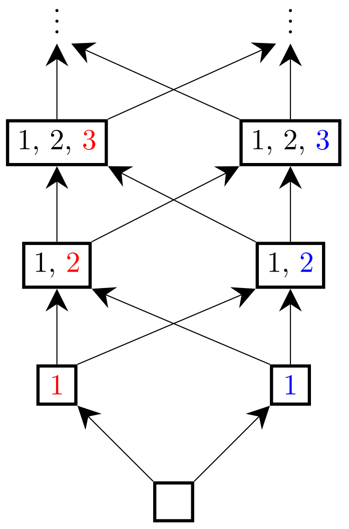

Figure 4.3: Counter-model to φ.

Let us add that Lemma 4.3.7 still equips us with additional information that can be used to guide proof search: We could add additional axioms that restrict the interpretation of functions. That is, we may assume that $f_\psi(w) = w$, if $\exists u(w = f_\psi(u))$ for all $\psi \in \mathcal{F}_\forall$ (i.e. that f_ψ is idempotent), and $f_\psi(w) = w$, if $\exists u(w \geq f_\psi(u))$ for all $\psi \in \mathcal{F}_\to$ (which is even stronger than idempotence). And indeed in our implementation we will add the first kind of additional axiom, which can be stated without the reference to the already eliminated $\geq$.

Chapter 5

Implementation

In this chapter we shall give an overview over our implementation of translation defined in the previous chapter. We have chosen to write our system in Rust. Rust was a good fit for this endeavour because:

- There is a great package for parsing files in tptp format [24] written by Michael Rawson that was tremendously helpful.
- Static typing reduced bugs induced by the many syntactically similar expressions that require different translation.
- Rust is blazingly fast.

The implementation spans over 3000 lines of code across the files

`normalized_formula.rs`, `encoding.rs`, `translation.rs`, `main.rs`.

`normalized_formula.rs` contains internal data structures representing the atomic formulas as well as normalized formulas as in $\mathcal{S}$ in Theorem 4.3.1. `encoding.rs` defines a visitor that traverses the structure of the input file and gives for each encountered subformula the corresponding encoding as in Definition 4.1.6. `translation.rs` handles additional included files. Finally `main.rs` puts it all together to define a complete translation.

5.1 Installation

Currently, the only available installation option is building the binary from source. The cargo rust complier [5] is required for this purpose. The source can be downloaded from github `https://github.com/lexpk/Proofs-as-Programs-in-Classical-Logic/tree/main/Implementation/eiicl` [2]. To build the binary go to the eiicl folder and run

```
cargo build --bin eiicl
```

The binary can be then found under

```
eiicl/target/debug/eiicl
```

5.2 Usage

The binary is to be used from the command line. It takes one or two arguments:

- The location of the tptp file that is to be translated.
- (Optionally) a file directory that contains all additional files which are to be included, per default the current directory.

Example 5.2.1. Suppose the `eiicl` binary is in the $PATH and the current directory contains a file `example.p` containing (besides a conjecture and axioms)

```
include('ax1.p').
```

as well as a folder `ax` containing a single file `ax1.p`. Then running the command

```
eiicl example.p axioms/
```

will output the correct translation according to the previous chapter.

Currently the translation does not support free variables, i.e. all free variables are expected to be explicitly quantified.

5.3 Benchmarking

	AGT	ALG	COM	CSR	GEO	GRA	GRP	HAL
Proven	5	170	0	0	58	3	4	2
Disproven	0	0	0	0	0	0	0	
Timeout	47	29	3	29	107	74	1	16

	KRS	LCL	MGT	MSC	NLP	NUM	PLA	PUZ
Proven	0	0	30	1	9	1	9	2
Disproven	0	0	0	0	12	0	0	0
Timeout	5	9	128	3	69	1	237	82

	SET	SWC	SWV	SYN	TOP	GEJ	GPJ	SYJ
Proven	0	5	27	1	64	65	54	1
Disproven	6	1	297	423	154	108	298	2

Figure 5.1: Performance on problem sections of the ILTP set.

The program was benchmarked on the ILTP problem set [4], which contains 2480 problems from the TPTP problem set [6], which is commonly used to benchmark classical provers. The problem `SYN007+1.014` was removed from the set as our translation mechanism would generate $> 2^{14}$ formulas. For the purpose of the benchmark, we report timeout for this problem. Since `SYN007+1.014` is rather artificial, this is not a big issue for us. For the classical proving part of our benchmark, we use the Vampire theorem prover [22], which is a state-of-the-art prover as evidenced by the CASC competition [1]. Then for each problem our system transforms the described transformation and feeds the result to Vampire with a time limit of 30 seconds which then reports validity, satisfiability of the negation or a timeout. The benchmark was run on an Intel Core i5-8400 CPU.

Out of the 2480 problems 511 were proven, 30 were refuted and 2128 resulted in a timeout, which corresponds to frequencies of 19.1%, 1.1% and 79.7% respectively. Out of the provers on the ILTP website this only puts it behind `ileanCoP` which managed to resolve 690 or 27.1% of the problems with a proof or refutation and ahead of five other provers. A caveat is that the ILTP benchmarks were run on a slightly less powerful machine but with a timelimit of 600s, which suggests that our result could be even better when run under the same conditions.

The ILTP problem set is divided into 24 separate sections consisting of different kinds of problems, e.g. ALG which contains problems tjat are primarily algebraic in nature. The performance on the individual sections can be seen it table 5.1. More detailed results for the individual problems can be found in Appendix C.

Chapter 6

Translation of counter-models and proofs

Finally, we elaborate on how generated proofs and counter-models for $\varphi^{\#}$ can explicitly be translated in the first-order case.

6.1 Translation of counter-models

We will merely have to summarize results here.

Definition 6.1.1. Let $\mathcal{M} = (M, I)$ be a counter-model to $\varphi^{\#}$. Then define the Kripke structure $\mathcal{K}(\mathcal{M}) = (W, (M_w, I_w))_{u \in W})$ as follows:

- As the set of worlds W take the term algebra over s and $\{f_{\vec{z}} := u \mapsto f(\vec{z}, u) \mid f \in \mathcal{F}, \vec{z} \in M\}$. Let the partial order be induced by the subterm relation.
- At each world u define the universe $M_u := \{m \in M \mid E^I(m, u^I)\}$ where $f_{\vec{z}}(t)^I := f^I(\vec{z}, t^I)$.
- For each function symbol f have $f^{I_u} := f^I|_{M_u}$.
- For each relation symbol R have $R^{I_u}(\vec{t}) :\Leftrightarrow R^{\#^I}(\vec{t}, u^I)$

Theorem 6.1.2. If $\mathcal{M} \not\models \varphi^{\#}$, then $\mathcal{K}(\mathcal{M}) \not\models \varphi$.

This follows directly from the proofs in section 4.3.

6.2 Proof translation

We want to transform Resolution proof for $\varphi^{\#}$ as given by Vampire into a **Gi** proof for φ. While the results of this section are also clear in principle, there is some additional work to be done in the details. Since syntax is of crucial importance here it might be useful to recall preliminary section 3.4. We will focus on how to obtain a **Gi** proof for φ from a **Gc** proof of $\varphi^{\#}$, which is a much closer calculus. To obtain one from a resolution proof one might first use a standard method, e.g. as given in [13], to translate the resolution proof into a **Gc** proof and then obtain a **Gi** proof via our method.

Recall that $\varphi^{\#}$ is of the form $K \to \bigwedge \mathcal{S}^{\#} \to P^{\#}(s)$, which is equivalent to $K, \mathcal{S}^{\#} \Rightarrow P^{\#}(s)$ where each $\psi \in \mathcal{S}^{\#}$ is of one of the forms

$$\begin{gathered}
\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, u) \to (B^{\#}(\vec{b}, u) \wedge C^{\#}(\vec{c}, u)))) \\
\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, u) \wedge B^{\#}(\vec{b}, u)) \to C^{\#}(\vec{c}, u)) \\
\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, u) \to (B^{\#}(\vec{b}, u) \vee C^{\#}(\vec{c}, u)))) \\
\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, u) \vee B^{\#}(\vec{b}, u)) \to C^{\#}(\vec{c}, u)) \\
\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, f_{\psi}(\vec{z}, u)) \to B^{\#}(\vec{b}, f_{\psi}(\vec{z}, u))) \to C^{\#}(\vec{c}, u)) \\
\forall \vec{z} \forall u(\forall x(E(x, f_{\psi}(\vec{z}, u)) \to A^{\#}(\vec{a}, f_{\psi}(\vec{z}, u))) \to B^{\#}(\vec{b}, u)) \\
\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, u) \to \forall x(E(x, u) \to B^{\#}(\vec{b}, u)))) \\
\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (\exists x(E(x, u) \wedge A^{\#}(\vec{a}, u)) \to B^{\#}(\vec{b}, u)) \\
\forall \vec{z} \forall u(\vec{E}(\vec{z}, u) \to (A^{\#}(\vec{a}, u) \to \exists x(E(x, u) \wedge B^{\#}(b, u))))).
\end{gathered}$$

Suppose now we have a **Gc** proof $\mathfrak{P}$ of $\varphi^{\#}$. Without loss of generality, suppose that the rules introducing the left $\forall$ quantifiers are at the end of the proof, i.e. each atom occurring until then will have ground terms in place of $\vec{z}/\vec{a}/\vec{b}/\vec{c}$ and u. This can be achieved by permutation of rules.

We can then split the proof into two parts $\mathfrak{A}$ and $\mathfrak{B}$, where in $\mathfrak{A}$ the left $\forall$ quantifiers have not yet been introduced. We can then associate with each formula ψ occurring in $\mathfrak{A}$ a world $u(\psi)$ which is the ground term that is in the end substituted for u. We say that a rule R *features* a world u if u is a subformula of $u(\psi)$ for some active formulas ψ.

Due to the subformula property, we can achieve by another permutation argument that each rule R$\to$ and R$\forall$ is followed by its associated L$\to$ rule, i.e. all rules R$\to$ and R$\forall$ occur as

$$\dfrac{\dfrac{A^{\#}(\vec{a}, f_{\psi}(\vec{z}, w)), \Gamma \Rightarrow \Delta, B^{\#}(\vec{b}, f_{\psi}(\vec{z}, w))}{\Gamma \Rightarrow \Delta, A^{\#}(\vec{a}, f_{\psi}(\vec{z}, w)) \to B^{\#}(\vec{b}, f_{\psi}(\vec{z}, w))}\text{R}\to \qquad C^{\#}(\vec{c}, w), \Gamma \Rightarrow \Delta}{A^{\#}(\vec{a}, f_{\psi}(\vec{z}, w)) \to B^{\#}(\vec{b}, f_{\psi}(\vec{z}, w)) \to C^{\#}(\vec{c}, w), \Gamma \Rightarrow \Delta}\text{L}\to$$

$$\dfrac{\dfrac{\dfrac{E(x, f_{\psi}(\vec{z}, w)), \Gamma \Rightarrow \Delta, A^{\#}(\vec{b}, f_{\psi}(\vec{z}, w))}{\Gamma \Rightarrow \Delta, E(x, f_{\psi}(\vec{z}, w)) \to B^{\#}(\vec{a}, f_{\psi}(\vec{z}, w))}\text{R}\to}{\Gamma \Rightarrow \Delta, \forall x(E(x, f_{\psi}(\vec{z}, w)) \to A^{\#}(\vec{a}, f_{\psi}(\vec{z}, w)))}\text{R}\forall \qquad B^{\#}(\vec{b}, w), \Gamma \Rightarrow \Delta}{\forall x(E(x, f_{\psi}(\vec{z}, w)) \to A^{\#}(\vec{a}, f_{\psi}(\vec{z}, w))) \to B^{\#}(\vec{b}, w), \Gamma \Rightarrow \Delta}\text{L}\to$$

for some ground term w. By another permutation argument we can move all these rule sequences to the bottom of the proof.

We will now show that $\mathfrak{A}$ can be transformed into an intuitionistically valid proof by induction on the number of such rule sequences.

Consider the last rule R$\to$. We must have $\Delta = P^{\#}(s)$. W.l.o.g. assume we are in the first case. Now consider the (possibly invalid) subproof above this rule R$\to$ consisting only of formulas that contain $f_\psi(\vec{z}, w)$ and rules that feature $f_\psi(\vec{z}, w)$.

- If this proof contains a valid proof of

  $$A^{\#}(\vec{a}, f_\psi(\vec{z}, w)), \Gamma \Rightarrow B^{\#}(\vec{b}, f_\psi(\vec{z}, w))$$

  then, substituting s for $f_\psi(\vec{z}, w)$, using the induction hypothesis we obtain an intuitionistic proof. But then applying the induction hypothesis also on the right branch of L$\to$ above we directly obtain an intuitionistic proof of

  $$(A^{\#}(\vec{a}, f_\psi(\vec{z}, w)) \to B^{\#}(\vec{b}, f_\psi(\vec{z}, w))) \to C^{\#}(\vec{c}, w), \Gamma \Rightarrow \Delta.$$

- On the other hand suppose the subproof does not contain a valid proof of

  $$A^{\#}(\vec{a}, f_\psi(\vec{z}, w)), \Gamma \Rightarrow B^{\#}(\vec{b}, f_\psi(\vec{z}, w)).$$

  Then removing all formulas containing $f_\psi(\vec{z}, w)$ and all rules featuring it and then adding missing redundant formulas in the axioms yields a valid classical proof of

  $$(A^{\#}(\vec{a}, f_\psi(\vec{z}, w)) \to B^{\#}(\vec{b}, f_\psi(\vec{z}, w))) \to C^{\#}(\vec{c}, w), \Gamma \Rightarrow \Delta.$$

  But then, since we removed one R$\to$ formula, we obtain an intuitionistic proof of the original claim via the induction hypothesis.

This gives us the following result

Theorem 6.2.1. There is an effective procedure that transforms a classical proof of $\varphi^{\#}$ into a intuitionistic proof of φ.

Example 6.2.2. Consider the intuitionistic theorem

$$\varphi = \bigwedge\{(A \to A) \to B, (A \to B) \to B\} \to B.$$

which is translated to

$$\varphi^{\#} = \exists x E(x, s) \to \bigwedge \left\{ \begin{array}{c} \forall z \forall u(E(z, u) \to (E(z, f(u)) \wedge E(z, g(u)))) \\ \forall u(A^{\#}(u) \to (A^{\#}(f(u)) \wedge A^{\#}(g(u)))) \\ \forall z \forall u(P^{\#}(z, u) \to (P^{\#}(z, f(u)) \wedge P^{\#}(z, g(u)))) \\ \forall u((A^{\#}(f(u)) \to A^{\#}(f(u))) \to B^{\#}(u)) \\ \forall u(A^{\#}(g(u)) \to B^{\#}(x, g(u))) \to B^{\#}(u)) \end{array} \right\} \to B^{\#}(s)$$

For this example, we shall consider the simplified but still valid alternative

$$\varphi^{\#} = \bigwedge \left\{ \begin{matrix} \forall u((A^{\#}(f(u)) \to A^{\#}(f(u))) \to B^{\#}(u)) \\ \forall u(A^{\#}(g(u)) \to B^{\#}(g(u))) \to B^{\#}(u)) \end{matrix} \right\} \to B^{\#}(s)$$

For this, we obtain the following proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\begin{matrix} A^{\#}(g(s)), \\ A^{\#}(f(s)) \end{matrix} \Rightarrow \begin{matrix} B^{\#}(g(s))), \\ B^{\#}(s), A^{\#}(f(s)) \end{matrix}}\,\text{Ax}}{A^{\#}(s,g(s)) \Rightarrow \begin{matrix} B^{\#}(g(s))), B^{\#}(s), \\ A^{\#}(f(s)) \to A^{\#}(f(s)) \end{matrix}}\,\text{R}\to \qquad \dfrac{}{B^{\#}(s), A^{\#}(g(s)) \Rightarrow \begin{matrix} B^{\#}(g(s))), \\ B^{\#}(s) \end{matrix}}\,\text{Ax}}{(A^{\#}(f(s)) \to A^{\#}(f(s))) \to B^{\#}(s), A^{\#}(g(s)) \Rightarrow B^{\#}(g(s)), B^{\#}(s)}\,\text{L}\to}{\forall u((A^{\#}(f(u)) \to A^{\#}(f(u))) \to B^{\#}(u)), A^{\#}(g(s)) \Rightarrow B^{\#}(g(s)), B^{\#}(s)}\,\text{L}\forall}{\forall u((A^{\#}(f(u)) \to A^{\#}(f(u))) \to B^{\#}(u)) \Rightarrow A^{\#}(g(s)) \to B^{\#}(g(s)), B^{\#}(s)}\,\text{R}\to \qquad \vdots}{\begin{matrix} \forall u((A^{\#}(f(u)) \to A^{\#}(f(u))) \to B^{\#}(u)) \\ A^{\#}(g(s)) \to B^{\#}(g(s))) \to B^{\#}(s) \end{matrix} \Rightarrow B^{\#}(s)}\,\text{L}\to}{\begin{matrix} \forall u((A^{\#}(f(u)) \to A^{\#}(f(u))) \to B^{\#}(u)) \\ \forall u(\forall x(A^{\#}(g(u)) \to B^{\#}(g(u))) \to B^{\#}(u)) \end{matrix} \Rightarrow B^{\#}(s)}\,\text{L}\forall$$

First, we push all the L∀ rules to the bottom and group L→ with R→/R∀:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\begin{matrix} A^{\#}(g(s)), \\ A^{\#}(f(s)) \end{matrix} \Rightarrow \begin{matrix} B^{\#}(g(s))), \\ B^{\#}(s), A^{\#}(f(s)) \end{matrix}}\,\text{Ax}}{A^{\#}(s,g(s)) \Rightarrow \begin{matrix} B^{\#}(g(s))), B^{\#}(s), \\ A^{\#}(f(s)) \to A^{\#}(f(s)) \end{matrix}}\,\text{R}\to \qquad \dfrac{}{B^{\#}(s), A^{\#}(g(s)) \Rightarrow \begin{matrix} B^{\#}(g(s))), \\ B^{\#}(s) \end{matrix}}\,\text{Ax}}{(A^{\#}(f(s)) \to A^{\#}(f(s)) \to B^{\#}(s)), A^{\#}(g(s)) \Rightarrow B^{\#}(g(s)), B^{\#}(s)}\,\text{L}\to}{(A^{\#}(f(s)) \to A^{\#}(f(s)) \to B^{\#}(s)) \Rightarrow A^{\#}(g(s)) \to B^{\#}(g(s)), B^{\#}(s)}\,\text{R}\to \qquad \vdots}{\begin{matrix} (A^{\#}(f(s)) \to A^{\#}(f(s)) \to B^{\#}(s)) \\ A^{\#}(g(s)) \to B^{\#}(s,g(s))) \to B^{\#}(s) \end{matrix} \Rightarrow B^{\#}(s)}\,\text{L}\to}{\begin{matrix} (A^{\#}(f(s)) \to A^{\#}(f(s)) \to B^{\#}(s)) \\ A^{\#}(g(s)) \to B^{\#}(g(s))) \to B^{\#}(s) \end{matrix} \Rightarrow B^{\#}(s)}\,\text{L}\forall}{\begin{matrix} \forall u((A^{\#}(f(u)) \to A^{\#}(f(u))) \to B^{\#}(u)) \\ \forall u(\forall x(A^{\#}(g(u)) \to B^{\#}(g(u))) \to B^{\#}(u)) \end{matrix} \Rightarrow B^{\#}(s)}\,\text{L}\forall$$

Next, we examine the last R→ rule and consider the subproof spanned by formulas containing $g(s)$ and rules featuring $g(s)$ above the last R→ rule. But there are no rules featuring $g(s)$, i.e. we obtain an invalid proof.

$$\dfrac{}{A^{\#}(g(s)) \Rightarrow B^{\#}(g(s))}$$

Therefore we remove all formulas and rules featuring $g(s)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A^{\#}(f(s)) \Rightarrow B^{\#}(s), A^{\#}(f(s))}\,\text{Ax}}{\Rightarrow B^{\#}(s), A^{\#}(f(s)) \to A^{\#}(f(s))}\,\text{R}\to \qquad \dfrac{}{B^{\#}(s) \Rightarrow B^{\#}(s)}\,\text{Ax}}{(A^{\#}(f(s)) \to A^{\#}(f(s)) \to B^{\#}(s)) \Rightarrow B^{\#}(s)}\,\text{L}\to}{\forall u((A^{\#}(f(u)) \to A^{\#}(f(u))) \to B^{\#}(u)) \Rightarrow B^{\#}(s)}\,\text{L}\forall$$

Then, we add the superfluous formulas back in the axioms:

$$
\dfrac{\dfrac{\dfrac{}{\begin{matrix}A^{\#}(f(s)),\\ \forall u(\forall x(A^{\#}(g(u))\to B^{\#}(g(u)))\to B^{\#}(u))\end{matrix} \Rightarrow B^{\#}(s), A^{\#}(f(s))}\,\text{Ax}}{\forall u(\forall x(A^{\#}(g(u)) \to B^{\#}(g(u))) \to B^{\#}(u)) \Rightarrow B^{\#}(s), A^{\#}(f(s)) \to A^{\#}(f(s))}\,\text{R}\!\to \qquad \begin{matrix}\dfrac{}{\quad}\,\text{Ax}\\ \vdots\end{matrix}}{\dfrac{\begin{matrix}(A^{\#}(f(s))\to A^{\#}(f(s))\to B^{\#}(s)),\\ \forall u(\forall x(A^{\#}(g(u))\to B^{\#}(g(u)))\to B^{\#}(u))\end{matrix} \Rightarrow B^{\#}(s)}{\begin{matrix}\forall u((A^{\#}(f(u))\to A^{\#}(f(u)))\to B^{\#}(u))\\ \forall u(\forall x(A^{\#}(g(u))\to B^{\#}(g(u)))\to B^{\#}(u))\end{matrix} \Rightarrow B^{\#}(s)}\,\text{L}\forall}\,\text{L}\!\to
$$

We then inductively proceed by examining the new last R→ rule, i.e. we consider the subproof spanned by formulas containing $f(s)$ and rules featuring $f(s)$ above the last R→ rule.

$$
\dfrac{}{A^{\#}(f(s)) \Rightarrow A^{\#}(f(s))}\,\text{Ax}
$$

This time however the subproof is completely valid. We then proceed inductively by transforming this subproof. Since it does not contain any more R→ or R∀ formulas we are done and can insert it back into the original proof and add the remaining formulas in the axioms, yielding an intuitionistically valid proof:

$$
\dfrac{\dfrac{\dfrac{}{\begin{matrix}A^{\#}(f(s))\to A^{\#}(f(s))\to B^{\#}(s), A^{\#}(f(s)),\\ A^{\#}(f(s))\to A^{\#}(f(s))\to B^{\#}(s)\end{matrix} \Rightarrow A^{\#}(f(s))}\,\text{Ax}}{\begin{matrix}A^{\#}(f(s))\to A^{\#}(f(s))\to B^{\#}(s), A^{\#}(f(s)),\\ \forall u(\forall x(A^{\#}(g(u))\to B^{\#}(g(u)))\to B^{\#}(u))\end{matrix} \Rightarrow B^{\#}(s), A^{\#}(f(s)) \to A^{\#}(f(s))}\,\text{R}\!\to \qquad \begin{matrix}\dfrac{}{\quad}\,\text{Ax}\\ \vdots\end{matrix}}{\dfrac{\begin{matrix}A^{\#}(f(s))\to A^{\#}(f(s))\to B^{\#}(s),\\ \forall u(\forall x(A^{\#}(g(u))\to B^{\#}(g(u)))\to B^{\#}(u))\end{matrix} \Rightarrow B^{\#}(s)}{\begin{matrix}\forall u((A^{\#}(f(u))\to A^{\#}(f(u)))\to B^{\#}(u))\\ \forall u(\forall x(A^{\#}(g(u))\to B^{\#}(g(u)))\to B^{\#}(u))\end{matrix} \Rightarrow B^{\#}(s)}\,\text{L}\forall}\,\text{L}\!\to
$$

Chapter 7

Conclusion

We have presented embeddings of intuitionistic into classical logic for the propositional and the predicate case. The transformation saw an exponential blow-up parameterized by $|\mathcal{F}_{\rightarrow}|$ in the propositional case and a complexity increase reflected by an arity-increase of one for all relations and the introduction of new function symbols in the predicate case. A key motivation for our work was a new approach for automated deduction for intuitionistic logic leveraging classical provers.

We have tested this approach on the ILTP problem set using the Vampire theorem provers. 20.2% of the benchmark problems were resolved, which is a decent result but falls short of the state-of-the-art. Nonetheless, this is a hopeful sign and proof that the used concept has merit.

So far, our translation is very straightforward and while this was conceptually very helpful, we expect that better results can be achieved by more elaborate mechanisms. We plan to establish better bounds in future work by utilizing structural properties of the input formula, in particular we plan to reduce the space of Kripke Frames that need to be considered by ascertaining that certain subformulas can be examined independently of each other and utilizing additional information about the functions in $\mathcal{F}$.

On the theoretical side, we also hope to give a new translation from QBF to IPC that improves our understanding of the relationship between intuitionistic propositional logic and the polynomial hierarchy.

Appendix A

Deutscher Abstract

Die berühmte Double Negation Translation stellt eine Einbettung von klassischer in intuitionistische Logik dar. Interessanterweise ist die umgekehrte Richtung in der Literatur noch nicht behandelt worden. Wir präsentieren eine effektive Einbettung von intuitionistischer in klassische Logik, sowohl im Fall der Aussagenlogik als auch im Fall der Prädikatenlogik, sowie eine effektive Einbettung von intuitionistischer Aussagenlogik in quantifizierte boolesche Formeln.

Darüber hinaus implementieren wir ein System, das intuitionistische Probleme der Prädikatenlogik im tptp-Dateiformat als Input nimmt und die Transformation durchführt. Dies ermöglicht die Verwendung von klassischen Beweissystemen zur Überprüfung intuitionistischer Gültigkeit. Wir testen unsere Implementierung mithilfe des Vampire-Beweissystems an der ILTP-Datenbank.

Schließlich diskutieren wir, wie die generierten klassischen Beweise des transformierten Problems in intuitionistische Beweise des ursprünglichen Problems zurückübersetzt werden können. Insgesamt wird damit ein neuartiger Ansatz zum automatisiertem Theorembeweis für intuitionistische Logik etabliert und ein erster Proof of Concept erbracht.

Appendix B

Zusammenfassung auf Deutsch

Mathematischer Konstruktivismus bezeichnet einen Ansatz, in dem die Existenz eines Objekts nur durch eine explizite Konstruktion nachgewiesen werden kann, im Gegensatz zur klassischen Mathematik, bei der die Existenz implizit nachgewiesen werden kann, z. B. durch die Annahme der Nichtexistenz und die Ableitung eines Widerspruchs. Der üblicherweise mit Konstruktivismus assoziierte Formalismus ist die intuitionistische Logik, die sich von der klassischen Logik im Wesentlichen dadurch unterscheidet, dass der Satz vom ausgeschlossenen Dritten $A \vee \neg A$ und der sogenannte Double Negation Shift $\forall x \neg\neg P(x) \to \neg\neg\forall x P(x)$ nicht gültig sind. Neben philosophischen Überlegungen gibt es eine besondere Motivation für das Studium der konstruktiven Mathematik aus der Perspektive der Informatik, da Beweise in der intuitionistischen Logik direkt Computerprogrammen entsprechen — wie in der Curry-Howard-Korrespondenz [19] ausgedrückt.

Das Interesse an intuitionistischer Logik hat zur Entwicklung von automatisierten Beweissystemen geführt. Die Fortschritte beim automatisierten Beweisen für intuitionistische Logik sind jedoch langsam, während Systeme für klassische Logik enorme Fortschritte gemacht haben, siehe z. B. die Wettbewerbe TPTP [1] und SAT [3]. Dies kann teilweise durch grundlegende Unterschiede zwischen den Logiken erklärt werden. Zunächst einmal ist die Bestimmung der intuitionistischen Gültigkeit rechnerisch komplexer: In der Aussagenlogik ist intuitionistische Gültigkeit `PSPACE`-vollständig [25] während klassische Gültigkeit `coNP`-vollständig [12] ist. Ein weiterer Vorteil der klassischen Logik ist die Existenz von Kalkülen, die sich besonders gut für Automatisierung eignen, wie z.B. Superposition [8], die sich auf Normalformen wie CNF und die Dualität zwischen Gültigkeit und Erfüllbarkeit stützt. Zwar gibt es auch für intuitionistische Logik einige (wenn auch komplexere) Normalformen, doch ist die Dualität zwischen Gültigkeit und Erfüllbarkeit der Negation nicht gegeben. Daher verwenden die meisten dedizierten intuitionistischen Beweissysteme [23, 26] eine direkte Suche nach einem schnittfreien Beweis durch die umgekehrte Anwendung der Regeln eines Beweiskalküls. Dieser Ansatz führt im Allgemeinen zu einer viel komplexeren Suche und ist daher nur schwer effizient zu implementieren. Schließlich fügen wir hinzu, dass im Gegensatz zu intuitionistischen Beweissystemen eine enorme Menge an Arbeit in die Optimierung von Beweissystemen für klassische Logik, insbesondere in der Aussagenlogik, d.h. SAT-Solver, investiert wurde.

Mit dieser Arbeit wollen wir einen neuen Ansatz für das intuitionistische Theorembeweisen vorschlagen, der den Fortschritt im klassischen Theorembeweisen nutzt:

- Für jede Formel φ geben wir eine Formel $\varphi^{\#}$ an, die genau dann klassisch gültig ist, wenn φ intuitionistisch gültig ist.
- Wir prüfen die Gültigkeit/Gültigkeit von $\varphi^{\#}$ mit einem klassischen Beweissystem.
- Wir transformieren den erzeugte Beweis/das Gegenmodell von $\varphi^{\#}$ zu einem von φ.

Der schwierigste Teil dieses Ansatzes besteht darin, die Übersetzung von φ in $\varphi^{\#}$ zu geben. Interessanterweise ist die umgekehrte Richtung, die berühmte Double-Negation Translation, seit langem bekannt und geht der Aussagenlogik auf Glivenko [16] und der Prädikatenlogik auf Gödel [18] und Gentzen [15] zurück. Im propositionalen Fall ist sie besonders einfach: φ ist klassisch gültig, genau dann wenn $\neg\neg\varphi$ intuitionistisch gültig ist. Intuitiv kollabiert die Übersetzung für jede Teilformel ψ von φ die Wahrheitswerte von ψ und $\neg\neg\psi$, die klassisch, aber nicht intuitionistisch äquivalent sind. Dies gibt uns eine erste Idee, warum die umgekehrte Richtung vielleicht schwieriger ist: Wir müssen die Wahrheitswerte von ψ und $\neg\neg\psi$ disambiguieren, d.h. wenn sie beide in φ vorkommen, müssen wir eine Möglichkeit haben, ihrem jeweiligen Gegenstücken in $\varphi^{\#}$ (klassisch) unterschiedliche Wahrheitswerte zuzuordnen. Dies erfordert insbesondere die Einführung neuer Variablen in unserem Verfahren.

Während wir die Übersetzung von Formeln etablieren, führen wir auch eine effektive Übersetzung von Gegenmodellen durch, d. h. für jedes intuitionistische Gegenmodell von φ konstruieren wir effektiv ein klassisches Gegenmodell von $\varphi^{\#}$ und vice versa. Wir stellen fest, dass die Existenz von Gegenmodellen ein Schlüsselkonzept ist, das ein Dual zur Gültigkeit bildet. Die Transformation und Reduktion von Gegenmodellen auf eine Normalform ist auch das, was letztlich unsere Übersetzung ermöglicht. Neben einer Übersetzung von Gegenmodellen beschreiben wir im letzten Kapitel auch explizit, wie man klassische Beweise von $\varphi^{\#}$ in intuitionistische Beweise von φ transformiert.

Als letzten Beitrag implementieren wir unsere Übersetzung in der Programmiersprache Rust. Unsere Implementierung transformiert ein Problem der Prädikatenlogik im tptp-Format [6] und gibt das übersetzte Problem in demselben Format aus. Der Code ist auf Github [2] veröffentlicht. Anschließend testen wir unsere Implementierung an der ILTP-Datenbank [4], d.h. wir übersetzen alle Probleme in der Datenbank und lassen dann das Vampire-Beweissystem [22] auf den übersetzten Problemen laufen. Unser Ansatz kann mit bestehenden Ansätzen für das intuitionistische Theorembeweisen mithalten aber die Spitzenreiter nicht erreichen. Da es noch viel Raum für Optimierungen gibt, ist dies ein vielversprechendes erstes Zeichen.

Appendix C

Complete Benchmark Results

All benchmarks that are not listed ended with timeout.

Problem	Result	Time	Problem	Result	Time
AGT001+1.p	Proven	0.054s	ALG048+1.p	Proven	1.586s
AGT001+2.p	Proven	0.085s	ALG049+1.p	Proven	2.467s
AGT002+1.p	Proven	0.059s	ALG050+1.p	Proven	2.347s
AGT002+2.p	Proven	0.066s	ALG051+1.p	Proven	1.639s
AGT017+2.p	Proven	4.868s	ALG052+1.p	Proven	3.164s
ALG014+1.p	Proven	0.386s	ALG053+1.p	Proven	2.168s
ALG015+1.p	Proven	0.749s	ALG054+1.p	Proven	0.169s
ALG016+1.p	Proven	0.405s	ALG055+1.p	Proven	0.899s
ALG017+1.p	Proven	0.043s	ALG056+1.p	Proven	0.892s
ALG020+1.p	Proven	0.205s	ALG057+1.p	Proven	1.005s
ALG021+1.p	Proven	0.457s	ALG058+1.p	Proven	0.673s
ALG022+1.p	Proven	0.493s	ALG059+1.p	Proven	0.976s
ALG023+1.p	Proven	0.713s	ALG060+1.p	Proven	1.346s
ALG024+1.p	Proven	1.059s	ALG061+1.p	Proven	0.594s
ALG025+1.p	Proven	2.532s	ALG062+1.p	Proven	0.440s
ALG026+1.p	Proven	1.316s	ALG063+1.p	Proven	0.456s
ALG027+1.p	Proven	5.175s	ALG064+1.p	Proven	0.782s
ALG028+1.p	Proven	0.355s	ALG065+1.p	Proven	0.528s
ALG031+1.p	Proven	0.375s	ALG066+1.p	Proven	0.553s
ALG032+1.p	Proven	1.275s	ALG067+1.p	Proven	0.574s
ALG033+1.p	Proven	1.182s	ALG068+1.p	Proven	0.962s
ALG034+1.p	Proven	9.518s	ALG075+1.p	Proven	0.135s
ALG035+1.p	Proven	9.285s	ALG076+1.p	Proven	0.266s
ALG036+1.p	Proven	0.302s	ALG077+1.p	Proven	0.200s
ALG037+1.p	Proven	1.004s	ALG078+1.p	Proven	0.190s
ALG038+1.p	Proven	0.762s	ALG079+1.p	Proven	0.203s
ALG039+1.p	Proven	0.042s	ALG080+1.p	Proven	0.334s
ALG042+1.p	Proven	0.202s	ALG081+1.p	Proven	0.196s
ALG043+1.p	Proven	0.196s	ALG082+1.p	Proven	0.228s
ALG044+1.p	Proven	0.249s	ALG083+1.p	Proven	0.168s
ALG045+1.p	Proven	0.571s	ALG084+1.p	Proven	0.211s
ALG046+1.p	Proven	0.832s	ALG085+1.p	Proven	0.206s

Problem	Result	Time	Problem	Result	Time
ALG086+1.p	Proven	0.273s	ALG137+1.p	Proven	1.047s
ALG087+1.p	Proven	0.305s	ALG138+1.p	Proven	0.208s
ALG088+1.p	Proven	0.195s	ALG139+1.p	Proven	0.998s
ALG089+1.p	Proven	0.293s	ALG140+1.p	Proven	1.416s
ALG090+1.p	Proven	0.270s	ALG141+1.p	Proven	0.958s
ALG091+1.p	Proven	0.466s	ALG142+1.p	Proven	0.910s
ALG092+1.p	Proven	21.681s	ALG143+1.p	Proven	1.345s
ALG093+1.p	Proven	0.551s	ALG145+1.p	Proven	1.407s
ALG094+1.p	Proven	0.280s	ALG146+1.p	Proven	1.061s
ALG095+1.p	Proven	0.358s	ALG147+1.p	Proven	6.912s
ALG096+1.p	Proven	1.859s	ALG148+1.p	Proven	1.260s
ALG097+1.p	Proven	1.531s	ALG149+1.p	Proven	1.434s
ALG098+1.p	Proven	2.604s	ALG150+1.p	Proven	1.175s
ALG099+1.p	Proven	3.372s	ALG151+1.p	Proven	29.674s
ALG100+1.p	Proven	2.517s	ALG152+1.p	Proven	3.644s
ALG101+1.p	Proven	3.101s	ALG153+1.p	Proven	26.962s
ALG104+1.p	Proven	2.526s	ALG157+1.p	Proven	2.262s
ALG105+1.p	Proven	0.765s	ALG158+1.p	Proven	4.045s
ALG106+1.p	Proven	2.808s	ALG159+1.p	Proven	14.128s
ALG107+1.p	Proven	1.168s	ALG161+1.p	Proven	1.952s
ALG108+1.p	Proven	2.324s	ALG162+1.p	Proven	2.372s
ALG109+1.p	Proven	1.958s	ALG164+1.p	Proven	1.593s
ALG110+1.p	Proven	1.049s	ALG165+1.p	Proven	1.910s
ALG111+1.p	Proven	0.960s	ALG166+1.p	Proven	1.755s
ALG112+1.p	Proven	1.348s	ALG167+1.p	Proven	1.744s
ALG113+1.p	Proven	0.846s	ALG168+1.p	Proven	0.963s
ALG114+1.p	Proven	0.889s	ALG169+1.p	Proven	0.947s
ALG115+1.p	Proven	1.004s	ALG170+1.p	Proven	0.746s
ALG116+1.p	Proven	1.253s	ALG171+1.p	Proven	0.080s
ALG117+1.p	Proven	1.121s	ALG172+1.p	Proven	0.076s
ALG118+1.p	Proven	1.133s	ALG173+1.p	Proven	0.085s
ALG119+1.p	Proven	1.539s	ALG174+1.p	Proven	0.080s
ALG120+1.p	Proven	1.905s	ALG175+1.p	Proven	0.088s
ALG121+1.p	Proven	0.962s	ALG180+1.p	Proven	0.225s
ALG122+1.p	Proven	2.674s	ALG181+1.p	Proven	0.143s
ALG123+1.p	Proven	1.099s	ALG182+1.p	Proven	0.135s
ALG124+1.p	Proven	1.390s	ALG183+1.p	Proven	0.138s
ALG125+1.p	Proven	1.243s	ALG184+1.p	Proven	0.142s
ALG126+1.p	Proven	2.903s	ALG185+1.p	Proven	0.499s
ALG127+1.p	Proven	2.391s	ALG186+1.p	Proven	0.391s
ALG128+1.p	Proven	6.782s	ALG187+1.p	Proven	0.430s
ALG130+1.p	Proven	0.548s	ALG188+1.p	Proven	0.445s
ALG131+1.p	Proven	7.478s	ALG189+1.p	Proven	0.508s
ALG132+1.p	Proven	1.040s	ALG190+1.p	Proven	2.462s
ALG133+1.p	Proven	7.531s	ALG191+1.p	Proven	2.207s
ALG134+1.p	Proven	1.663s	ALG192+1.p	Proven	0.648s
ALG135+1.p	Proven	0.914s	ALG193+1.p	Proven	0.584s
ALG136+1.p	Proven	1.122s	ALG194+1.p	Proven	28.952s

Problem	Result	Time
ALG195+1.p	Proven	4.695s
ALG196+1.p	Proven	7.115s
ALG197+1.p	Proven	4.805s
ALG198+1.p	Proven	0.127s
ALG199+1.p	Proven	1.533s
ALG200+1.p	Proven	0.263s
ALG203+1.p	Proven	26.932s
ALG204+1.p	Proven	0.242s
ALG205+1.p	Proven	0.320s
ALG206+1.p	Proven	0.274s
ALG207+1.p	Proven	0.823s
ALG208+1.p	Proven	0.703s
ALG209+1.p	Proven	0.783s
ALG210+2.p	Proven	6.846s
ALG211+1.p	Proven	0.327s
GEJ002+1.p	Proven	7.672s
GEJ002+2.p	Proven	0.030s
GEJ004+3.p	Proven	23.293s
GEJ007+2.p	Proven	0.156s
GEJ007+4.p	Proven	0.098s
GEJ009+1.p	Proven	0.487s
GEJ009+2.p	Proven	0.054s
GEJ009+3.p	Proven	0.338s
GEJ009+4.p	Proven	0.046s
GEJ010+1.p	Proven	2.915s
GEJ010+2.p	Proven	7.168s
GEJ010+3.p	Proven	1.635s
GEJ038+1.p	Proven	0.029s
GEJ038+2.p	Proven	0.028s
GEJ042+1.p	Proven	0.057s
GEJ042+2.p	Proven	0.055s
GEJ043+2.p	Proven	17.815s
GEJ044+2.p	Proven	29.105s
GEJ051+1.p	Proven	0.030s
GEJ051+2.p	Proven	0.030s
GEJ060+1.p	Proven	0.082s
GEJ061+1.p	Proven	0.153s
GEJ062+1.p	Proven	0.036s
GEJ063+1.p	Proven	0.085s
GEJ064+1.p	Proven	0.093s
GEJ065+1.p	Proven	0.062s
GEJ066+1.p	Proven	0.078s
GEJ067+1.p	Proven	0.080s
GEJ068+1.p	Proven	0.071s
GEJ069+1.p	Proven	0.073s
GEJ070+1.p	Proven	0.111s
GEJ071+1.p	Proven	0.117s
GEJ072+1.p	Proven	0.103s
GEJ073+1.p	Proven	0.106s
GEJ074+1.p	Proven	0.102s
GEJ075+1.p	Proven	0.106s
GEJ076+1.p	Proven	0.091s
GEJ077+1.p	Proven	0.093s
GEJ078+1.p	Proven	0.090s
GEJ079+1.p	Proven	0.095s
GEJ080+1.p	Proven	0.120s
GEJ081+1.p	Proven	0.063s
GEJ082+1.p	Proven	0.071s
GEJ083+1.p	Proven	0.082s
GEJ084+1.p	Proven	0.088s
GEJ085+1.p	Proven	0.074s
GEJ086+1.p	Proven	0.083s
GEJ087+1.p	Proven	0.108s
GEJ088+1.p	Proven	0.108s
GEJ089+1.p	Proven	0.121s
GEJ090+1.p	Proven	0.125s
GEJ091+1.p	Proven	0.102s
GEJ092+1.p	Proven	0.110s
GEJ093+1.p	Proven	0.106s
GEJ094+1.p	Proven	0.117s
GEJ095+1.p	Proven	0.113s
GEJ096+1.p	Proven	0.119s
GEJ097+1.p	Proven	0.091s
GEO080+1.p	Proven	0.227s
GEO085+1.p	Proven	0.266s
GEO086+1.p	Proven	0.697s
GPJ001+1.p	Proven	13.428s
GPJ001+2.p	Proven	0.187s
GPJ002+1.p	Proven	0.067s
GPJ003+1.p	Proven	0.077s
GRA010+1.p	Proven	0.038s
GRA010+2.p	Proven	0.038s
KRS065+1.p	Proven	0.039s
KRS130+1.p	Proven	0.097s
KRS134+1.p	Proven	0.052s
KRS135+1.p	Proven	0.049s
KRS136+1.p	Proven	0.031s
KRS137+1.p	Proven	0.074s
KRS139+1.p	Proven	0.128s
KRS141+1.p	Proven	0.083s
KRS142+1.p	Proven	0.096s
KRS143+1.p	Proven	0.084s
KRS144+1.p	Proven	0.105s
KRS145+1.p	Proven	0.090s
KRS146+1.p	Proven	1.792s
KRS148+1.p	Proven	0.331s

Problem	Result	Time
KRS149+1.p	Proven	0.493s
KRS150+1.p	Proven	0.247s
KRS152+1.p	Proven	0.090s
KRS154+1.p	Proven	0.788s
KRS156+1.p	Proven	0.539s
KRS157+1.p	Proven	1.134s
KRS158+1.p	Proven	0.184s
KRS160+1.p	Proven	0.102s
KRS163+1.p	Proven	0.065s
KRS164+1.p	Proven	0.072s
KRS165+1.p	Proven	0.045s
KRS166+1.p	Proven	0.047s
KRS167+1.p	Proven	0.170s
KRS169+1.p	Proven	1.030s
KRS170+1.p	Proven	0.048s
KRS171+1.p	Proven	0.055s
LCL414+1.p	Proven	0.880s
MGT003+1.p	Proven	3.148s
MGT006+1.p	Proven	27.800s
MGT009+1.p	Proven	28.410s
MGT013+1.p	Proven	10.306s
MGT014+1.p	Proven	10.485s
MGT036+3.p	Proven	0.087s
MGT045+1.p	Proven	0.077s
MGT049+1.p	Proven	0.059s
MGT052+1.p	Proven	0.086s
MSC010+1.p	Proven	0.038s
NLP001+1.p	Proven	0.237s
NLP002+1.p	Refuted	14.428s
NLP003+1.p	Refuted	14.224s
NLP004+1.p	Proven	4.340s
NLP007+1.p	Proven	4.476s
NLP009+1.p	Proven	4.150s
NLP011+1.p	Proven	4.546s
NLP104+1.p	Refuted	1.138s
NLP105+1.p	Refuted	1.216s
NLP106+1.p	Refuted	1.216s
NLP107+1.p	Refuted	1.241s
NLP108+1.p	Refuted	1.088s
NLP109+1.p	Refuted	1.178s
NLP110+1.p	Refuted	1.068s
NLP111+1.p	Refuted	1.095s
NLP112+1.p	Refuted	1.112s
NLP113+1.p	Refuted	1.380s
NLP117+1.p	Proven	0.304s
NLP122+1.p	Proven	0.302s
NLP204+1.p	Proven	3.293s
NLP208+1.p	Proven	3.487s
NUM304+1.p	Proven	2.448s
NUM333+1.p	Proven	27.096s
PUZ001+1.p	Proven	0.215s
PUZ031+1.p	Proven	0.252s
PUZ047+1.p	Proven	0.248s
PUZ060+1.p	Proven	0.113s
PUZ061+1.p	Proven	0.104s
SET002+3.p	Proven	0.050s
SET019+4.p	Proven	0.051s
SET047+1.p	Proven	0.656s
SET054+1.p	Proven	0.833s
SET055+1.p	Proven	0.041s
SET061+1.p	Proven	1.388s
SET062+3.p	Proven	0.127s
SET062+4.p	Proven	0.197s
SET063+3.p	Proven	6.494s
SET148+3.p	Proven	0.052s
SET194+3.p	Proven	1.440s
SET366+4.p	Proven	0.215s
SET574+3.p	Proven	0.096s
SET575+3.p	Proven	0.035s
SET583+3.p	Proven	0.036s
SET585+3.p	Proven	4.499s
SET589+3.p	Proven	0.155s
SET590+3.p	Proven	29.211s
SET602+3.p	Proven	0.494s
SET604+3.p	Proven	0.216s
SET618+3.p	Proven	17.657s
SET627+3.p	Proven	0.401s
SET631+3.p	Proven	0.306s
SET639+3.p	Proven	13.976s
SET658+3.p	Proven	0.873s
SET687+4.p	Proven	0.223s
SET705+4.p	Proven	0.240s
SWC128+1.p	Proven	0.086s
SWV011+1.p	Proven	0.053s
SWV014+1.p	Proven	0.281s
SWV016+1.p	Refuted	0.879s
SWV018+1.p	Refuted	0.973s
SWV022+1.p	Proven	0.054s
SWV023+1.p	Proven	0.050s
SWV043+1.p	Proven	0.039s
SWV045+1.p	Proven	0.041s
SWV047+1.p	Proven	0.042s
SWV054+1.p	Proven	0.042s
SWV057+1.p	Proven	0.039s
SWV058+1.p	Proven	0.042s
SWV059+1.p	Proven	0.065s

Problem	Result	Time	Problem	Result	Time
SWV060+1.p	Proven	0.041s	SWV192+1.p	Proven	0.394s
SWV061+1.p	Proven	0.046s	SWV193+1.p	Proven	0.510s
SWV062+1.p	Proven	0.039s	SWV194+1.p	Proven	0.520s
SWV063+1.p	Proven	0.054s	SWV195+1.p	Proven	0.457s
SWV064+1.p	Proven	0.161s	SWV196+1.p	Proven	0.387s
SWV065+1.p	Proven	0.140s	SYJ001+1.001.p	Proven	0.030s
SWV066+1.p	Proven	0.044s	SYJ002+1.001.p	Proven	0.033s
SWV068+1.p	Proven	0.040s	SYJ004+1.001.p	Proven	0.052s
SWV069+1.p	Proven	12.016s	SYJ004+1.002.p	Proven	0.053s
SWV070+1.p	Proven	0.040s	SYJ004+1.003.p	Proven	0.055s
SWV071+1.p	Proven	0.054s	SYJ004+1.004.p	Proven	0.085s
SWV072+1.p	Proven	0.043s	SYJ004+1.005.p	Proven	0.078s
SWV073+1.p	Proven	0.053s	SYJ004+1.006.p	Proven	0.079s
SWV074+1.p	Proven	0.049s	SYJ004+1.007.p	Proven	0.073s
SWV075+1.p	Proven	0.057s	SYJ004+1.008.p	Proven	0.070s
SWV076+1.p	Proven	0.045s	SYJ004+1.009.p	Proven	0.096s
SWV077+1.p	Proven	17.571s	SYJ004+1.010.p	Proven	0.078s
SWV078+1.p	Proven	2.419s	SYJ004+1.011.p	Proven	0.101s
SWV080+1.p	Proven	0.051s	SYJ004+1.012.p	Proven	0.091s
SWV081+1.p	Proven	0.043s	SYJ004+1.013.p	Proven	0.104s
SWV082+1.p	Proven	0.043s	SYJ004+1.014.p	Proven	0.096s
SWV083+1.p	Proven	0.051s	SYJ004+1.015.p	Proven	0.082s
SWV084+1.p	Proven	0.045s	SYJ004+1.016.p	Proven	0.098s
SWV085+1.p	Proven	0.041s	SYJ004+1.017.p	Proven	0.092s
SWV086+1.p	Proven	0.040s	SYJ004+1.018.p	Proven	0.099s
SWV087+1.p	Proven	0.044s	SYJ004+1.019.p	Proven	0.097s
SWV088+1.p	Proven	0.041s	SYJ004+1.020.p	Proven	0.100s
SWV105+1.p	Proven	0.043s	SYJ013+1.p	Proven	0.057s
SWV106+1.p	Proven	0.040s	SYJ014+1.p	Proven	0.048s
SWV107+1.p	Proven	0.049s	SYJ015+1.p	Proven	0.050s
SWV119+1.p	Proven	0.042s	SYJ025+1.p	Proven	0.065s
SWV121+1.p	Proven	0.038s	SYJ026+1.p	Proven	0.047s
SWV123+1.p	Proven	0.050s	SYJ027+1.p	Proven	0.047s
SWV126+1.p	Proven	0.038s	SYJ028+1.p	Proven	0.068s
SWV128+1.p	Proven	0.040s	SYJ031+1.p	Proven	0.090s
SWV131+1.p	Proven	0.051s	SYJ033+1.p	Proven	0.099s
SWV132+1.p	Proven	0.052s	SYJ034+1.p	Proven	0.135s
SWV145+1.p	Proven	0.161s	SYJ035+1.p	Proven	1.537s
SWV146+1.p	Proven	0.229s	SYJ038+1.p	Proven	0.049s
SWV147+1.p	Proven	0.141s	SYJ039+1.p	Proven	0.086s
SWV148+1.p	Proven	0.138s	SYJ101+1.p	Proven	0.028s
SWV149+1.p	Proven	0.139s	SYJ102+1.p	Proven	0.032s
SWV150+1.p	Proven	0.137s	SYJ103+1.p	Proven	0.032s
SWV169+1.p	Proven	0.266s	SYJ104+1.p	Proven	0.030s
SWV171+1.p	Proven	0.388s	SYJ107+1.001.p	Proven	0.035s
SWV172+1.p	Proven	0.447s	SYJ107+1.002.p	Proven	0.047s
SWV179+1.p	Proven	0.324s	SYJ107+1.003.p	Proven	0.057s
SWV180+1.p	Proven	0.362s	SYJ107+1.004.p	Proven	0.068s

Problem	Result	Time	Problem	Result	time
SYJ108+1.p	Proven	0.026s	SYN390+1.p	Proven	0.035s
SYJ109+1.p	Proven	0.036s	SYN401+1.p	Proven	0.045s
SYJ111+1.p	Proven	0.049s	SYN402+1.p	Proven	0.044s
SYJ112+1.p	Proven	0.036s	SYN404+1.p	Proven	0.045s
SYJ114+1.p	Proven	0.046s	SYN405+1.p	Proven	0.049s
SYJ115+1.p	Proven	0.045s	SYN406+1.p	Proven	0.060s
SYJ116+1.001.p	Proven	0.058s	SYN410+1.p	Proven	0.044s
SYJ116+1.002.p	Proven	1.048s	SYN490+1.p	Refuted	13.282s
SYJ117+1.p	Proven	0.048s	SYN491+1.p	Refuted	20.030s
SYJ118+1.p	Proven	0.073s	SYN495+1.p	Refuted	5.038s
SYJ119+1.p	Proven	0.052s	SYN497+1.p	Refuted	25.937s
SYJ120+1.p	Proven	0.052s	SYN516+1.p	Refuted	3.231s
SYJ121+1.p	Proven	0.057s	SYN517+1.p	Refuted	1.989s
SYJ122+1.p	Proven	0.072s	SYN523+1.p	Refuted	1.218s
SYJ123+1.p	Proven	0.059s	SYN527+1.p	Refuted	8.835s
SYJ124+1.p	Proven	0.062s	SYN532+1.p	Refuted	3.138s
SYJ202+1.002.p	Proven	0.056s	SYN533+1.p	Refuted	23.206s
SYJ202+1.006.p	Proven	0.956s	SYN721+1.p	Proven	0.078s
SYJ204+1.002.p	Proven	0.037s	SYN722+1.p	Proven	0.105s
SYJ204+1.006.p	Proven	0.052s	SYN733+1.p	Proven	0.073s
SYJ204+1.010.p	Proven	0.073s	SYN915+1.p	Proven	0.034s
SYJ206+1.002.p	Proven	0.035s	SYN916+1.p	Refuted	0.030s
SYJ210+1.002.p	Refuted	0.037s	SYN924+1.p	Proven	0.059s
SYJ210+1.006.p	Refuted	0.050s	SYN926+1.p	Proven	0.039s
SYJ210+1.010.p	Refuted	0.064s	SYN927+1.p	Proven	0.050s
SYN044+1.p	Proven	0.045s	SYN928+1.p	Proven	0.034s
SYN045+1.p	Proven	0.048s	SYN931+1.p	Proven	0.053s
SYN054+1.p	Refuted	0.089s	SYN932+1.p	Proven	0.045s
SYN055+1.p	Proven	0.061s	SYN943+1.p	Proven	1.681s
SYN057+1.p	Proven	0.104s	SYN944+1.p	Proven	0.115s
SYN058+1.p	Proven	0.096s	SYN948+1.p	Proven	0.053s
SYN061+1.p	Proven	0.057s	SYN949+1.p	Proven	0.078s
SYN062+1.p	Proven	0.052s	SYN952+1.p	Proven	0.051s
SYN065+1.p	Proven	0.066s	SYN958+1.p	Proven	0.044s
SYN066+1.p	Proven	0.086s	SYN959+1.p	Proven	0.040s
SYN071+1.p	Proven	0.059s	SYN960+1.p	Proven	0.050s
SYN072+1.p	Proven	0.089s	SYN961+1.p	Proven	0.054s
SYN079+1.p	Proven	0.056s	SYN963+1.p	Proven	0.056s
SYN080+1.p	Proven	0.218s	SYN964+1.p	Proven	0.043s
SYN082+1.p	Proven	0.101s	SYN973+1.p	Proven	0.034s
SYN323+1.p	Proven	0.066s	SYN974+1.p	Proven	0.045s
SYN346+1.p	Proven	0.061s	SYN976+1.p	Proven	0.140s
SYN356+1.p	Proven	0.076s	SYN979+1.p	Proven	0.074s
SYN357+1.p	Proven	0.041s	SYN980+1.p	Proven	0.127s
SYN358+1.p	Proven	0.048s	TOP022+1.p	Proven	0.153s
SYN359+1.p	Proven	0.060s			
SYN379+1.p	Proven	0.062s			
SYN387+1.p	Refuted	0.029s			

Bibliography

[1] The cade atp system competition (casc). `https://www.tptp.org/CASC/28/`. Accessed: 2022-07-14.

[2] Implementation of the embedding. `https://github.com/lexpk/Proofs-as-Programs-in-Classical-Logic/tree/main/Implementation/eiicl`. Accessed: 2022-8-15.

[3] The international sat competition. `http://www.satcompetition.org/`. Accessed: 2022-07-15.

[4] The intuitionistic logic theorem proving (iltp) library. `http://iltp.de/results.html`. Accessed: 2022-07-14.

[5] The rust programming language. `https://www.rust-lang.org/`. Accessed: 2022-10-14.

[6] The tptp problem library. `https://www.tptp.org/TPTP/TR/TPTPTR.shtml`. Accessed: 2022-08-15.

[7] Matthias Baaz and Rosalie Iemhoff. The skolemization of existential quantifiers in intuitionistic logic. *Annals of Pure and Applied Logic*, 142(1-3):269–295, 2006.

[8] Leo Bachmair and Harald Ganzinger. Resolution theorem proving. *Handbook of automated reasoning*, 1(02), 2001.

[9] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[10] Errett Bishop. Foundations of constructive analysis. 1967.

[11] Luitzen Egbertus Jan Brouwer. *Over de grondslagen der wiskunde*. Maas & van Suchtelen, 1907.

[12] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

[13] Hans de Nivelle. Translation of resolution proofs into short first-order proofs without choice axioms. *Information and Computation*, 199(1-2):24–54, 2005.

[14] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[15] Gerhard Gentzen. Die widerspruchsfreiheit der reinen zahlentheorie. *Mathematische annalen*, 112(1):493–565, 1936.

[16] Valery Glivenko. Sur quelques points de la logique de m. brouwer. *Bulletins de la classe des sciences*, 15(5):183–188, 1929.

[17] Valery Glivenko. Sur quelques points de la logique de m. brouwer. *Bulletins de la classe des sciences*, 15(5):183–188, 1929.

[18] Kurt Gödel. Zur intuitionistischen arithmetik und zahlentheorie. *Ergebnisse eines mathematischen Kolloquiums*, 4(1933):34–38, 1933.

[19] William A Howard. The formulae-as-types notion of construction. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, 44:479–490, 1980.

[20] Rosalie Iemhoff. The eskolemization of universal quantifiers. *Annals of Pure and Applied Logic*, 162(3):201–212, 2010.

[21] Laura Kovács and Andrei Voronkov. First-order theorem proving and vampire. In *International Conference on Computer Aided Verification*, pages 1–35. Springer, 2013.

[22] Laura Kovács and Andrei Voronkov. First-order theorem proving and vampire. In *Computer Aided Verification*, pages 1–35. Springer Berlin Heidelberg, 2013.

[23] Sean McLaughlin and Frank Pfenning. Efficient intuitionistic theorem proving with the polarized inverse method. In *International Conference on Automated Deduction*, pages 230–244. Springer, 2009.

[24] Michael Rawson. tptp crate. `https://github.com/MichaelRawson/tptp`. Accessed: 2022-10-14.

[25] Richard Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9(1):67–72, 1979.

[26] Tanel Tammet. A resolution theorem prover for intuitionistic logic. In *International Conference on Automated Deduction*, pages 2–16. Springer, 1996.

[27] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic proof theory*. Number 43. Cambridge University Press, 2000.

[28] Grigori S Tseitin. On the complexity of derivation in propositional calculus. In *Automation of reasoning*, pages 466–483. Springer, 1983.